N° d'ordre : 19/2016 - C/IN

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediène

Faculté d'Electronique et Informatique

جامعة هواري بومدين
للعلوم والتكنولوجيا
U S T H B

THESE

Présentée pour l'obtention du **diplôme** de **DOCTORAT 3eme Cycle**

En : **INFORMATIQUE**

Spécialité : Systèmes Informatiques

Par : **ABDMEZIEM Mohammed Riad**

Sujet

Data Confidentiality in the Internet of Things

Soutenue publiquement, le 27 /04 /2016, devant le jury composé de :

M.	A. BELKHIR	Professeur	à l'USTHB	Président
M.	D. TANDJAOUI	Maître de Recherche A	au CERIST	Directeur de thèse
Mlle.	C. BENZAID	Maître de Conférences A	à l'USTHB	Examinatrice
M.	M. BENCHAIBA	Professeur	à l'USTHB	Examinateur
M.	Y. CHALAL	Professeur	à l'ESI	Examinateur
M.	Y.ZAFFOUNE	Maître de Conférences A	à l'USTHB	Examinateur

Data Confidentiality in the Internet of Things

by

Mr. Mohammed Riyadh Abdmeziem

Abstract

Internet of Things (IoT) paradigm makes billions of our everyday life objects part of the Internet. This interconnection generates a huge amount of private data, which needs to be processed, transferred, and stored. Using classical security solutions to ensure data confidentiality in this context is challenging. In fact, these solutions have to deal with high scalability requirements, heterogeneity of the involved building blocks, in addition to resources scarcity of the embedded devices such as energy and computational limitations. In this thesis, we investigate the issue of designing safe security protocols while taking into consideration IoT specificities. We proceed by studying the IoT concept, its applications, along with the different architectures and communication protocols. These latter are at the basis of any security protocol deployment. Then, we focus on the efforts of the research community by analyzing the existing state of the art approaches. Relying on the detected shortcomings, we propose four secure and energy-aware key management protocols to address the issue of establishing secure communication channels between IoT entities. To assess our protocols, we conduct an analysis using a formal validation tool to validate their security properties. Furthermore, we estimate both their computational and communication overhead to highlight energy savings. The obtained results prove the viability of our proposed solutions in the sensitive and constrained environment of IoT applications.

Thesis Supervisor: Dr. Djamel Tandjaoui
Title: Research director

Acknowledgments

First and foremost, I am thankful to Almighty Allah for providing me health, courage, and knowledge to finish this thesis.

I express my deepest gratitude to my parents, and my sister Farah for their encouragements, prayers and unconditional support without which this work would not have been completed.

I am highly grateful to my supervisor Dr. Djamel Tandjaoui (CERIST) for his support, guidance, and his constant involvement from the early stages of this research until the very end, which provided me the required motivation and guidance to complete this work.

I genuinely thank Dr. Imed Romdhani with whom I had a great pleasure and honor to work with. He opened for me the doors of his lab in Edinburgh Napier University, which allowed building strong and fruitful research collaboration.

I would like to thank Pr. Abdelkader Belkhir for having accepted to serve as the chair of my thesis jury. My gratitude goes as well to Pr. Mahfoud Benchaiba (USTHB), Dr. Chafika Benzaid (USTHB), Pr. Yacine Challal (ESI), and Dr. Youcef Zafoune (USTHB) for honoring me with their acceptance to serve as examiners in my thesis jury.

Last but not least, I sincerely thank my friends and colleagues from USTHB, CERIST, and Edinburgh Napier University for their daily emotional and moral support.

Contents

List of Figures

List of Tables

General Introduction

Internet of Things (IoT) is one of the main communication development in the last decade. Through this concept, it is possible to connect countless low-powered smart embedded objects to each other and to the Internet. The pervasive presence around us of various wireless technologies such as Radio Frequency IDentification(RFID) tags, sensors, actuators and mobile phones constitute the cornerstone of the IoT concept. These objects can send and receive data autonomously, thus opening new horizons for home, health, and industrial applications. In fact, technology advances along with increasing demand will foster a wide spread deployment of IoT's services, which would radically transform our corporations, communities and personal lives. From the perspective of a private user, IoT's introduction will play a leading role in several services.

IoT applications will spread across several areas. For instance, e-health is seen as one of the most interesting applications as it will provide medical monitoring to millions of elderly and disabled patients while preserving their autonomy and comfort. By using body sensors, physiological data is gathered and transmitted to qualified medical staff that can intervene in case of emergency. However, users are concerned about security and privacy implications of the Internet of Things. For these users, data confidentiality is among the crucial issues to be addressed. In fact, IoT applications are unlikely to fulfil a widespread deployment until they provide strong security foundations.

Ensuring data confidentiality in IoT applications necessarily passes through key management protocols that are in charge of distributing and maintaining security credentials between involved entities. These security credentials are required for any use of cryptographic means. However, two main constraints hinder the deployment of classic developed security solutions. The first limitation is related to the lack of energy power and computational capabilities in such kind of environment. The second one is related to the unprecedented amount of data generated in the context of

IoT. Therefore, IoT security protocols will have to be highly scalable.

This thesis includes two major parts. The state of art part, and the contribution part. In the state of the art, we targeted multiple goals:

- Studying the IoT concept, its applications along with its challenges.
- Addressing technologies and architectures that will constitute the basement of IoT deployment.
- Investigating security threats on IoT applications, in particular, e-health applications.
- Analyzing the adabtability of existing key management approaches in the literature to IoT characteristics.

In the contribution part, we organized our approaches into two main categories. The category of "centralized approaches", and the category of "decentralized approaches".

In the category of centralized approaches, we propose our first protocol, which aims to establish a secure channel between various constrained nodes and a Base Station. Our protocol is based on a lightweight Public Key Infrastructure (PKI) that is only used at the registration phase to establish a symmetric session key. Using this key, our protocol encrypts the exchanged data to ensure confidentiality. Furthermore, to ensure authentication, it computes a Message Authentication Code (MAC) using the same key, and adds timestamps to prevent replay attacks. In order to assess our protocol, we conduct a formal validation regarding security properties. In addition, we evaluate both communication and computational costs to highlight energy savings. We compare the energy consumption of our protocol to other protocols, such as simplified SSL protocol and simplified Kerberos protocol. The results show that our protocol is less energy consuming where its security properties are kept safe. In our second protocol, we focus on a standard-based protocol, namely, Mikey-Ticket.

This latter needs to be tailored for constrained environments in order to adapt to their resources limitations. To this end, we introduce, two solutions to tailor Mikey-Ticket to e-health environments without weakening its security properties. In the first solution, we propose a new 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) header compression scheme for Mikey-Ticket. The proposed solution allows to reduce the header length from 12 bytes to 3 bytes in the best compression case. The objective being to save energy and avoid 6LoWPAN fragmentation that may occur when a datagram size exceeds the link layer MTU [1]. Indeed, fragmentation is undesirable, as 6LoWPAN is vulnerable to fragmentation attacks. In the second solution, we propose a new exchange mode to reduce the number of exchanged messages from six to four. The main concern being to reduce the involvement of the constrained nodes in the exchange process thus reducing the computational cost.

In the category of decentralized approaches, we introduce our third protocol. This protocol is based on collaboration to establish a secure end-to-end communication channel between a highly resource constrained node and a remote entity. The secure channel allows the constrained node to transmit captured data while ensuring confidentiality and authentication. To achieve this goal, we propose offloading highly consuming cryptographic primitives to third parties. As a result, the constrained node obtains assistance from powerful entities. To assess our protocol, we conduct a formal validation regarding security properties. In addition, we evaluate both communication and computational costs to highlight energy savings. The results show that our protocol provides a considerable gain in energy while its security properties are ensured. In our fourth contribution, we tackle the issue of securing group communications. Several group key management protocols have been introduced to address this issue. However, the majority of the proposed solutions are not adapted to the IoT and its strong processing, storage, and energy constraints. In this context, we introduce a new decentralized batch-based group key management protocol called DBGK to secure multicast communications in the context of IoT. Our protocol is

[1]Maximum Transmission Unit of the IEEE 802.15.4 protocol

designed to take into account resources scarcity and the mobility of IoT devices. To mitigate the single point of failure issue, we opt for a decentralized architecture. In addition, to reduce the *1-affects-n* phenomenon, we consider that each sub group of the network is secured with a different group key. Moreover, we use a time-driven approach where a group key is used in each time slot or interval. Members only request the required keys for each particular interval. As a consequence, less memory is required to store the keys, and only active members are involved in the rekeying process, which further reduces the *1-affects-n* issue. By using a different group key for each area, our protocol allows to handle mobile members. To assess our protocol, we analytically evaluate its security properties and its performances compared to similar schemes proposed in the literature. In addition, we validate this analysis through extensive simulations using Cooja simulator.

The remaining of this thesis is organized into five chapters. In the first chapter, we present the IoT concept highlighting its applications, its building blocks and its challenges. In the second chapter, we discuss and analyse the different architectures and protocols proposed in the literature for the IoT. In the third chapter, we study and clarify the different security concepts involved in our contributions. We also introduce e-health applications that will be used as a use case scenario. Furthermore, we present an in depth overview on the state of the art of the proposed security solutions for the IoT in the literature. In the fourth and fifth chapters, we present, validate and compare our approaches with existing approaches. In chapter four, we introduce two centralized and lightweight key management protocols, while in chapter five, we present two distributed key management protocols. We conclude with our perspectives, in addition to the list of published articles as a result of our research activities during this thesis.

Part I

State of the art

Chapter 1

Introduction to the Internet of Things

In this chapter, we introduce the concept of Internet of Things (IoT). Firstly, we provide a complete definition along with a broad overview on the impact of IoT on our societies through its different applications. Secondly, we present the enabling technologies that are expected to form the building blocks of the IoT. Thirdly, we describe a high level architecure that is commonly accepted to constitute the basement of the future IoT architecture.

1.1 Definition & applications

Internet of Things (IoT) is one of the main communication development in recent years. It makes our everyday objects (e.g. health sensors, industrial equipements, vehicules, clothes, etc.) connected to each other and to the Internet. According to [19], the basic concept behind IoT is the pervasive presence around us of various wireless technologies such as Radio-Frequency IDentification (RFID) tags, sensors, actuators and mobile phones, in which computing and communication systems are seamlessly embedded. Through unique adressing schemes, these objects interact with each other, and cooperate to reach common goals. In fact, this interconnection allows the objects surrounding us to share data, to interact, and to act autonomously on behalf of their users. This prospect opens new doors toward a future, where the real and virtual world merge seamlessly through the massive deployment of embedded

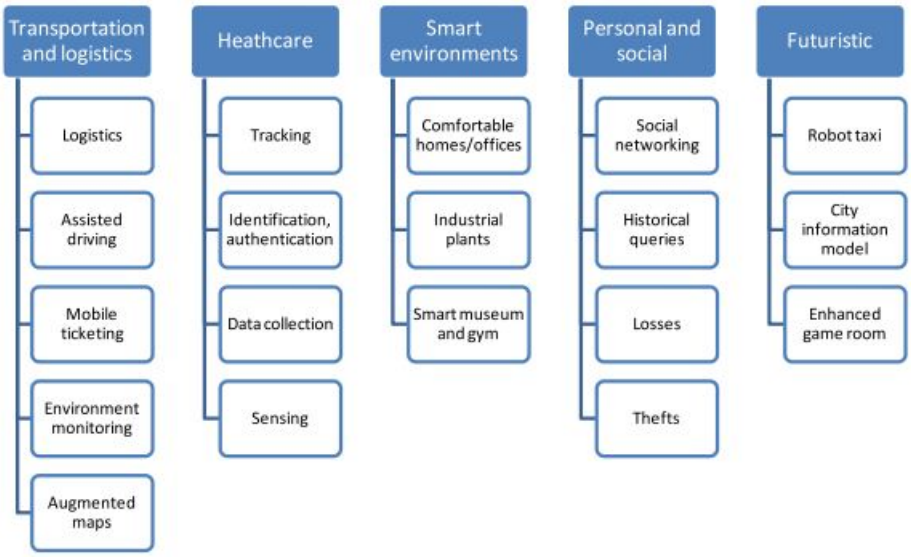

Figure 1-1: IoT Applications. Source: [19]

devices. These latter enhance dumb objects with computational, communication and storage capabilities transforming them into smart objects. By enabling interactions with and among smart objects, IoT has the potential to add a new dimension in the communication sector [90]. In addition, technology advances coupled with users need will encourage the wide spread deployement of IoT applications. These applications would deeply affect our corporations, communities, and personal lives. Indeed, enabling the objects in our everyday environment to possibly communicate with each other, and process the gathered information will open wide horizons for unpredicted applications (Figure 1-1).

From the perspective of a private use, e-health is one of the most interesting applications providing medical monitoring to millions of elderly and disabled patients while preserving their autonomy and comfort anywhere. For instance, using sensors planted in or around a patient, physiological data is gathered and transmitted to qualified medical staff that can intervene in case of an emergency. At home, energy management could be improved through the control of home equipments such as air conditioners, refrigerators, washing machines, etc. An other illustration of IoT

applications in the personal sphere relies on social networking paradigm. Indeed, an interesting development would be using a Twitter like concept. In this concept, various objects in the house can periodically tweet the readings, which can be easily followed from anywhere [49]. From the perspective of business use, environmental monitoring can be achieved by keeping track of the number of occupants, and by managing the utilities within a building. Supply chains could also benifits from the introduction of RFID and NFC (Near Field Communication) devices. As a result, real-time and precise data on the inventory of finished goods could be gathered. In addition, from the perspective of utility services, smart grids are one of the most interesting applications. Using these applications, efficient energy consumption can be achieved through continuous monitoring of electric consumption. Furthermore, gathered data is used to maintain the load balance within the grid ensuring high quality of service [141].

1.2 Building blocks

Instead of emerging as a completely new category of systems, the Internet of Things is likely to rise through an incremental development approach. In order to reach the physical realm, IoT building blocks will be progressively integrated to the existing Internet. In this section, we focus on the enabling technologies that are expected to form the IoT building blocks. Each technology is briefly introduced, along with its future impact on IoT. The different technologies are classified into three categories.

- The sensing technologies through which the required data is gathered.
- The middleware layer that is in charge of processing and managing the obtained raw data. It provides an abstraction level to users and developers.
- The actuating technologies that represent the physical extension of IoT applications. As a result, IoT would not only provide a digital support but also a physical one that can directly affect our real world.

In the following, we briefly introduce the building blocks of each category.

1.2.1 Sensing

In the IoT, wireless technologies will play a central role in data harvesting and data communication. In fact, the major part of data traffic between objects will be carried through wireless media. Wireless Sensor Networks (WSN) and radio-frequency identification (RFID) are considered as the two main building blocks of sensing and communication technologies for IoT [90]. Indeed, their ability of sensing the environment and self-organizing into ad hoc networks represent an important feature from the IoT perspective. Nevertheless, these technologies suffer from different constraints (e.g. energy limitation, realiability of wireless medium, security and privacy, etc). In particular, the scarcity of energy resources available in the embedded devices is a sensitive issue. Consequently, to increase energy efficiency, a number of solutions have been introduced in the literature. For instance, lightweight MAC protocols [140], energy efficient routing protocols [36], and tailored security protocols [11] have been proposed to mitigate the impact of resources scarcity on sensing technologies. Still, their limited autonomy remains a considerable obstacle to their widespread deployment into our daily lives. Besides, the future objects, enhanced with sensing capabilities, are expected to share a set of common characteristics and functionnalities. Indeed, these objects will have to properly manage heterogeneity in order to move towards an incremental deployment. In the following, we provide a broad presentation of RFID, WSN, and their integration into the IoT.

RFID technology is considered as an important development in the embedded devices field. RFID allows the design of tiny microchips (called tags), which can be appended to an object of our daily life. As a result, stored data in these tags can automatically be used to identify and extract useful information from the object. Thus, the tag acts as an electronic barcode.

From a hardware perspective (Figure 1-2) an RFID tag is a tiny microchip (e.g. 0.4 mm x 0.4 mm x 0.15 mm) attached to an antenna, which is used for both receiv-

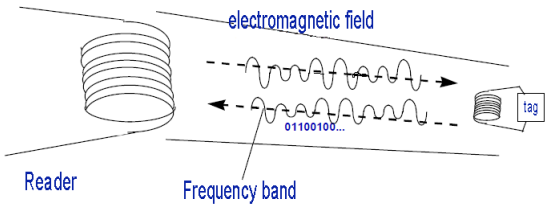

Figure 1-2: RFID tag and reader

ing the reader signal and transmitting the tag identity. The tag is manufactured in a package that can be used as an adhesive sticker [134].

Generally, RFID devices are classified into two categories: passive and active. The passive RFID tags are not battery powered. Instead, they use the power of the reader interrogation signal to communicate their data. A lot of applications from several fields use this kind of tags. Particularly, in retail, supply chain management, and transportation. They are also used in bank cards and road toll tags as an access control mean. However, the active RFID readers possess their own battery energy, and are able to trigger a communication. Although the radio coverage is more important compared to passive tags, this is obtained at the expense of higher production costs. In fact, one of the most interesting advantge in the use of RFID technology is the limited cost, which would allow a widespread adoption. Among other applications, active RFID tags can be used in port containers for monitoring cargo, robotics in a smart home context, and in hotels to provide automated check-in for customers[94].

Sensor networks on their side will also play a crucial role in the future deployment of IoT. Indeed, they can cooperate with RFID systems to better track the status of things (e.g. their location, temperature, movements, etc). Doing so, WSN are able to augment their awareness of the environment. Hence, they act as a further bridge between the physical and the digital world.

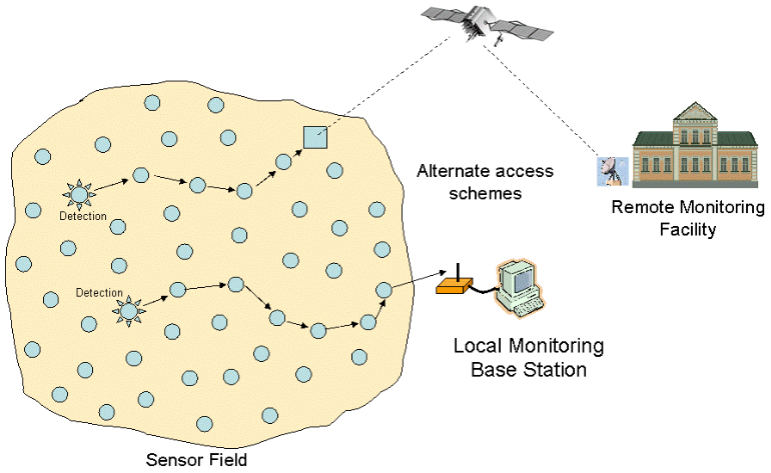

Figure 1-3: Wireless Sensor Network

Sensor networks consist of a certain number, which can be very high, of sensing nodes communicating in a wireless multi-hop fashion (Figure 1-3). In general, nodes report their sensing results to a small number of special nodes called sinks (or base stations). A lot of effort has been undertaken by the scientific community on sensor networks. Indeed, many work have addressed several problems at the different layers of the protocol stack. In these works, the main issues concern energy efficiency (which is a limited resource in WSN), scalability (the number of nodes can rise significantly), reliability (the system might be involved in critical applications), and robustness (nodes might be subject to failure) [16].

Integration of sensing technologies into passive RFID tags would bring completely new applications into the IoT context. Sensing RFID systems will allow to build RFID sensor networks, which consist of small RFID-based sensing and computing devices. RFID readers would constitute the sinks of data generated by sensing RFID tags. Moreover, they would provide the power for the different network operations. Efficiently networking tag readers with RFID sensors would allow real-time queries on the physical world. This could lead to better forecasts, new business models, and improved management techniques [142].

1.2.2 Middleware

The middleware is a software interface between the physical layer (i.e. hardware) and the application one. It provides the required abstraction to hide the heterogeneity and the complexity of the underlying technologies involved in the lower layers. In fact, the middleware is essential to spare both users and developers from the exact knowledge of the heterogeneous set of technologies adopted by the lower layers. It allows the developpers to primarily focus on issues related to the designed applications. Hence, it spares these developers losing time and efforts on issues in relation with the management and the utilization of the underlying IoT physical technologies.

The approaches based on service-oriented computing (SOC) could be in charge of playing the middleware role in the context of IoT. A service-oriented architecture (SOA) is a set of communicating services based on standardized interaction models [96]. SOC can be used to manage web services and to make them act like a virtual network. Thus, it adapts the applications to the specific users needs. Besides, Cloud computing [136] is based on a distributed architecture, in which entities are treated in a uniform way and accessed via standard interfaces. Thus, providing a common set of services and an environment for service composition. Actually, combining cloud computing with SOA could provide an efficient middleware for IoT supporting a high level of heterogeneity and flexibility.

The service based approaches lying on a cloud infrastructure open the door toward highly flexible and adaptive middleware for the IoT. For instance, Sensor-Cloud is one of the most interesting design idea to handle the huge amount of sensing devices, and the unprecedented amount of generated data. A Sensor-Could infrastructure provides to the end user service instances based on virtual sensors in a automatic way. Actually, the platform offers a virtual feeling to the user as if these sensors are part of its classical IT resources (e.g. disk storage, CPU, memory, etc.) [54]. The end users do not have to bother with their actual physical location or their actual

state. In addition, they do not even have to own the physical sensors. Instead, it is possible to create a set of sensor services to be exploited in different applications for different users through the cloud [143]. Moreover, decoupling the application logic from the embedded devices, and moving it to the cloud will allow developers to provide applications for the heterogeneous devices that will compose the future IoT environment [75].

1.2.3 Actuating

Internet of Things enhances the dumb objects around us with processing and communication capabilities. Hence, the resulted pervasive applications have the potential to deeply impact our way of life. In fact, the range of domains that might be concerned is impressive. In these domains, solutions might be deployed in both public and private areas. However, bringing to reality the future vision of our societies under the umbrella of IoT can not be achieved by limiting the scope of technology enhancement to cyberspace. Indeed, physical support (i.e. actuating) in the real world is definitely required [56].

As an illustration, let us consider an e-health scenario. E-health applications are highly promising solutions intending to provide unobtrusive support to frail and elderly people. In particular, these applications might be highly critical in case of a medical emergency. In the following, we present an e-health scenario that highlights the importance of actuating capabilities, in addition to emphasizing the involved IoT building blocks, along with their specific functionalities. Firstly, specialized sensing nodes planted in, or on a patient body are used to collect health-related data (e.g. blood glucose level), plus contextual sensors that gather data such as room temperature and humidity level. Then, gathered data is transmitted to a middleware back-end infrastructure through wireless connexion (e.g. Bluetooth, ZigBee, Wifi). Upon adequate processing, decisions can be made such as alerting medical staff, or family members. To understand the role of actuating devices, we consider the case where a hypoglycemia is detected. If the influence of the system is limited to the digital

world, the application would only trigger an alarm. Actually, an hypoglycemia could rapidly engender disastrous consequences to the brain [137]. Thus, a rapid intervention is required. In fact, waiting for emergency teams to arrive might be too late. Consequently, e-health applications have to be enhanced with actuating capabilities through which a decision to provide the patient with sugar (e.g. using an injection) can be executed immediately, probably, saving his life.

Cloud-Robotics could constitute an ideal candidate to fulfill the role of physical support to IoT applications. In fact, Cloud-Robotics abstracts robotic functionalities and provides a means for utilizing them. Various equipments and devices that can measure the world or interact with people in both the physical and digital worlds are treated uniformly. Such devices include individual robots, sensors, and smartphones. These robots are logically gathered to form a cloud of robots by networking. Hence, they realize an integrated system that provides seamless support for daily activities using the available resources on demand [67].

1.3 High level architecture

A well defined IoT architecture is still not established. However, a three-layer high level architecture is commonly accepted [139]. This architecture consists of three layers: Perception Layer, Network Layer, and Application layer (Figure 1-4). A brief description of each layer is given:

Perception Layer: the main task of the perception layer is to perceive the physical properties of things around us that are part of the IoT. This process of perception is based on several sensing technologies (e.g. RFID, WSN, GPS, NFC, etc.). In addition, this layer is in charge of converting the information to digital signals, which are more convenient for network transmission. However, some objects might not be perceived directly. Thus, microships will be appended to these objects to enhance them with sensing and even processing capabilities. Indeed, nanotechnologies

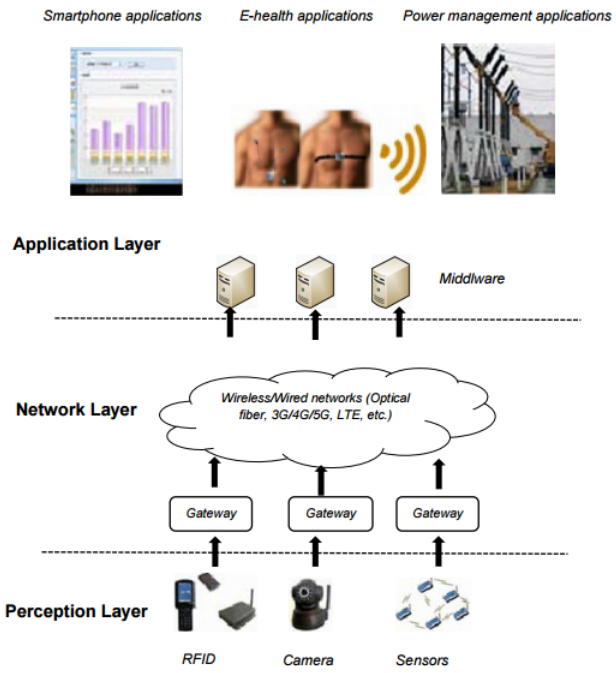

Figure 1-4: The three-layer IoT architecture

and embedded intelligence will play a key role in the perception layer. The first one will make chips small enough to be implanted into the objects used in our every day life. The second one will enhance them with processing capabilities that are required by any future applications.

Network Layer: the network layer is responsible for processing the received data from the Perception Layer. In addition, it is in charge of transmitting data to the application layer through various network technologies, such as wireless/wired networks and Local Area Networks (LAN). The main media for transmission include FTTx, 3G/4G, Wifi, bluetooth, Zigbee, UMB, infrared technology, and so on. Huge quantities of data will be carried by the network. Hence, it is crucial to provide a sound middleware to store and process this massive amount of data. To reach this goal, cloud computing is the primary technology in this layer. This technology offers a reliable and dynamic interface through which data could be stored and processed. Indeed, research and development on the processing part is significant for the future

development of IoT.

Application Layer: the application layer uses the processed data by the previous Layer. This layer constitutes the front end of the whole IoT architecture through which IoT potential will be exploited. Moreover, this layer provides the required tools (e.g. actuating devices) for developpers to realize the IoT vision. In this vision, the range of possible applications is impressive (e.g. Intelligent transportation, logistics management, identity authentication, location based services, safety, etc.).

1.4 Conclusion

In this chapter, we provided a clear overview on the IoT technology. In particular, we adopted a consensus definition encompassing the various involved building blocks. Indeed, their interconnection and integration to the Internet constitute the foundation of IoT applications deployement. We classified the different building blocks into three main categories (i.e. sensing, middleware, and actuating) based on the role they play in the overall IoT architecture. Furthermore, we introduced the three-layer architecture that provides a high level framework through which the different building blocks can be implemented. In the next chapter, we go a step further by classifying and assessing the existing IoT approaches that are based on the high level architecture, resulting either from public projects, or academic research.

Chapter 2

Deploying the Internet of Things

Before addressing security issues in the Internet of Things, an in depth comprehension of the architectures and protocols that will shape IoT deployement is crucial. In fact, designing appropriate IoT security protocols is strongly linked to the protocols and architectures adopted in designing IoT applications. However, several challenges stand between the conceptual idea of IoT and the full deployement of its applications into our daily life. Indeed, IoT successful deployment is closely related to the establishment of a standard architecture. This latter should cover IoT characteristics and support future extensions, the same way current Internet architecture achieved during the past forty years. A well defined, scalable, backward compatible, and secure architecture is required to bring the IoT concept closer to reality. In the literature, several architectures have been proposed. Nevertheless, each architecture brings a share of drawbacks, and fails covering all IoT characterisitcs. These characteristics can be summarized as follows:

- **Distributivity:** IoT will likely evolve in a highly distributed environement. In fact, data might be gathered from different sources and processed by several entities in a distributed manner.

- **Interoperability:** Devices from different vendors will have to cooperate in order to achieve common goals. In addition, systems and protocols will have to be designed in a way that allows objects (devices) from different manufacturers

to exchange data and work in an interoperable way.

- **Scalability:** in IoT, billions of objects are expected to be part of the network. Thus, systems and applications that run on top of them will have to manage this unprecedent amount of generated data.

- **Resources scarcity:** both power and computation ressources will be highly scarce.

In this chapter [14], we discuss and gather the different approaches into two categories, clean slate architectures and tailored architectures. Furthermore, we provide a thorough analysis of the proposed architectures based on their technical aspect and their ability to match IoT specificities.

2.1 Tailored architectures

2.1.1 IETF protocol suite

Given that the protocol suite TCP/IP is recognized as the cornerstone of the current Internet, it is understandable to consider the same protocol stack to be used for IoT deployment [65]. Nevertheless, IoT specificities such as resources scarcity, instable wireless links, and heterogeneity of both traffic and devices, will seriously hinder IP-based protocols deployment in IoT environments. To the end of tailoring the existing TCP/IP architecture to IoT, the Internet Engineering Task Force (IETF) is working on standardizing the corresponding communication protocols for each layer of the communication stack. Namely, IEEE 802.15.4 [6] for the data link layer, IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [123] as a lightweight addressing scheme, Routing Protocol for Low Power and Lossy Networks (RPL) [5] as a routing protocol, and Constrained Application Protocol (CoAP) [124] to be adopted in the application layer. In the following, we briefly introduce each protocol.

- *IEEE 802.15.4* is a standard developed by the IEEE 802.15 Personal Area Network (PAN) Working Group. It specifies both physical layer and media

access control for wireless constrained devices. Due to its provided features, which aim to be as less resource consuming as possible, several protocols such as WirelessHART [128] and ZigBee are based on the IEEE 802.15.4. In addition, more and more IoT devices are built as IEEE 802.15.4-compliant devices.

- *6LoWPAN* is a standard that aims to transfer IPv6 packets to IEEE 802.15.4 based networks. 6LoWPAN uses IPV6 header compression mechanisms of IPv6 datagrams. Compression mechanisms are motivated by the limited space available in 802.15.4 frames to encapsulate IPv6 packets. 6LoWPAN defines encoding formats for compression based on shared state within contexts. In other words, it takes advantage of the fields that are implicitly known to all nodes in the network or can be deduced from the MAC layer.

- *RPL* is a a standardized distance-vector routing protocol designed for constrained IP-based environments. It takes into consideration limitations either in energy power or in computational capabilities of such networks. The protocol organizes a logical representation of the network topology as a Directed Acyclic Graph (DAG). This graph is composed of one or more Destination Oriented DAGs (DODAGs) with one root per DODAG. Each root is typically a border router (BR). This latter establishes an optimum path based on defined routing metrics, which it receives thourgh broadcast messages.

- *CoAP* is an application layer protocol devloped by the IETF CoRE Working Group. It is designed for contrained environements. Based on a REST style architecture, the protocol considers the various objects in the network as resources. A Unique Universal Resource Identifier (URI) is assigned to each resource. The protocol uses the corresponding URI to operate the different resources.

2.1.2 SENSEI project

Future networks will be enhanced with ambient intelligence capabilities enabling IoT applications to spread in our environment. To realize this future vision of our com-

munications patterns, heterogeneous wireless sensor and actuator networks have to be integrated into a common framework of global scale. In addition, they have to be made available to services and applications via universal interfaces. The SENSEI project [8] solves the inaccessibility of low-resource end devices by collecting all data from the end devices, and making them available in a centrally accessible database. In fact, it provides necessary network and information management services to enable reliable and accurate context information retrieval and interaction with the physical environment.

The main results of the SENSEI project can be summarized as follows [27]:

- A highly scalable architectural framework with corresponding protocol solutions. These solutions enable easy plug and play integration of a large number of globally distributed devices (i.e. things) into a global system. Doing so, it provides support for network and information management, security, privacy and trust, and accounting.

- An open service interface and corresponding semantic specifications to unify the access to context information and actuation services offered by the system.

- Efficient WSN and actuators solutions consisting of a set of cross-optimised and energy aware protocol stacks.

- Pan European test platform. This platform enables enabling large-scale experimental evaluation of SENSEI results. In addition, it provides a tool for long term evaluation of WSN and actuators integration into IoT.

By adding mechanisms for accounting, security, privacy and trust, SENSEI will enable an open and secure market space for contextawarness and real world interactions.

2.1.3 CASAGRAS project

CASAGRAS is considered as the first view on relevant topics of the IoT (e.g. architecture, features, governance, etc.), which is the result of an international analysis and

discussion [27]. CASAGRAS project [28] aims to collect, review and analyze current and emerging proposals and solutions in the IoT. Although CASAGRAS reference architecture provides the basis for implementing a distributed IoT, the processing is not pushed to the edge of the network, which is in charge of data gathering only. In fact, the logic is located in the Information Management System Layer. The CASAGRAS model includes three layers:

- **Physical Layer:** this layer identifies physical objects, and delivers the sensed data. In order to provide interoperability, objects are organized in networks through the specific Automatic Identification and Capture (AIDC) technology. In fact, an Universal Data Capture Appliance Protocol (UDCAP) is envisioned, whereby each AIDC technology will use its own implementation of UDCAP.

- **Interrogator-Gateway Layer:** it connects object-devices with information management systems.

- **Information Management Systems:** this layer provides the functional platform for supporting applications and services.

2.1.4 Server based approach

In [23], the authors introduce a Server-Based Internet of Things Architecture (SBIOTA). The main idea is to develop protocols, algorithms and services, based on a gateway server. This latter allows networked devices with extremely limited computation and communication capabilities to be part of the IoT in an effective, efficient, and secure way. In the following, we provide a broad overview of the main features of this approach:

- **Physical and link layer connectivity:** it is assumed that each small device is directly connected to a single server, which provides an intelligent gateway function between the device and the Internet.

- **Network layer connectivity:** IP connectivity will be based on IPv6 networking. In this addressing scheme, a gateway will handle any necessary IPv4 to

IPv6 translations or tunnelling. By using IPV6, each device will have a dynamically assigned IP address. Because a full IPv6 implementation is costly for small devices, the 6lowPAN [123] protocol for communication on the links between the server and the device will be prefered. The server will also act as a firewall for each device.

- **Transport layer functions:** the two major IP-based transport layer protocols are UDP and TCP. The server will act as an endpoint for these protocols. Since UDP is more lightweight and hence more adapted to the IoT context, the server will communicate with the devices using UDP over 6lowPAN.

- **Application layer functions:** the Internet is moving away from providing access to data and hosts towards providing access to services [120]. In this context, every device will offer a HTTP web-server interface to its functionalities for authorized users. Each of these web-servers will be hosted on the gateway server.

2.1.5 Network virtualization

A solution based on virtual networks is introduced in [62]. According to the authors, current solutions that integrate smart resource-constrained objects into the Internet are mostly gateway-based. Their approach focuses on the objects, both resource-constrained and non-constrained, that need to cooperate. This integration is achieved by integrating the objects into a secured virtual network, named an Internet of Things Virtual Network or IoT- VN.

The authors have categorized the different approaches to expose services offered by resource-constrained devices to the internet into two main categories. The first one is based on using gateways that are in charge of translating between protocols used in the Internet and protocols used in the sensor networks. The second one is based on integrating sensors into the IP-world. This approach allows direct end to end communication between the end sensors.

Both approaches have their advantages and disadvantages, characterized by the degree of openness in accessing the services on the resource-constrained devices. In fact, the use of gateways has certainly many advantages (e.g. high degree of access control, offload heavy computational operations, etc.) at the expense of a reduced flexibility of usage. Besides, IP-enabled sensors allows to overcome some drawbacks of the previous approach, such as providing the possibility of having gateways and sensors from different vendors. However, allowing direct communication between resource-constrained devices, new challenges related to connectivity, scalability and security are introduced. In this context, the authors propose a novel complementary approach.

Based on the fact that in several cases there is no need to expose the data generated by resource-constrained devices to the whole network. In fact, only a limited number of devices are involved. The proposed complementary approach aims to realize a secured and confined environment in which all objects that need to cooperate can communicate in an end-to-end manner. This is achieved by creating a virtual network of all involved devices, including resource-constrained devices.

Inside this virtual network, communication can take place between the networked objects regardless whether they are resource-constrained or not. This is achieved through the use of protocols that take into account the limitations of the most resource-constrained devices. The authors described how this concept can constitute a valid alternative approach for realizing certain real-life scenarios. To reach such goal, they provide several generic use cases such as partitioning, aggregating multiple sensor networks, and extending a sensor network with non-constrained devices.

2.2 Clean slate architectures

2.2.1 BRIDGE project

The Electronic Product Code Information Services (EPCIS) are used for storage and retrieval of processed information regarding supply-chain events. EPCIS provides a complete decentralized architecture. In fact, they include two separate interfaces, one for query requests and the other one for capture operations. A secure lookup service for locating the different providers of the distributed shares of information is required. Indeed, objects full information in relation with its lifecycle history or its complete supply-chain is spread through the different entities.

To enable RFID and EPCglobal standard solutions in practice, technical, social, and educational constraints, particularly in the area of security must be overcome. BRIDGE (Building Radio frequency IDentification solutions for the Global Environment) [25] extends the Electronic Product Code (EPC) network architecture and focuses on the following aspects [27]:

a- Network

- Serial-level lookup service to enable unique item level product information storage and retrieval
- Identification and authentication of tags and readers
- Data management of large amounts of real-time data

b- Application Software

- Serial-level inventory management
- Management of large networks of EPC readers
- Models to exploit environmental data (e.g. temperature, humidity, etc)

c- Security

- Security and privacy to prevent illicit use of EPC
- Prevention of cloning and emulation of tags in EPC
- Secure transmission of data between readers and tags

In a nutshell, BRIDGE aims to enable the deployment of EPC global applications in Europe. Its main axis are focused on developing security mechanisms in hardware, software, and business practises.

2.2.2 IDRA approach (direct connectivity)

In the future IoT, a tremendous amount of heterogeneous devices (i.e. things) using vendor-specific proprietary network solutions will be connected. As a result, communication will only be possible through the use of gateway nodes, resulting in inefficient use of the wireless medium. In fact, there is no existing architecture yet that:

- Enables optimized communication, at a network and also at a link level, between co-located heterogeneous networks without the use of complex translation gateways;
- Has been implemented and evaluated as a prototype in a large scale experimental setting;
- Is compact enough to fit even on low-resource embedded devices;
- Is fully clean slate, but is also backward compatible with legacy networks.

In order to enable an end to end communication and overcome the use of gateways, the authors in [100] have tailored the IDRA architecture [124] to the context of IoT. This latter was designed specifically to enable connectivity between heterogeneous resource constrained objects. Its main advantages can be summurized as follows:

- IDRA can connect co-located objects directly, without the need for complex translation gateways;

- The architecture is clean slate, but supports backward compatibility with existing deployments;

- Due to its low memory footprint, the architecture can be used in resource-constrained objects.

Based on its characteristics, IDRA architecture aims to provide an approach that fills the gap between the current architectures and the future IoT requirements.

2.2.3 EPC based approach

In [27], the authors present an EPC (Electronic Product Code) based Internet of Things (IoT) architecture. The key concept of this architecture is deploying EPC over heterogeneous networks. It focuses on a ZigBee network as it can collect various information. In fact, the EPC Network provides certain static information such as names and manufacturers of the objects.

According to the authors, an EPC based architecture requires a minimum set of features, such as uniquely identifying an object and automatic registration into the network. Moreover, it should provide Standard Application Programing Interfaces (APIs) to search, register, observe, and control objects made by different companies. In order to deal with the precedent requirements, the proposed architecture provides two functions. The first one is how to register new objects or devices to a home area network. The second one is how to make objects communicate through the Internet with generic protocols. The proposed EPC architecture uses combination of sensor networks and EPC networks, which provide product information through web services from the manufacturers. This architecture uses UPnP protocol to automatically collect the EPC of a new connected object. In addition, ZigBee network system is applied for communication, and XML based web services are used for the application protocol. Genuine HTTP is a heavy protocol particularly for low bandwidth network, such as ZigBee and IEEE 802.15.4. Therefore, CoAP (Constrained

Application Protocol) is adopted to support web services over ZigBee network. End to end communication is thus established regardless of the type of the network.

2.2.4 Cloud based approach

In the IoT paradigm, information and communication systems are invisibly embedded in the environment around us. This will result in the generation of huge amount of data, which has to be stored, processed and presented in a seamless, efficient, and easily interpretable way. According to [49], cloud computing is the most recent paradigm to emerge, promising high reliability, scalability, and autonomy. In fact, it provides ubiquitous access, dynamic resource discovery, and composability required for future IoT applications. This platform acts as a receiver of data from the ubiquitous sensors, as a computer to analyze and interpret data, as well as a provider to understand web based visualizations. The Cloud not only reduces costs of deploying ubiquitous applications, but is also highly scalable.

Sensing service providers can join the network and offer their data using a storage cloud, analytic tool developers can provide their software tools, artificial intelligence experts can provide their data mining and machine learning tools, and finally computer graphics designers can offer a variety of visualization tools.

Cloud computing can offer these services to the IoT as infrastructures, platforms, or softwares where the full potential of human creativity can be exploited. The generated data, used tools, and the process of generating complex visualizations are hidden in the background.

2.2.5 Social network approach

The Social Internet of Things (SIoT) architecture is introduced in [20]. The approach establishes a link between social networks and IoT. The main idea is that a large number of individuals tied in a social network can provide far more accurate answers

to complex problems than a single individual (even knowledgeable one). In the future, things will be associated to the services they can deliver. Thus, to better implement services within a given social network of objects, a key objective will be to publish information/services, find them, and discover novel resources. This can be achieved by navigating a social network of 'friend' objects instead of relying on typical Internet discovery tools that cannot scale to the trillions of future devices.

Authors in [20], claim that social relationships among humans might be applicable to certain kinds of behaviors of typical objects implementing pervasive applications. There is no doubt that many applications and services should be associated with groups of objects, which will cooperate in order to reach the overall interest of providing services to users (e.g. the same idea is behind the approaches involving the use of swarm intelligence and swarm robotics).

The social architecture relies upon basic kinds of relationships such as the **Parental object relationship (POR)**, which is established among objects belonging to the same production batch, or the **Ownership object relationship (OOR)**, which is based on heterogeneous objects belonging to the same user (e.g. mobile phones, game consoles, etc.). The authors draw attention about the fact that the establishment and the management of such relationships should occur without human intervention. This is not in contrast with a future vision of fully networked humans. These latter are only responsible for setting the rules of the objects and their social interactions. This is a clear paradigm shift from other proposals, where the objects just participate in the human social network built by their owners.

2.3 Critics and Analysis

The proposed architectures, either the public projects or those introduced by the research community, aim to reduce the gap between the concept of the IoT and its real deployment into our daily lives. We have proposed a classification that gathers the

different architectures into two categories. The first one, called the tailored architectures, contains the approaches that propose an evolution of the current Internet to a more suitable network for the IoT such as network virtualization, and server based approach. This category will certainly provide the advantage of backward compatibility with existing architectures. However, several issues remain such as security and resources limitations. The second category includes clean slate architectures such as the IDRA approach and the social network approach. These approaches claim a novel vision of the future IoT that inherently copes with next-generation network challenges. In fact, this provides the benefit of a design, completely dedicated to be tailored to IoT characteristics. Nevertheless, backward compatibility with existing approaches remains a challenge.

In the following, we propose an analysis of each architecture, highlighting the matching of its characteristics with IoT requirements.

The IETF is focusing its efforts on adapting existing protocols, which have been developed for the classical Internet to the constrained environment of IoT. To this end, the IETF proposes an equivalent to the existing protocols for each layer of the TCP/IP stack, such as 6LoWPAN for IPV6 and CoAP for HTTP. However, although the precedent solutions constitute a sound basement on which further efforts can be made, several challenges should be addressed. For instance, the limited channel capacity of the IEEE 802.15.4 can hinder the scalability and the traffic load of future IoT applications. Moreover, Quality of Service (QoS) support for networks with heterogeneous traffic is still problematic in IEEE 802.15.4 [125]. In addition, several studies such as [86] highlight security breaches in the IETF protocol suite. Thus, the IETF protocol suite has to be strengthen regarding the security aspect, which is considered as a primary concern in the IoT.

SENSEI [8] focuses on equipping the objects with a certain kind of intelligence by embedding processing capabilities into them. The project provides the architecture

for connecting heterogeneous objects via the specification of open service interfaces. However, the use of centrally accessible database results in a significant network overhead, and could constitute a single point of failure. Additionally, the SENSEI project is still under development. It needs to reach a mature state before an effective evaluation. CASAGRAS [28] also proposes a vision of the IoT whereby both virtual and physical generic objects are connected through a global infrastructure. The project focuses too much on RFID as the main building block of the IoT while it is likely to have a multitude of integrated technologies forming the future IoT. Like SENSEI, CASAGRAS presents a narrow-waist. Any interaction has to pass through the Management System at the service, or application layer. BRIDGE [25] aims to research, develop, and implement tools to enable the deployment of Radio Frequency Identification (RFID) and EPCglobal Network applications. The core of BRIDGE is communication centric. It addresses the problem of handling queries between distributed entities. Nevertheless, the work with sensors does not extend the EPC network standards.

The IDRA [100] architecture proposes a clean slate approach that challenges the layered vision of the current internet architecture. IDRA aims to enable a direct connectivity between heterogeneous objects through a network-service-oriented architecture. However, additional processing might impede an efficient deployment in a resource-constrained environment. The virtualization approach [62] also aims to establish an end-to-end communication between the devices that need to cooperate. In fact, this approach integrates them into a secured virtual network regardless whether the resources are constrained or not. Yet, the scalability has not been proven, and the complexity of the protocols used might be an issue. To provide an end to end communication regardless of the type of the access network, another promising architecture has been presented in the EPC based approach [51]. The main idea is to combine sensor networks with EPC networks, which provide product information through web services from manufacturers. Server based approach [23] proposes a different solution to connect networks from different vendors, or devices that use different protocols.

The idea is to use a translation gateway. Nevertheless, this solution breaks the end to end communication principle. In addition, the gateway could represent a single point of failure.

The social network approach [20] introduces an interesting idea by making the parallel between the current social networks and a future network of objects. The goal is to publish, find information, and discover novel resources to better implement the services. Nevertheless, this approach does not deal with the issues of lower layers of the network. Besides, in order to take into account the scarcity of resources in the future IoT, the cloud approach [49] proposes offloading resource intensive tasks to more capable nodes. In fact, the cloud offers both flexibility and a high scalability level. However, the cloud architecture does not deal with the connectivity challenges at lower levels of the network.

2.4 Conclusion

In this chapter, we introduced a classification highlighting the suitability of the proposed architectures to IoT characteristics. In addition, we underlined the main shortcomings of the current approaches. In a nutshell, we do believe that a well-defined architecture is required instead of letting the current Internet raise to the IoT in an uncontrolled way. Issues like security need to be addressed during design time. In addition, we consider that the different proposed architectures are not contradictory; an hybrid architecture including several approaches might be an efficient way to address the IoT specificities. Based on the commonly accepted three-layer architecture, each approach might be implemented in the appropriate layer. For instance, the cloud approach affects the application layer whereby the future applications will need to be ubiquitously accessible, while the IDRA approach could be implemented in the network layer to secure a dynamic adaptation of the network.

Chapter 3

Security in the Internet of Things

In this third chapter, we discuss the security aspect in the Internet of Things. Firstly, we introduce the main security concepts involved throughout the remaining of the thesis highlighting the differences between them. Then, we present e-health applications in the context of IoT as a use case scenario along with the main security threats that might limit their deployment. We conclude this chapter with an in depth discussion and critic of existing security approaches in the literature.

3.1 Concepts definition

In the following, we provide concise definitions of key concepts involved in our work. The aim is to allow the reader to properly make the difference between the involved concepts.

3.1.1 Data security and privacy

Privacy and security are often used interchangeably. Although these two concepts are closely related, important differences exist. In fact, privacy is related to persons. It ensures that persons keep control over the information they disclose in the context of a particular application (e.g. on the Internet). Indeed, ensuring privacy means that personal information disclosed for a specific purpose with specific entities are

not made available to other unauthorized entities, and not exploited to infer further information. Security on its side is related to data, and usually refers in the literature to the different means that are deployed in order to guarantee a set of properties. In the following, we provide a brief definition of each property.

- *Confidentiality:* ensures that, apart from the authorized involved enitites, the exchanged data during a communication are kept confidential. Confidentiality is generally ensured through encryption.

- *Integrity and Authenticity:* integrity ensures that exchanged data between two entities during a communication process has not been altered by unauthorized entities. However, authenticity validates the origin of the data. Message Authentication Code (MAC) messages are used to provide both properties.

- *Availability* ensures that data is available when needed by authorized entities. This implies that the communication system has to remain functional despite security attacks (i.e. Denial of Service) and hardware failures. Backup systems and redundancy are used as a mean to provide availability.

- *Non-repudiation:* ensures the means to verify that an entity has actually participated in an exchange of information, such as sending/receiving information or a digital signature.

- *Access control:* ensures that the involved entities are authorized to be part of the communication, and that protected information is only accessed by authorized entities. Access control is usually ensured through three successsive steps. **Identification** which is a claim of identity (i.e. who someone is or what something is). This claim is then verified through **Authentication**. This step ensures that the identities provided by the involved entities are correct. Upon successful identification and authentication, **Authorization** allows to determine what information can be accessed and what actions can be carried out.

The relationship between security and privacy is that security is necessary but not sufficient to protect privacy. In fact, any breach in security properties, in particular

data confidentiality, will have a direct impact on privacy. Nevertheless, eventhough security properties are ensured, volontarily disclosed data can be used by malicious entities to infer information for illegal purposes.

3.1.2 Data confidentiality and key management protocols

Ensuring data confidentiality is crucial for IoT applications. In fact, any failure would seriously threaten users privacy. Thus, a wide deployment of IoT applications might be hindered. To provide data confidentiality, cryptographic algorithms are generally employed to cipher data. Doing so, even if the exchanged data is eavesdropped, the attacker will not be able to access its content. In contrast to the *security by obscurity* principle, Kerckhoffs principle [73] states that a cryptosystem should rely on the secrecy of the keys. In fact, this principle assumes that an attacker is able to access and master the cryptographic protocol. Its strength should then be placed in the secrecy of the keys. Cryptographic algorithms are categorized into two main categories.

- *Symmetric protocols:* in this category of algorithms, the same shared key between the involved entites is used to encrypt and decrypt data. The main drawback of symmetric encryption is the requirement that the involved parties have access to the shared secret key. In fact, establishing a secure channel to distibute the secret key is challenging. However, symmetric protocols are less resource consuming compared to asymmetric protocols [89]. Message Authentication Code(MAC) messages are computed using symmetric algorithms. Their aim is to provide authenticity and integrity. In fact, MAC messages are computed using as an input of a hash function (e.g. HMAC) the exchanged message and the shared symmetric key. The receiver computes its own MAC using the same shared key and compares it with the received one. If the two MAC messages are identical, it implies that the message has not being altered, thus, ensuring integrity. Otherwise, the two MAC messages would not have been identical. In addition, the compatibility of the two MAC also indicates that the

message is authentic as it ensures that an entity in possession of the shared symmetric key has sent the message. AES-CCM mode that defines AES-CBC for MAC generation with AES-CTR for encryption are examples of symmetric protocols [42].

- *Asymmetric protocols:* in this category of algorithms, a pair of public/private keys is used in the encryption/decryption process. The encrypting entity uses the public key of the receiver to encrypt data. Public keys are not kept secret. To decrypt the encrypted message, the receiver uses its private key. Unlike public keys, private keys are kept secret and only available to their owner. Digital signatures are based on asymmetric protocols. In fact, an entity signs a message by encrypting it using its private key. The receiving entity uses the public key of the sending entity to check the signature. Digital signatures provides authentication of the source of a message. Indeed, private keys are bound to a specific entity. Hence, a valid signature proves that a message is actually sent by that specific entity. Digital signatures also provides integrity considering that if a message is altered during its transmission, the signature would no longer be valid. Furthermore, non-repudiation is also guranteed as the access to the signing private key is limited to its owner. The main drawback of asymmetric protocols in the context of Internet of Things is their high enery cost compared to symmetric protocols [89]. RSA and Elliptic Curve Cryptography (ECC) are examples of asymmetric cryptographic primitives [50].

Kerckhoffs principle is widely adopted in the design of security systems [122]. Thus, key management protocols represent the cornerstone of any cryptographic system. They are in charge of generating and distributing the required keying materials. Key management protocols can be gathered into two main categories of approaches [112].

- *Pre-shared approaches:* are based on the pre-sharing of keying materials between the two entities willing to secure their communications. These keying materials are used to derive a secret shared key. The major issue with these

approaches is the initial distribution of the keying materials. In fact, the distribution is prior to any exchange of information. As a result, these protocols are not applicable between two entities that have not established upstream a shared context. Nevertheless, pre-shared approaches offer a negligeable computation overhead as no complex operation is required to establish the shared secret.

- *Public Key approaches:* are based on asymmetric primitives to establish a shared secret between two entities that have no previous pre-established context. The main issue with public key approaches is their high computation overhead. For instance, Diffie-hellman key exchange protocol [111] uses exponential operations that are costly in particular for the constrained entities of the IoT [133]. However, public key approaches offer the ability of establishing a secret between previously unknown entities, which might be necessary for future dynamic IoT applications.

To assess key management protocols, several properties are taken into consideration [127] [126]

- *Distribution:* This property is considered regarding the process through which the initial information used in the key establishment are distributed. This distribution can be achieved in an offline mode or in an online mode. In the offline mode, the required information is set upstream. On the other side, the online mode allows the involved entities to engage in an exchange process without any pre-established context. In the context of the dynamic IoT, the protocols that allow an online mode distribution are preferred.

- *Authentication:* This property ensures that the entities involved in the key exchange are authenticated. This can either be achieved through the use of digital signatures in the case of public key approaches, or through the initial shared secret in the case of Pre-shared approaches. Authentication is highly sought in IoT applications as the authenticity of data sources is crucial, in particular, for sensitive applications.

- *Extensibility:* This property is related to the possibility of involving further entities after the initial key exchange. In fact, in some key management protocols, the number of entities that can be involved in the key exchange process is limited. Extensibility is an essential property for IoT applications where the number of connected objects is high.

- *Resilience:* A key management protocol is resilient if the corruption of one entity, and thus the extraction of secret information, has limited consequences on the overall system. Ensuring this property for IoT applications would definitely strengthen the security level. In fact, entities in the context of IoT will likely remain unattended for long period of time, which make them vulnerable to physical attacks and corruption.

- *Scalability:* This property is ensured if the number of cryptographic materials stored in an entity does not scale linearly (or worst exponentially) with the implication of new entities in the key exchange process. Scalability is highly sought for IoT applications as the number of connected objects is expected to grow significantly.

- *Collusion freedom:* This property is related to the fact that any set of corrupted users are unable to access the generated secret.

Group communications (multicast) constitute an important component of future IoT communication. They include one-to-many, many-to-many, and many-to-one communications. To secure these types of communications, group key management protocols are used. These latter are in charge of generating, distributing, and maintaining a shared secret key. In addition to the required security properties of two parties key management protocols, two main security properties have to be ensured in group key management protocols [29].

- *Backward secrecy:* This property is related to the dynamic of group members. When a new member joins a group, exchanged information before its arrival can be accessed. In fact, if the new member has previously stored the

exchanged information, it would be possible to decrypt them after the receipt of the group key. Backward secrecy ensures that a new member can not access communications that have taken place before its joining.

- *Forward secrecy:* This property is considered in the case of a member departure. Forward secrecy is provided when a leaving member is not able to decrypt exchanged information after it leaves the group.

3.2 E-health applications: a use case scenario

Internet of Things deployment will open doors to a huge number of applications that would deeply improve our daily life. E-health applications are one of the typical applications that are gaining more and more attention [19]. An e-health system is defined as a radio-frequency-based wireless networking technology that provides ubiquitous networking functionalities. It is based on the interconnection of tiny nodes enhanced with sensing and/or actuating capabilities planted, or placed around the human body. E-health applications are context-aware, personal, dynamic and anticipative by nature. As IoT is designed to meet these key characteristics, it provides a natural and suitable environment for their efficient deployement. In fact, an extensive research study on using IoT paradigm in e-health has been reported [63]. Population ageing and the increase of survival chances from disabling accidents and illnesses will lead to an increased demand from today's population that requires a continuous health care and monitoring [38].

E-health applications could spare a patient from being admitted in hospitals for a long period of time. Reducing the number of nights that a patient may spend in a hospital and the associated risks that may result is a key area of focus for the medical community. Additionally, a continuous monitoring capability, if available, can anticipate the need for an emergency intervention. Moreover, early stage diagnostics could also be achieved remotely [97]. In brief, e-health applications in the context of IoT constitute a cost effective and unobtrusive solution that is of best interest of

today patients. Nevertheless, e-health applications are seriously challenged by many security threats that limit their large scale deployment.

Studies in [79][64][80][95] have underlined that e-health applications might be more vulnerable to attacks compared to other IoT applications as the generated data is highly sensitive and private. The health related records are always private in nature, and any security breach in the confidentiality of such data would seriously repulse patients from adopting e-health solutions. For instance, many people would not like their personal health information, such as early stage of pregnancy or details of certain medical conditions, be divulged to third parties [17]. In fact, the eavesdropped communications could be used for several illegal purposes. Moreover, any eventual modification of health related captured data could lead to disastrous consequences as it could engender wrong medical prescription or delay an emergency intervention.

Classical countermeasures are not suitable to the constrained environment of IoT due to several factors such as power and computation limitations, weak reliability of wireless links and the scalability issue. Thus, a considerable effort has been made by the research community to provide viable solutions to secure IoT applications. The next section provides an in-depth overview on the state of the art of the proposed security approaches and explains the motivations behind our contributions.

3.3 Related work

In this thesis, we focus on key management protocols that are in charge of establishing secret shared credentials in the context of IoT. To ensure data confidentiality, these latter are used as an input of cryptographic protocols. Security issues such as routing vulnerabilities, ensuring trust, identification challenges or securing stored data are not treated. To address the challenging aspect of establishing secret shared credentials to secure communications for the constrained IoT applications, we propose four different approaches. These approaches are rigorously introduced and validated in chapter 4

and chapter 5. In this section, we provide an overview on the existing approaches in the literature along with an in-depth critic regarding their ability to match IoT requirements.

E-health is considered as an Internet of Things application [19]. Consequently, it inherits the main IoT security threats and challenges. There is a huge literature on how security issues could hinder IoT deployment. In fact, studies have shown that security in any IoT application will be crucial as billions of intelligent things will cooperate with each other in a random and unpredictable way [113] [85][90] [135]. It has also been shown that eventhough IoT infrastructure is expected to involve protocols and interfaces similar to those running on Internet, it will be daunting to directly handle IoT threats based on classical known countermeasures due to:

- The scarcity of both power and computational resources will hinder classical solutions deployment.

- Distributivity and heterogeneity of the devices that will compose IoT (constrained and non-constrained) might lead to gaps in end to end security.

- IoT will be highly scalable and dynamic, thus, traditional public key infrastructures need to be adapted to meet these requirements.

- Things will have to manage dynamic identities to deal with context aware applications.

- Wireless connectivity will constitute the main media of communication, which could lead to different attacks such as eavesdropping and side channel attacks.

- Objects in IoT might be unattended for long period and thus are more vulnerable to physical attacks.

The creation of a secure channel between gateways (i.e. base stations) and objects (i.e. nodes) is crucial to implement security mechanisms. To establish this channel,

key management protocols are required to allow two remote devices to negotiate security credentials. An e-health system could be assimilated to a classic Wireless Sensor Network (WSN). Various approaches have been proposed in the literature to deal with key establishment process in WSN. For instance, Public Key Cryptography could be suitable if used only in early stages of a key establishment process [81]. Moreover, the Pre-Shared keys solutions could be used in limited real life scenarios where the distribution of keys in an offline mode is possible [101]. Besides, key pool paradigm includes several approaches that improve scalability while sacrificing their key connectivity [44].

Despite the fact that e-health systems are based on WSN, they present specific features that make WSN solutions in some particular applications not suitable. In fact, the number of nodes and the range between them is limited, physical access to the nodes is difficult as they are under continuous surveillance of the person carrying them. In addition, an e-health system involves heterogeneous nodes (constrained and non-constrained) that might lead to gaps in end to end security. These specific characteristics have to be taken into consideration when designing security protocols for e-health applications.

Several specific solutions for e-health applications have been proposed in the literature. For instance, hardware solutions are proposed to deal with the scarcity of resources [55] [88]. However, these approaches still present some drawbacks as they do not offer AES (Advanced Encryption Standard) decryption (only base stations can decrypt the transmitted data). In addition, they are highly platform-dependant and not all the nodes are equipped with hardware encryption capabilities. Besides, TinySec is part of the official TinyOS release that aims to achieve link-layer encryption and authentication of data in biomedical sensors [69]. This protocol is based on a single key shared among nodes which constitutes its main weakness as node capture would give access to the entire network. A different approach based on biometric techniques is therefore proposed [31] [99]. These techniques use the human body to manage the

key establishement process based on physiological values (e.g., electrocardiogram).

A different but complementary research direction has seen several interesting approaches that aim to tailor security protocols for the IP-based IoT. The main focus of these works is to make standard based security protocols suitable for constrained IoT environments. In particular, several compression schemes for the IP-based IoT have been proposed. The compression of IPv6 headers, extension headers along with UDP (User Datagram Protocol) headers has been standardized through the 6LoWPAN adaptation layer in [92] [57]. Moreover, authors in [48] and [106] have presented 6LoWPAN based compression techniques for IPsec payload headers: AH (Authentication Header) and ESP (Encapsulating Security Payload), that have been later standardized in [107]. Besides, an IKE (Internet Key Exchange) compression scheme has been also proposed in order to provide a lightweight automatic way to establish security associations for IPsec [110]. Likewise, header compression layers for DTLS (Datagram Transport Layer Security), HIP DEX (Host Identity Protocol Diet Exchange), and HIP BEX (HIP Base Exchange) were respectively introduced in [108], [58], and [116].

Apart from packet compression schemes, further design improvement approaches have been introduced to tailor security protocols to the IoT. Authors, in [60], have proposed complementary lightweight extensions to HIP DEX (Host Identity Protocol Diet Exchange) that could be generalized to DTLS (Datagram Transport Layer Security) and IKE. Following the same way, authors in [61] have introduced design ideas to reduce the overhead of the DTLS handshake where, their primary goal was to make the use of certificates for authentication purposes viable in IoT contexts. Moreover, to offload the computational load to third parties, delegation procedures of protocol primitives have been proposed. Authors in [117] [119] [118] have introduced collaboration for HIP. Their idea is to take advantage of more powerful nodes in the neighborhood of a constrained node to carry heavy computations in a distributed way. Likewise, IKE session establishment delegation to a gateway has been proposed

in [24]. Furthermore, authors in [45] have introduced a delegation procedure that enables a client to delegate certificate validation to a trusted server. While the proposed delegation approaches reduce the computational load at the constrained nodes, they break the end-to-end principle by requiring a third trusted party.

Group key management protocols have traditionally been classified in the literature into three main categories: centralized, distributed, and decentralized [37] [105] [114].

Several approaches have been proposed within the centralized category. In this category, the key management is ensured by a central entity called Key Management Server (KMS). The KMS is a powerful entity that is in charge of rekeying the entire group. To do so, a trusted channel is established between the KMS and the nodes of the group during an initialization phase. This channel is then used to securely rekey the group. Authors in [53] have proposed the Group Key Management Protocol (GKMP). In this protocol, the KMS maintains a Group Key Packet (GKP) that contains a Group Traffic Encryption Key (GTEK) to secure the traffic, and a Group Key Encryption Key (GKEK) to secure the transmission of the GKP. Upon a join event, the KMS uses the old GTEK to distribute the new GKP. However, upon a leave event, the KMS sends the new GKP as a unicast message to each member. This engenders a $O(n)$ complexity, which makes this protocol not scalable to large and dynamic networks. Authors in [132] have introduced an interval-based centralized protocol. The proposed protocol predicts when members might leave the group. In fact, when a member first joins the group, the KMS transmits the required keying materials for the period of time during which the member intends to be part of the group. When the period expires, the member can leave the group without triggering a rekeying event. However, this approach brings several drawbacks. Indeed, predicting the leaving time of members is not realistic and practical for highly dynamic networks. Furthermore, constrained members planning to remain for a long period of time in the group risk to suffer from storage issues. Hence, this protocol is not tailored to

dynamic networks with high number of unpredictable leaving events such as the IoT context.

The Secure Lock protocol introduced in [33] is based on the Chinese Remainder theorem. The basic idea is to rekey the group with a single broadcast instead of peer-to-peer messages. This approach minimizes the number of exchanged messages at the expense of a high computational cost. This cost is due to the Chinese Remainder calculation before each rekeying. Hierarchical based protocols, such as the Logical Key Hierarchy (LKH) protocol [138] and the One-way Function Tree protocol that improves LKH [21], intend to further reduce the rekeying cost (i.e. $OLog(n)$). These protocols are based on a KMS, which shares Key Encryption Keys (KEK) with subgroups of the network. Upon a rekeying event, the KMS uses the shared secret with the subgroups that are unknown to the concerned members to distribute the new TEK. Thus, the number of required rekeying messages is reduced. In brief, centralized protocols take advantage of symmetric algorithms, and avoid peer-to-peer communications within the group. Nevertheless, they still suffer from the single point of failure and the scalability issue.

In distributed protocols, the members collaborate in the rekeying process, and therefore there is no need for a central entity as in centralized protocols. However, peer-to-peer communications are still required between members. Tree-based Group Diffie-Hellman protocol (TGDH) [74], which was later improved by [78] is based on a hierarchical binary tree. Each node of the tree is associated with two types of keys: a secret key and a blinded one (public). TGDH relies on the classical two-party Diffie-Hellman protocol. Hence, the calculation of a non-leaf node secret key is based on the knowledge of the secret key of one child and the blinded key of the other one. In one word, distributed protocols offer the advantage of being highly reliable as they do not rely on a single trusted entity. Nevertheless, full peer to peer communications between the group members are required. In addition, distributed protocols generate a large amount of exchanged messages in addition to the use of complex asymmetric

operations.

Decentralized protocols divide the network into several areas. Each area is associated with a hierarchical level. A KMS is in charge of ensuring the key management process for each area. Traditionally, this category is further classified into two sub categories [37]: the common TEK per area [26] [104], and the independent TEK per area [98] [87]. In the first subcategory, the same TEK is used to secure communications across the different areas of the group. This avoids data translations between the areas. However, when a rekeying event occurs, all group members are affected. Hence, this category is affected by the *1-affects-n* issue. The second category mitigates this issue as each rekeying concerns only the area where a new key needs to be established. As a result, data path is affected. In fact, data passing from an area to another has to be translated at the edge of each area. In [29], authors classifies the decentralized protocols into Time-Driven rekeying subcategory [26] [121] and Membership-Driven rekeying subcategory [104] [22]. In the Time-Driven approach, a rekeying is triggered after the end of each interval of time regardless of membership events. This approach reduces the number of exchanged messages by triggering one rekeying for several events. Nevertheless, a leaving member would be able to communicate until the end of the interval. Similarly, a new joining member would have to wait the beginning of a new interval before being able to access data. In the Membership-Driven subcategory, the group key is changed upon each membership event.

The pervasiness and distributivity of IoT applications make mobility as one of the most important IoT specificities. However, most of the precedent cited approaches do not take into consideration members mobility from an area to another. Instead, mobility is considered as a leave from the source area, and a joining to the destination area. This vision implies a rekeying for both areas. IoT resources scarcity makes this solution not feasible. In fact, few works have been proposed in the literature to efficiently handle mobility in group key management protocols [46] [66]. Indeed, to

reduce the rekeying overhead, these solutions consider that forward secrecy is inherently achieved. Doing so, the number of exchanged messages is reduced by avoiding a rekeying operation in the source area at the expense of forward secrecy violation. In addition, a list that handles the mobile members is generally implemented in the Key Management Servers. In large and highly dynamic networks such as IoT, maintaining a list of moving members might quickly become highly complex to manage.

3.4 Conclusion

In this chapter, we clarified the different security concepts and protocols involved in IoT. In particular, we highlighted the differences between data security, data privacy, and data confidentiality. Furthermore, we focused on the importance of key management protocols in the establishment of secured communications. In addition, we highlighted the main security properties that need to be ensured in key management protocols. Besides, we presented e-health applications that have been used in some of our contributions as a use case scenario. Indeed, we focused on the sensitivity and privacy of the generated data in such applications. Finally, we introduced a detailed analysis of the existing solutions in the literature with respect to their suitability to IoT constraints.

Part II

Contributions

Chapter 4

Centralized approaches

This chapter is divided into two parts. In the first part, we introduce our first protocol entitled "A Lightweight Key Management Protocol for E-health Applications" [10]. This protocol aims to establish a secure communication channel between nodes (i.e. sensors) and a gateway (i.e. base station) in the context of an e-health application. In the second part, we introduce our second protocol, which, in fact, is a tailoring of the standard based Mikey-Ticket protocol. In this second contribution entitled "Lightweighted and Energy-Aware Mikey-Ticket For E-Health Applications" [11], we aim to establish a secure channel between nodes and a remote entity. Both contributions aim to propose lightweight protocols based on a central entity. In the following, we present and validate both protocols.

4.1 A Lightweight Key Management Protocol for E-health Applications

In this contribution, we propose a new key management protocol to establish a secure channel between the different nodes and the base station. Our solution is based on a lightweight PKI that is only used at the registration phase to establish a symmetric session key. Using this key, our protocol encrypts the exchanged data to ensure confidentiality. Furthermore, to ensure authentication, it computes a Message Au-

thentication Code (MAC) using the same key, and adds timestamps to prevent replay attacks. In order to assess our protocol, we conduct a formal validation regarding security properties. In addition, we evaluate both communication and computational costs to highlight energy savings. We compare the energy consumption of our protocol to other protocols, such as simplified SSL protocol and simplified Kerberos protocol. The results show that our protocol is less energy consuming where its security properties are kept safe. In the following, we highlight the main features of our protocol:

- avoids the use of public key certificates, which are highly resource consuming for constrained nodes [45]. In fact, we introduce additional assumptions regarding the public key of the constrained nodes to authenticate the exchanged messages without using public key certficates.

- offloads the generation of the security credentials to more powerful nodes (i.e. base station). This spares the constrained nodes from processing the pseudo random generation function during each rekeying operation.

- uses Elliptic Curve Cryptography (ECC) [129] as the asymmetric cryptographic protocol instead of RSA. Doing so, energy savings are achieved while providing the same level of security.

- is not specific to e-health applications. Indeed, our protocol can be implemented for a wide range of IoT applications. In addition, our protocol does not alter any cryptographic primitive allowing a high level of backward compatibility. As a result, any existing standard based cryptographic algorithm might easily be integrated.

4.1.1 The proposed protocol

In this section, we present our protocol to establish a secure channel between nodes planted in human bodies and the base station of an e-health system. This protocol ensures key exchange with minimal resource consumption. At first, we describe the

Figure 4-1: High level architecture

general architecture of our e-health system. Then, we define the assumptions of our network scenario, and finally we present the functioning details. Table 4.1 summarizes the different used notations.

4.1.1.1 High level architecture

The high level architecture is mainly composed of three components: mobile and contextual sensor nodes, a base station and a back-end infrastructure as shown in Figure 4-1. The system enables health-related data to be collected from sensor nodes planted in, or on the body plus contextual sensors that gather data like room temperature or humidity level. These data are transmitted to the BS using wireless interfaces (e.g., Bluetooth or ZigBee). Then, the BS transmits the gathered data to caregivers and family members using the back-end infrastructure through an Internet connection (e.g., Wifi, Edge or 3G). Cloud computing could play the role of the back-end infrastructure as it enables convenient on demand access to a shared group of resources. Data management and visualization of health related data are then improved. We could use the personal user's smartphone as the BS because it is always close to the user either at home or outside. This will spare the user from carrying an inconvenient device. Additionally, energy consumption is decreased as nodes will avoid transmitting data over long range radio transmission.

4.1.1.2 Assumptions

We set the following assumptions before presenting the details of our protocol:

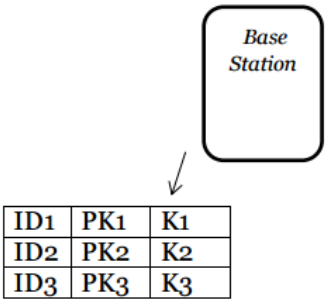

Figure 4-2: Binding table

Notation	Description
ID	The identity of a node
BS	Base Station
PK	Public Key
K	Generated symmetric key
TS	Timestamp
MAC	Message Authentication Code

Table 4.1: Terminology Table

- Each node has an ID, a PK and a private key that is kept secret.
- Only the BS knows the PK of each node.
- The BS has more computational and energy capabilities than nodes, and thus is able to perform classical public key operations to secure data transmission with remote servers.
- The nodes have the ability to perform symmetric and asymmetric encryption.
- The BS keeps a table that binds each node ID with the corresponding symmetric key K and its Public Key PK (Figure 4-2).
- An offline dealer inserts the ID and the PK of each new registered node into BS binding table.
- We consider that our network architecture is clear from interferences.
- The BS is considered as a trusted entity.

4.1.1.3 Functioning

Our protocol is divided into two phases: a registration phase and a key establishment phase. The first one aims to register the nodes before launching the key exchange. In the second one, the registered node requests the BS to establish a new symmetric key. In addition to these two phases, our protocol handles key updates and nodes joining. We present the functioning of each phase, then we focus on the description of key updates and nodes joining.

4.1.1.3.1 Registration phase: during this phase, the different nodes are registered with the BS. The registration is proceeded by an offline dealer that inserts the ID and the PK of each node (Figure 4-2). Only the BS knows the PK of each node to ensure BS authenticity during the key establishment phase.

4.1.1.3.2 Key establishment phase: in this phase, the aim is to establish a symmetric key K between the BS and each node. This key will be added to the binding table, and will be used to secure the communication channel. To this end, each node willing to exchange data with the BS has to perform the following steps (Figure 4-3):

- The node sends a request that contains its ID plus a TS to ensure freshness and avoid replay attacks. The BS uses the ID to retrieve the corresponding PK of the node. This latter embeds the message *REQUEST_MESSAGE* and sends it to the BS using the underlying protocol.

- Upon reception of *REQUEST_MESSAGE*, the BS checks the validity of the ID, then, it generates a random key K that will be added to the line corresponding to the node ID in the binding table. The BS embeds K in the *REQUEST_RESPONSE* message along with a TS. Afterward, it encrypts this

message with the node's PK. The encryption is performed using an Elliptic Curve Cryptography (ECC) algorithm that is more energy efficient keeping the same security threshold as the conventional algorithms such as RSA [71]. The encrypted message is sent to the corresponding node.

- The node decrypts *REQUEST_RESPONSE* using its private key and checks its freshness. Next, it retrieves the symmetric key K, which is used to secure patient related data based on symmetric encryption (e.g. AES [103]). In addition, K is involved to compute a MAC. This latter is inserted in the exchanged messages to ensure integrity.
- To prove the correct reception of K, the node sends a *VERIFICATION_MESSAGE*, which contains its ID along with the received TS. Using K, the node encrypts the message and appends it with a MAC for authenticity. In fact, if the verification of the joined MAC fails, this means that an attacker has succeeded to alter the message. Otherwise, upon successful verification, the BS and the node agree on using K to secure their future communications.

4.1.1.3.3 Key update: we should keep in mind that an e-health system can be running for months. The system is then vulnerable to long term attacks that analyze the encrypted traffic over the network to discover encryption keys. In our protocol, we propose to perform periodic key updates to deal with this issue. To establish the updates frequency, a trade-off between efficiency and the level of protection has to be found. In each update, our protocol only requires the execution of the key establishment phase. In fact, each node has to send a new request to renew its symmetric key.

4.1.1.3.4 Node joining: in case where a new node joins the network (we can imagine a physician asksing the patient to record a new health related value), an offline dealer has to load its ID and its PK into the binding table of the BS. The node

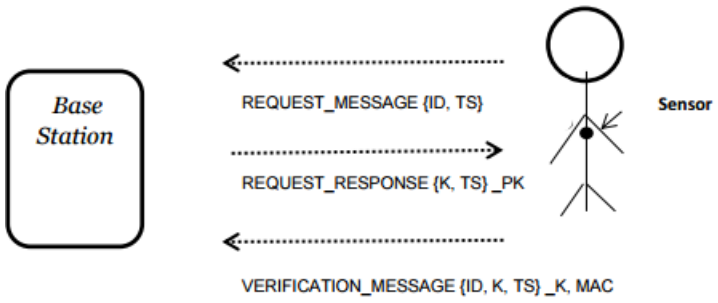

Figure 4-3: Message exchanges during the key establishment phase

is then able to launch the key establishment phase.

4.1.2 Analysis

In this section, we provide a detailed analysis of our protocol both in terms of security properties and energy consumption. Firstly, we propose a theoretical analysis regarding security properties. Our analysis is then validated using an automated validation tool called Avispa Tool [91], which is based on formal models. Secondly, we focus on the energy cost savings. We use energy models to estimate the total energy cost composed of both computational and communication costs. We compare our results with the energy costs of simplified SSL protocol [15], simplified Kerberos protocol [76] and the proposed Trust Key Management Scheme for Wireless Body Area Networks [82].

4.1.2.1 Security properties

We provide an analysis of security features provided by our protocol based on the properties presented in [112]. We have added an analysis concerning integrity and confidentiality as we consider them being critical in an e-health application.

- *Confidentiality:* the exchanged data between the nodes and the BS are kept secret in our protocol. This feature is ensured by using symmetric encryption as it is less resource consuming than an asymmetric one. In addition, our

protocol performs periodic key updates to enforce confidentiality and prevent long-term attacks.

- *Integrity:* in our protocol, we ensure that the captured data transmitted to the BS are not altered using a MAC. This latter is computed and appended to each exchanged message.

- *Distribution:* the distribution of the credentials is performed through two steps. The first step is the registration phase where the credentials are distributed in an offline mode. This step is performed only once in a node lifetime, whereas the second phase is done in an online mode allowing frequent key updates in an automatic way.

- *Authentication:* both nodes and BS authenticate each other, either in the key establishment process or during data exchanges. In the first case, using the node's PK, the BS encrypts the message containing K and sends it to the node which makes it impossible for another entity to masquerade as the BS. The node, in turn, is authenticated by the fact that it is the only entity that detains its own private key, and thus can decrypt the received message from the BS. In the second case, a MAC based on the generated symmetric key is added to the transmitted messages to ensure authentication during data exchanges.

- *Overhead:* our protocol limits the communication overhead as the handshake only includes two exchanges between the node and the BS (i.e., a request sent from the node to the BS and a response sent from the BS to the node). The computational overhead is also very limited as the node has to perform a lightweight PKI scheme that includes only one ECC decryption.

- *Resilience:* the resilience of our protocol is high. In fact, the loss of a node and thus its key affects only the corresponding node as each node stores only its private key. The BS maintains a different symmetric key for each node in the binding table.

- *Extensibility and scalability:* our protocol is extensible as it allows new nodes to be integrated into the system. Also, its scalability is high, as no operation is required on the nodes in case of a new joining. However, the ID and PK of new nodes have to be added into the BS by an offline dealer.

Several techniques have been introduced to model and formally validate a security protocol regarding its properties. Model checking [35] is one of the formal methods used to validate finite-state-concurrent systems such as communication protocols. It usually involves verification tools to exhaustively search all possible execution sequences for desired properties in a protocol specification. Many security protocols have been validated through model checking [130] [52], and several validation tools are based on model checking [1] [4] [3]. We highlight some advantages of model checking compared to classical approaches, which are developed around simulation, testing, and deductions:

- Gives the possibility to the users to check every single step of the execution process, allowing them to detect any malfunction in a highly accurate way. However, using simulation or testing, only a broad overview of the protocol behaviour is provided. In addition, some flaws might remain unfound until the protocol's production stage is initiated.

- Allows prompt and automated verifications through different tools that implement model checking. In fact, by adopting model checking, users can avoid prototyping their protocols.

AVISPA (Automated Validation of Internet Security Protocol and Applications) is a state-of-the-art verification tool for security protocols that includes a set of model checkers with a common front end. The tool follows the Dolev-Yao intruder model [39] to intercept messages or to insert modified data. It performs analytical rules to state whether the protocol is safe or not. In case of unsafety, the tool provides a trace highlighting the steps that led to the attack. In fact, Avispa is considered as an effective tool for the analysis of different Internet security protocols and applications. In

the literature, several security protocols have been validated throught Avispa [34] [83] [30] [115]. Moreover, the security protocols standardized by the Internet Engineering Task Force (IETF) have been analyzed by the AVISPA community (e.g. IKE, TLS, AAA), and some of the protocols have been found to be flawed [91] [1].

We carry out the formal validation of our protocol using Avispa tool to prove that it does not violate the required security properties, in particular, confidentiality, authentication, delivery proof and replay protection. Protocol models in Avispa are written in a role-based language called High Level Protocol Specification Language, or HLPSL [32]. The actions of the different entities are specified in a module called *basic role*, while their interactions are defined by composing multiple *basic roles* together into a *composed role*. In addition, the security goals of the analyzed protocol are specified in the *goal section* before launching the analysis. Besides, Avispa uses four different automatic protocol analysis techniques to validate the analyzed protocol against the specified security goals: on-the-fly model-checker (OFMC), constraint-logic based attack searcher (CL-AtSe), SAT-based model checker (SATMC), and tree automata based on automatic approximations for the analysis of security protocols (TA4SP) [91].

In our modeling, we have first specified a *basic role* to describe the actions of the different entities involved. Then, we have specified how the participants interact with each other in a *composed role*. For clarity reasons, we present our modeling using Alice-Bob ($A - B$) notation.

- $A- > B : \{ID, TS\}$
- $B- > A : \{K, TS\}_{PK_A}$
- $A- > B : \{ID, K, TS\}_K, MAC$

Where:

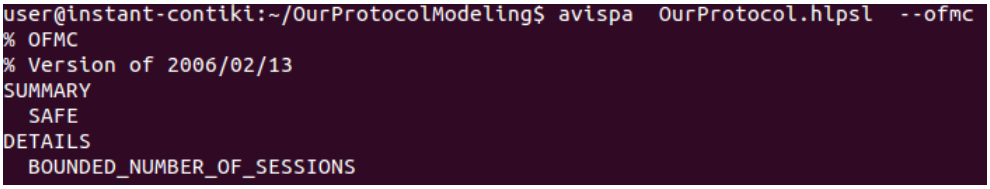

Figure 4-4: Avispa output (OFMC)

- *A: Sensor node*

- *B: Base Station*

Upon complete modeling of our protocol, we have checked its correctness using a protocol animation tool called SPAN [47] that has been introduced to help protocol developers in writing Avispa specifications. The security goals were subsequently evaluated by executing the four Avispa's backends (i.e. OFMC, $CL-AtSe$, SATMC and TA4SP). Besides, we have used the default Dolev-Yao intruder model, which allows to simulate an intruder that has full control over the network. All messages sent and received by the different entities might be intercepted, analyzed, modified (as far as the keys are known), or sent to other entities.

The results of the validation were indicated in reports for each back-end model produced by Avispa tool. These reports show that our protocol is "SAFE" against OFMC (Figure 4-4), $CL-AtSe$ (Figure 4-5) and SATMC (Figure 4-6). However, against TA4SP database, the result was "INCONCLUSIVE". According to Avispa user manual [76], an inconclusive result does not imply that an attack has been detected (Figure 4-7). Consequently, based on the obtained results, we can safely affirm that our protocol is safe regarding the specified security goals. It is impossible for an attacker to violate any of the specified security properties and disrupt its functioning.

Following our formal validation, we focus in the next section on the energy cost savings achieved through our lightweight key management protocol. The results are then compared to other existing protocols.

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl --cl-atse

SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
```

Figure 4-5: Avispa output ($CL - AtSe$)

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl  --satmc
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
```

Figure 4-6: Avispa output (SATMC)

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl  --ta4sp
SUMMARY
   INCONCLUSIVE
```

Figure 4-7: Avispa output (TA4SP)

4.1.2.2 Energy cost analysis

As explained above, our contribution focuses on establishing a secured channel between the constrained nodes (i.e. sensors) and the BS in e-health applications. To this end, we propose a new key management protocol based on a lightweight PKI. In this section, we provide a performance analysis of our protocol, and compare energy consumption with existing protocols, such as simplified SSL protocol [15] and simplified Kerberos protocol [76]. First, we describe the energy model on which our estimations are based. Then, we evaluate the communication and computational costs. The analysis is concluded with a discussion of the total energy cost highlighting the obtained energy savings.

4.1.2.2.1 Energy model: authors, in [89], have presented an energy evaluation of Wireless Sensor Nodes (WSN) regarding the communication cost. This latter is composed of the costs of transmission, reception and listening. Besides, the energy consumption of ECC encryption algorithm has also been assessed. Both implementations were processed on tiny nodes (i.e. TelosB) with few MHz of computational power, several kilobytes of RAM and several tens of kilobytes of ROM.

In our evaluation, we consider the total energy cost as the sum of the communication cost and the computational cost. Based on the energy measurements presented in [89], we estimate the energy consumption of tiny nodes regarding both communication and computational aspect. The deduced values, summarized in Table 4.2, are used as an energy model of the different operations on constrained nodes. We consider transmission, reception, listening, and cryptographic operation costs for the evaluation of the overall energy cost.

We set the following assumptions with respect to our evaluation:

- Our evaluation only covers energy consumption of the constrained nodes as the

Operation	Cost
Transmit 1 bit	0.72 uJ
Receive 1 bit	0.81 uJ
Listen for 1 ms	0.29 uJ
ECC-160 point mult	17 mJ

Table 4.2: Estimated energy costs on constrained nodes (TelosB)

	Listening	Sending	Receiving	Cryptography	Total
Energy cost	45.02 uJ	97.92 uJ	311.04 uJ	17 mJ	17.44 mJ

Table 4.3: Total energy cost of the constrained node (TelosB)

base station is not limited by resource scarcity. Hence, the efforts of reducing energy consumption are focused on the constrained part of our network model.

- To estimate the size of messages, we assume the maximum size of nodes ID and messages headers of the protocols involved in our analysis (i.e. [15] [76] [82]). Hence, 1 byte and 12 bytes are considered for nodes ID and protocol headers respectively. Four bytes are required to generate timestamps (5 bits for the year, 4 bits for month, 5 bits for the day, 5 bits for the hour, 6 bits for minutes and 6 bits for seconds). In addition, as we advocate the use of AES for encryption [103], 256 bits are considered for the generated key.

4.1.2.2.2 Energy consumption First, we present the energy evaluation of our protocol regarding the communication cost, then we focus on the computational cost. The communication cost is the sum of the costs of transmission, reception and listening. Both transmission and reception costs are estimated based on packet size and the values defined in the used energy model (Table 4.2). As described in our protocol, a node has to send a request to the BS containing its ID (1 byte), a timestamp (4 bytes) plus 12 bytes of protocol header. Thus, the size of the sent packet is 17 bytes. In addition, the node receives a response from the BS containing the generated key (256 bits), a timestamp (4 bytes) plus 12 bytes of protocol headers. The maximum size of the received message is 48 bytes. The results are depicted in Table 4.3. Fur-

thermore, listening time is included in our evaluation as a component of the total communication cost. We consider the constrained node (CN) listening for a period of time equal to the sum of packets propagation delay (Δ), packets computation time (Comp) and transmission latency (T). We also assume the BS being at one hop from the constrained node. Moreover, we consider the BS being 100 times more powerful than the tiny node for the estimation of computational time. Furthermore, for the estimation of communication latency, an effective data rate of 75 *kbps* for nodes (e.g. TelosB) is taken into account [89]. Listening time is thus computed as follows:

$T_{listening} = Comp(BS) + T(BS) + \Delta(BS- > CN)$.

Where:

- *Comp(BS): Computational time of BS*

- *T(BS): Transmission latency of BS*

- $\Delta(BS- > CN)$*: Packets propagation delay from BS to CN*

Based on messages size and our energy model, we estimate the computational cost. This latter is computed using the cost of ECC algorithm. The results of our evaluation considering both communication and computational aspect are depicted in Table 4.3.

4.1.2.2.3 Discussion & comparison Firstly, our discussion focuses on the estimation of the overall energy cost of our protocol. We have chosen to compare our results with simplified SSL protocol [15], simplified Kerberos protocol [76] and the proposed Trust Key Management Scheme for Wireless Body Area Networks [82]. Indeed, these protocols are widely used for key management in WSN [102]. Secondly, we have conducted an energy evaluation of several rekeying operations.

Upon energy cost evaluation regarding both communication and computational aspects, we have compared the overall energy cost considering our protocol, simplified SSL protocol [15], simplified Kerberos protocol [76] and the proposed Trust Key

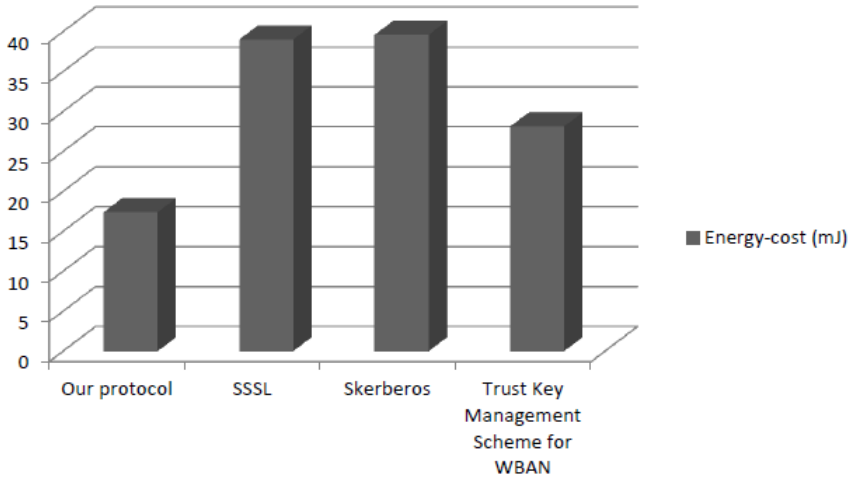

Figure 4-8: Energy cost analysis

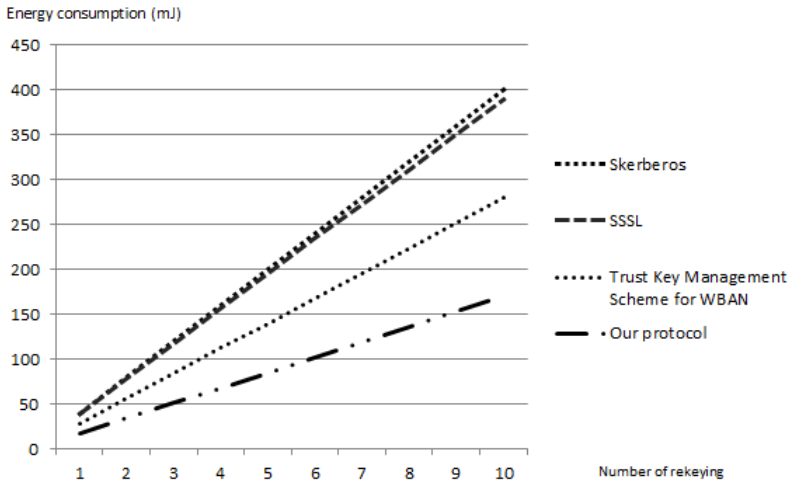

Figure 4-9: Energy consumption evolution through several rekeying operations

Management Scheme for Wireless Body Area Networks [82]. The results, synthesized in Figure 4-8, show that our protocol consumes less energy than the other protocols. In fact, the total energy cost of our protocol is equal to 17.44 mJ whereas the energy costs of simplified SSL protocol, simplified Kerberos protocol and the proposed Trust Key Management Scheme for Wireless Body Area Networks are respectively 39 mJ, 39.6-47.6 mJ and 28.13 mJ.

We have compared the different energy costs regarding several rekeying operations (see Figure 4-9). Frequent updates are likely to be performed in order to avoid long term attacks. The results indicate that our protocol shows a considerable gain in energy consumption that increases with the increase of rekeying operations, which is critical for tiny nodes with highly constrained resources (e.g. increasing battery lifetime). In addition, energy savings are greater, compared to the other protocols.

The analysis allows us to safely affirm that our lightweight protocol is more energy saving while it offers a high level of security as confidentiality, integrity and authentication are provided. As a consequence, our protocol is suitable to be applied in e-health applications in a resource-constrained Internet of Things environment.

4.1.3 Conclusion

In this part, we introduced a new key management protocol for e-health applications in the context of Internet of Things based on a lightweight Public Key Infrastructure. In our protocol, Public Key primitives are only involved at the registration phase to establish a symmetric session key. This latter will be used to establish a secure channel between the different nodes and the base station. We evaluated our protocol regarding both security features and energy cost savings. The results show that the security properties of our protocol are kept safe. Moreover, in comparison with some existing protocols such as simplified SSL protocol and simplified Kerberos, our protocol has lower energy costs of communication and computation. It is then suitable to be applied in e-health applications deployed in a resource-constrained environment.

Nevertheless, this protocol is limited to the establishment of a shared secret between a node and a gateway. Indeed, nodes will surely take part in communications with remote entities. In addition, the proposed protocol uses asymmetric primitives, which remain highly energy consuming for constrained entities. To address these issues, we propose in the next part our second protocol. This latter is based on the tailoring of Mikey-Ticket key management protocol. In fact, this protocol only uses symmetric promitives and allows the nodes to establish a secure channel with remote entities.

4.2 Lightweighted and Energy-Aware Mikey-Ticket For E-Health Applications

Mikey-Ticket protocol needs to be tailored for constrained environments in order to be adapted to the resources constraints of such environments. To this end, we introduce, in this contribution, two solutions to tailor Mikey-Ticket to e-health environments without weakening its security properties. In the first solution, we propose a new 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) header compression scheme for Mikey-Ticket. Our scheme is intended to save energy and avoid 6LoWPAN fragmentation that may occur when a datagram size exceeds the link layer MTU [1]. Indeed, fragmentation is undesirable, as 6LoWPAN is vulnerable to fragmentation attacks [59]. In the second solution, we propose a new exchange mode to reduce the number of exchanged messages from six to four. The main concern being to reduce the involvement of the constrained nodes in the exchange process.

4.2.1 Background

4.2.1.1 Mikey-Ticket choice

In this section, we focus on the motivations that are behind our choice of Mikey-Ticket over other existing protocols, particularly IKE. This latter is a key exchange protocol that aims to perform mutual authentication and to provide Security Associations

[1] Maximum Transmission Unit of the IEEE 802.15.4 protocol

(SAs) to be used as input for IPsec. Indeed, securing IoT communications at the IP level is likely to be achieved through the use of IPsec [72]. In this way, Mikey-Ticket aims to achieve the same goal. In our study, we have focused on Mikey-Ticket instead of the widely adopted IKE. The reasons behind this choice are the following.

- The proposed e-health constrained scenario involves the use of tiny nodes that are highly limited by their computational capabilities. In fact, during its first request/response exchange process (i.e.IKE_SA_INIT), IKE involves the two parties in a Diffie-Hellman instantiation phase, which requires an important energy consumption due to exponential operations. Indeed, Public key operations are not suitable for highly constrained environements. Besides, the Pre-Shared mode of Mikey-Ticket only involves symmetric operations, which are much more energy saving compared to asymmetric approaches [133]. Furthermore, Mikey-Ticket is designed to involve a central trusted entity which makes it more suitable to our network scenario compared to IKE. The trusted entity has a double role to play. Firstly, it acts as a gateway (i.e. 6LoWPAN Border Router) through which 6loWPAN headers are compressed and decompressed. Secondly, it spares the constrained node from using public key cryptography by generating and distributing the required security credentials.

- Mikey-Ticket is a product of the IETF (Internet Engineering Task Force) such as IKE [71], DTLS[43] and other standard based protocols. In fact, our approach to address data confidentiality in IoT applications aims to propose new extensions to standardized protocols in order to adapt them to the IoT context. Following this approach, Mikey-Ticket sounds to be the adequate protocol that can be extended to ensure secure communications in IoT.

- A lot of efforts have been carried out by the research community to optimize the IKE protocol. As IoT is only in its first stages of deployment, the protocol suite that should be implemented to secure IoT based applications is not clear yet. Our research effort attempts, therefore, to bring a contribution in this process of adapting and selecting existing protocols for IoT environments.

Table 4.4: Terminology Table

Notation	Description
I	Initiator
R	Responder
KMS	Key Management Server
X_{ID}	The Identity of X
N_X	Nonce generated by X
$K_{X,Y}$	Shared key between X and Y
$aK_{X,Y}$	Shared authentication key between X and Y
$eK_{X,Y}$	Shared encryption key between X and Y
$[data]_K$	Data encrypted with the key K
$Ticket$	Object used to identify and deliver keys

4.2.1.2 Mikey-Ticket overview

Mikey-Ticket [84] is a key distribution protocol designed to enhance the Multimedia Internet KEYing protocol (Mikey) [18]. It defines new modes of key distribution which are well adapted to centralized based scenarios where a third trusted entity is available. Mikey-Ticket considers two entities that aim to establish a shared secret. One of the two entities assumes the Initiator role whereas the second one assumes the Responder role. The key establishment relies on a Key Management Server to generate and deliver the needed credentials. Such design spares the peers from a pre-distribution phase that would require credentials storing. Instead, peers can request such credentials only when required. In this work, we only consider the Pre-Shared Key mode (PSK) of Mikey-Ticket as the Public Key (PK) mode and the Diffie-Hellman key exchange mode are ruled out due to their inadequacy with IoT constrained environments.

We provide a brief description of Mikey-Ticket message exchanges and the general Mikey header (HDR) format. Table 4.4 summarizes the used notations.

4.2.1.2.1 Message exchanges Mikey-Ticket uses six messages to establish a new key between the Initiator I and the Responder R (see Figure 4-10). The protocol relies on the Key Management Server (KMS) which delivers the generated key. The

Initiator and the Responder do not share any credentials. Instead, they share a secret master key with the *KMS*. This key is used to derive an authentication key and an encryption key. The generated keys are used to secure the communication for I and R whereas *KMS* provides data authenticity, data integrity and confidentiality.

We briefly describe the content of each exchanged message of the full three round-trip Mikey-Ticket mode :

REQUEST_INIT: through this message, node I expresses its willingness to establish a shared key with node R. The message contains information about the responder's identity. To ensure authenticity, a Message Authentication Code (MAC) computed with $aK_{I,KMS}$ is included.

REQUEST_RESP: after successful verification, the request is authorized and the *KMS* generates the requested key K and encodes it in a ticket. The message is sent to I.

TRANSFER_INIT: upon reception of REQUEST_RESP message, node I derives an authentication key aK and an encryption key eK to secure data transmission between I and R. Then, node I transfers the ticket to R through TRANSFER_INIT message. Also, a MAC is computed using aK and included in the message.

RESOLVE_INIT: through this message, node R asks the *KMS* to return the key K encoded in the ticket. The message is protected by a MAC based on $aK_{KMS,R}$.

RESOLVE_RESP: if node R is authorized to receive the generated key encoded in the ticket, the *KMS* sends RESOLVE_RESP message that includes the generated key K. The message is protected through encryption and a MAC message based on $aK_{KMS,R}$.

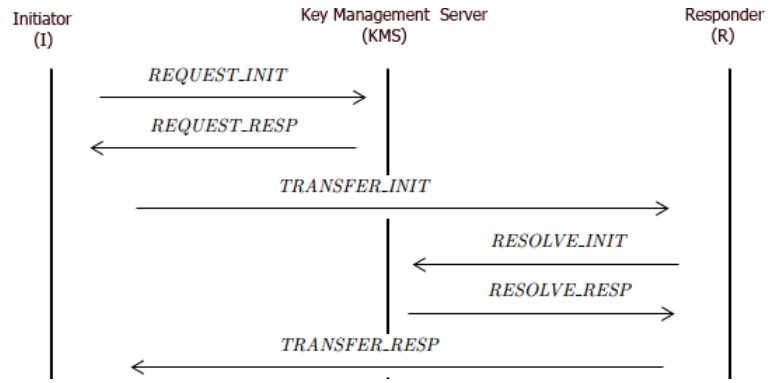

Figure 4-10: Mikey-Ticket full three round-trip mode exchange (RFC 6043)

TRANSFER_ RESP: R is in possession of the generated key K. TRANSFER_INIT's MAC can thus be checked. The exchange is concluded through TRANSFER_RESP message to prove the correct reception and derivation of the generated session key. It is worth noticing that the different messages contain a nonce for protection against replay attacks.

Figure 4-10 depicts the signaling for the full three round-trip Mikey-Ticket mode. Nevertheless, RFC 6043 [84] introduces four different modes according to the specificities of both the Initiator and the Responder. Mode 1 represents actually the full three round-trip mode where only the KMS is in charge of generating, deriving and distributing the keying materials. Both I and R have to request/resolve messages with the KMS. In mode 2, the exchanges between the KMS and R are omitted (i.e. $RESOLVE\_INIT$ and $RESOLVE\_RESP$). However, R has to be able to resolve the ticket without assistance from the KMS. In mode 3, the *ticket request* exchange (i.e. $REQUEST\_INIT$ and $REQUEST\_RESP$) can be omitted if I is able to create the keying materials without an assistance from KMS. Mode 4 only contains a *ticket transfer* exchange (i.e. $TRANSFER\_INIT$). However, it requires from I and R to share security credentials prior to the start of the procotol session.

4.2.1.2.2 Common Header Format (HDR) The Common Header payload (Figure 4-11) contains information about the different exchanged messages. It is

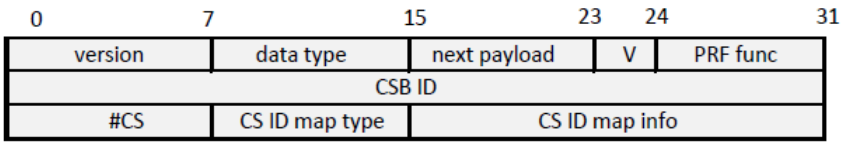

Figure 4-11: Mikey Common Header Format (RFC 3830)

always present as the first payload in each message. In the following, we present a succinct description of each field contained in the Mikey_Ticket header. We refer to RFC3038 [18] and RFC6043 [84] for a more detailed description:

- *Version (8 bits):* version of MIKEY.
- *Data type (8 bits):* type of the exchanged message.
- *Next Payload (8 bits):* identifies the payload added after the current payload.
- *V (1 bit):* flag to indicate the use of a verification message.
- *PRF func (7 bits):* indicates the key derivation function.
- *CSB ID (32 bits):* crypto Session Bundle (CSB) is a collection of one or more Crypto Sessions (CS). CSB ID field identifies the CSB.
- ♯ *CS (8 bits):* a Crypto Session refers to a data steam protected by a single instance of a security protocol. ♯ CS field indicates the number of Crypto Sessions within the CBS.
- *CS ID map type (8 bits):* specifies the method of uniquely mapping crypto sessions to the security protocol sessions.
- *CS ID map info (variable length)* identifies and maps crypto sessions to the security protocol sessions.

4.2.1.3 6LoWPAN Adaptation Layer

The 6LoWPAN standard defined in [57] aims to transfer IPv6 packets to IEEE 802.15.4 based networks. 6LoWPAN uses IPV6 header compression mechanisms of

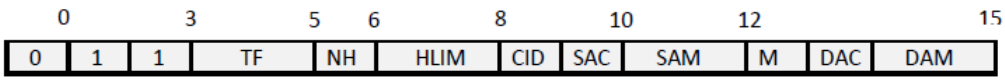

Figure 4-12: IPHC

IPv6 datagrams. Compression mechanisms are motivated by the limited space available in 802.15.4 frames to encapsulate IPv6 packets. In fact, the size of the 802.15.4 frame payload (102 bytes) leaves limited space for an IPv6 packet as 48 bytes are required only for its header. 6LoWPAN defines encoding formats for compression based on shared state within contexts. In other words, it takes advantage of the fields that are implicitly known to all nodes in the network or can be deduced from the MAC layer. The compression scheme consists of IP Header Compression (IPHC) and Next Header Compression (NHC).

IPHC encoding describes how an IPv6 header is compressed. As depicted in Figure 4-12, 13 bits of the 2 bytes long IPHC are used for compression. The IPv6 header fields that are not compressed are placed immediately after IPHC. Moreover, NH field in IPHC indicates whether the following header is encoded using NHC. If so, NHC encoding follows immediately the compressed IPv6 header. Compression formats for different next headers are identified by a variable ID bits plus the specific header compression encoding bits. The NHC to encode IPv6 extension headers and UDP header are already defined. For more details on 6LoWPAN, we refer the reader to RFC 6282 [57].

4.2.2 Network scenario

We consider a scenario of an e-health application where smart objects (contextual sensors), gateways and remote entities are used (see Figure 4-13). IP-enabled smart objects are in charge of sensing health related data (e.g. blood pressure, blood glucose level, temperature level, etc.). They are planted in the human body. Gateways

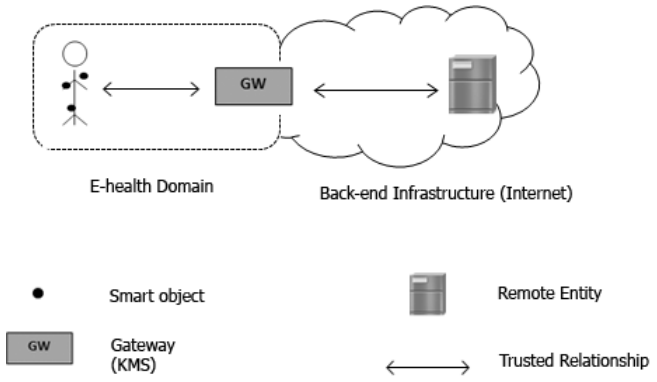

Figure 4-13: Network Scenario

connect these objects to a backend infrastructure such as Internet. It is worth mentioning that user's smartphones could be used as gateways. Remote entities are in charge of processing and analyzing the received data.

Smart objects have limited computational power, memory and energy resources, whereas gateways are much less resource constrained and are comparable to standard routers. Remote entities can take the form of a server hardware or being distributed in a Cloud infrastructure with dynamic resources.

The mapping with Mikey-Ticket concepts is defined as follows:

- *Initiator*: smart object (e.g. IP-enabled tiny sensor).
- *Key Management Server*: gateway (e.g. smartphone).
- *Responder* : remote entity (e.g. servers disposed in hospitals automatically trigger an exchange in order to check on a patient's vital signs).

Securing e-health applications relies on efficient key management schemes that ensure reliable key distribution. We do believe that the best approach to tackle security challenges in the evolving IoT is to focus our efforts on standard based protocols. We have chosen Mikey-Ticket for its simplicity and its adaptation to centralized scenarios which suits well our e-health application. However, current key management

protocols such as Mikey-Ticket were designed to be used in an unconstrained environment which does not take into consideration resources limitation. In the next section, we present in detail our contribution to make Mikey-Ticket more lightweight while preserving its security properties.

4.2.3 Reducing the overhead of Mikey-Ticket

In order to reduce the communication overhead of Mikey-Ticket protocol when implemented on constrained entities, we have adopted two complementary approaches. Firstly, we have reduced the size of the exchanged messages by proposing a new header compression scheme. Secondly, we have minimized the number of exchanged control messages by proposing a new exchange mode.

4.2.3.1 New header compression scheme

In this section, we describe our proposed 6LoWPAN header compression scheme for Mikey-Ticket. Our compression is based on the fact that the fields which are implicitly known to all entities in the network or those that can be deduced from the MAC layer can be removed. As explained in section 4.2.1.3, the NHC is used to encode the IPv6 extension headers and UDP header. Nevertheless, despite 6LoWPAN has defined header compression for UDP, no NHC compression is defined in case where headers contained in UDP payloads are compressed. In fact, Mikey-Ticket common header is contained in the UDP payload. Therefore, we propose to use the 6LoWPAN extension proposed in [108] to extend 6LoWPAN header compression mechanisms. These extensions indicate that the headers of protocols that are part of the UDP payload are compressed with 6LoWPAN-NHC.

Mikey-Ticket common header is 12 bytes long. It is appended to each packet through the different exchanged messages. We propose a 6LoWPAN-NHC to compress Mikey-Ticket header called 6LoWPAN-NHC-HDR. The proposed approach allows to reduce the header length from 12 bytes to 3 bytes (2 bytes for our 6LoWPAN-

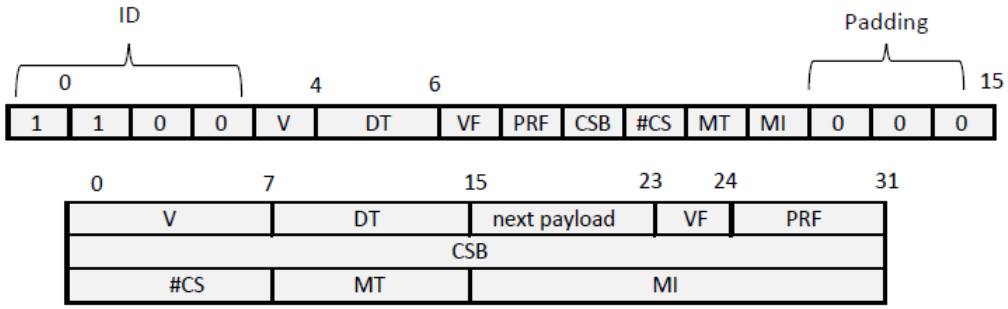

Figure 4-14: Our 6LoWPAN-NHC-HDR encoding compared to the basic Mikey's header

NHC-HDR plus 1 byte for the Next Payload field that is always carried inline) in the best compression case. In fact, only 13 bits are required to encode the different fields. Nevertheless, in order to remain standard compliant (i.e. the size of NHC encodings is multiple of bytes), our 6LoWPAN-NHC-HDR is 2 bytes long. In addition, to comply with 6LoWPAN-NHC encoding schemes, the first four bits implement an ID field to uniquely identify our NHC encoding. We set the ID bits to 1100. To the best of our knowledge, the 1100 bits are currently unused as NHC identifiers. In the following, we present in detail the encoding approach for each field (see Table 4.5 and Figure

- *Version (V):* if 0, the version is the default and latest Mikey-Ticket version defined in [18] and the field is skipped. If future versions are defined, the bit is set to 1 and the version number is carried inline after the 6LoWPAN-NHC-HDR header. Our compression is thus kept dynamic and flexible.

- *Data type (DT):* the data type field describes the type of the exchanged messages. Based on our new exchange mode, we only consider three types of messages (i.e. REQUEST_INIT, REQUEST_RESPONSE, TRANSFER_END) plus the ERROR type. In addition, only the Pre-Shared Key (PSK) mode is considered. The other modes are ruled out due to their inadequacy with our constrained network scenario. Doing so, we are then able to use just 2 bits encoding for the data type field instead of 8 bits in the original version:

00 : REQUEST_INIT

01 : REQUEST_RESPONSE

10 : TRANSFER_END

11 : ERROR

- *Verification V (VF):* the VF field encoding is similar to the non-compressed header. If it is set to 0, no verification message is used. When it is set to 1, a verification message is required.

- *PRF func (PRF):* if 0, the default PRF function defined in [18] is used. If set to 1, the PRF function value is carried inline.

- *CSB ID (CSB):* the CSB ID is chosen by the Initiator and needs to be unique between each Initiator-Responder pair. Instead of carrying its 32 bits size inline, we propose to derivate the CSB ID from the concatenation of lower layer identifiers (e.g. IPv6 addresses). One bit is sufficient for the encoding. If set to 0, the CSB ID is derived instead of being carried inline. If set to 1, the 32 bits CSB ID are carried after the 6LoWPAN-NHC-HDR header.

- ♯ *CS:* if we assume in our constrained scenario that there is only one CS in each CSB, there is no need therefore for keeping 8 bits to indicate the number of crypto sessions. We are then able to encode the ♯ CS with 1 bit. If this bit is set to 0, only one CS is considered. In addition, to make our compression flexible, if the bit is set to 1, the number of CS is carried inline.

- *CS ID map type(MT):* if 0, the default GENERIC-ID map type defined in [84] is used. If set to 1, the CS ID map type is carried inline.

Table 4.5: Mikey-Ticket Common Header compression

Field (sizes in bits)	**Mikey Common Header**	**Our 6LoWPAN-NHC-HDR**
Version (V)	8	1
Data type (DT)	8	2
Next Payload	8	8
Verification V (VF)	1	1
PRF func (PRF)	7	1
CSB ID (CSB)	32	1
♯ CS	8	1
CS ID map type (MT)	8	1
CS ID map info (MI)	Variable length	1

- *CS ID map info (MI):* the CS ID map info size is kept variable in [84]. If we assume that there is only one CS in each CSB, we could use 1 bit for the encoding. If 0, the unique CS is identified with its corresponding mapping to the security protocol for which security associations are created. If set to 1, the map info field is carried inline.

The next payload field is always carried inline as it is impossible to predict or deduce the next payload content. In addition, the three last bits are used as padding bits to remain standard compliant with RFC6282 [57] (NHC size is defined as 2 bytes long).

4.2.3.2 New Mikey-Ticket exchange mode

Our new communication exchange mode for Mikey-Ticket is designed to minimize the involvement of constrained nodes. We consider the constrained node as the Initiator of the protocol and the remote entity as the Responder. The constrained node is in charge of requesting the establishment of a session key with the remote entity and periodically sending updates. We assume that I and R are sharing security credentials

with the *KMS* that is in charge of generating, deriving and delivering the required keying materials. Besides, AES-CTR (AES in Counter Mode) algorithm, which is specified as mandatory-to-implement in RFC 3830 [18] is used for encryption. Also, AES-CBC (AES in Cipher Block Chaining mode) is used for MAC computation. Our communication exchange mode is depicted in Figure 4-15 and Table 4.4 summarizes the different notations used. It is worth mentioning here that although mode 2, mode 3 and mode 4 introduced in RFC 6043 [84] reduce the number of exchanged messages compared to the full three round-trip mode, they introduce strong assumptions on the ability of both I and R to either handle the generation and distribution of security credentials or to share credentials prior to the start of the session. For these reasons, our proposed exchange mode can be considered as an extension of the proposed exchange modes defined in RFC 6043 [84]. In fact, our new exchange mode does not assume any capabilities regarding neither I nor R as it is intended to be adaptable to constrained e-health scenarios.

REQUEST_INIT: the Initiator starts the exchange process by sending a *REQUEST_INIT* message to *KMS*. This message contains the identities of I (I_{ID}), *KMS* (KMS_{ID}), and R (R_{ID}). In addition, it contains a nonce N_I generated by I, which will be used as a session identifier. Furthermore, node I computes a MAC using $aK_{I,KMS}$ to ensure message authenticity. The message is then sent to *KMS*. *REQUEST_INIT* has the following structure: $\{[I_{ID}, R_{ID}, KMS_{ID}, N_I]_{eK_{I,KMS}}, MAC\}$.

REQUEST_RESPONSE: when *KMS* receives the *REQUEST_INIT* message, it validates the MAC using $aK_{I,KMS}$. Upon successful verifications, *KMS* decrypts the message using $eK_{I,KMS}$ and retrieves the different identities and the nonce N_I. If the request is authorized, *KMS* generates the requested key $K_{I,R}$ and uses the key derivation function defined in RFC3830 [18] to derive both $aK_{I,R}$ and $eK_{I,R}$. Then, *KMS* constructs two versions of *REQUEST_RESPONSE* message. The first message is intended to I. It is encrypted with $eK_{I,KMS}$ and contains a MAC computed using $aK_{I,KMS}$. In addition, the message contains the nonce N_I. The sec-

ond message is intended to R. It contains a MAC computed using $aK_{KMS,R}$ and is encrypted using $eK_{KMS,R}$. In addition, KMS generates a nonce N_{KMS} and includes it in the message along with N_I. The $REQUEST\_RESPONSE$ is intended to node I, and has the following structure: $\{[I_{ID}, R_{ID}, KMS_{ID}, aK_{I,R}, eK_{I,R}, N_I]_{eK_{KMS,I}}, MAC\}$. The $REQUEST\_RESPONSE$ intended to R has the following structure: $\{[I_{ID}, R_{ID}, KMS_{ID}, aK_{I,R}, eK_{I,R}, N_I, N_{KMS}]_{eK_{KMS,R}}, MAC\}$. The two versions are then sent to I and R.

TRANSFER_END: upon receiving a $REQUEST\_RESPONSE$ message, R checks the freshness of N_{KMS} and validates the MAC using $aK_{KMS,R}$. Upon successful verification, R decrypts the message and retrieves both $aK_{I,R}$ and $eK_{I,R}$. Node I proceeds similarly and retrieves $aK_{I,R}$ and $eK_{I,R}$ upon receiving $REQUEST\_RESPONSE$ message. R constructs $TRANSFER\_END$ as a verification message. It includes the nonce N_I and computes a MAC using $aK_{I,R}$. The message is then sent to I. This message has the following structure: $\{[I_{ID}, R_{ID}, N_I]_{eK_{I,R}}, MAC\}$. Upon receiving $TRANSFER\_END$ message, node I checks the freshness of N_I to avoid any replay attack and validates the MAC. A successful verification is considered as a proof of R's knowledge of both $aK_{I,R}$ and $eK_{I,R}$.

Our new communication exchange mode reduces therefore the number of exchanged messages from six to four messages compared to the basic Mikey-Ticket defined in RFC 6043 [84] regardless of the ability of I and R to generate, derive or distribute security credentials. The constrained node processes and exchanges few messages. Moreover, we offload the derivation of the authentication and encryption keys to the *KMS*, which further reduces the overhead in the context of e-health applications.

4.2.4 Analysis

In this section, we provide a detailed analysis of our proposed tailoring for Mikey-Ticket both in terms of security analysis and energy consumption. Firstly, we conduct

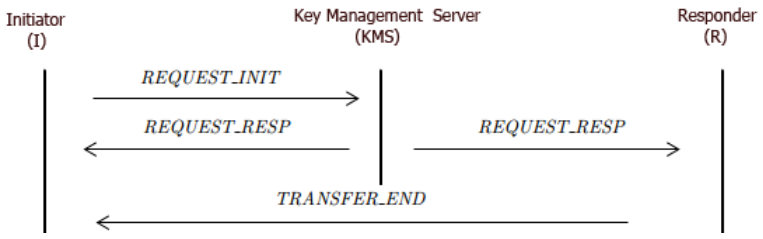

Figure 4-15: New Mikey-Ticket exchange mode

a theoretical security analysis of our new exchange mode. In addition, we analyze our protocol's behaviour against the well-known attacks that could hinder the establishment of a secure channel in an e-health environment. Our analysis is then validated using Avispa [1] which is based on formal models. After validating the security properties, we focus on the energy gain of our approach. Different energy models are used to estimate the total energy cost composed of both computational and communication costs. The results are compared with the basic version of Mikey-Ticket.

4.2.4.1 Security analysis

4.2.4.1.1 Key exchange properties The security features of our new Mikey-Ticket exchange mode have been assessed based on the properties presented in [112]. We have added extra analysis concerning integrity and confidentiality as we consider them critical for e-health applications. Hereafter, our communication channel is split into two parts or segments: Seg1) from the Initiator to the *KMS* and Seg2) from the *KMS* to the Responder.

- *Confidentiality :* The exchanged data between the different entities involved in our protocol are kept confidential. According to [18], AES-CTR is the default and mandatory-to-implement encryption algorithm. Nowadays, more and more tiny sensors include AES hardware coprocessors which help to decrease the overhead. For *Seg1*, encryption is based on the encryption key $eK_{I,KMS}$ shared between the Initiator (constrained node) and the *KMS*, whereas in *Seg2*, encryption is ensured by the use of the encryption key $eK_{KMS,R}$ shared between

the *KMS* and the Responder (remote server). In addition, periodical updates of the established keys are required in order to strengthen the confidentiality and prevent long term attacks.

- *Authentication and integrity:* By using *MAC* messages either in *Seg1* or in *Seg2* communication parts, our new exchange mode ensures that the exchanged data is genuine. In particular, it ensures that data has not been altered and has been sent from legitimate nodes. *MAC* messages are computed and appended to the exchanged messages based on AES-CBC mode using $aK_{I,KMS}$ in *Seg1* and $aK_{R,KMS}$ in *Seg2*. Furthermore, nonces (e.g. time-stamps or random values) are included in the exchanged messages to avoid replay attacks.

- *Distribution:* The distribution of security credentials in both communication segments is performed by an offline dealer during the initialization phase. This constitutes one of the major drawbacks of key distribution schemes based on a pre-shared context. In return, these schemes simplify the cryptographic operations (i.e. Symmetric) at the nodes side which is highly desirable in constrained environments. Besides, upon the establishment of a shared context, our new exchange mode can be run in an online manner, which allows autonomous update procesing.

- *Overhead:* The computation overhead is particularly low. Our compression scheme allows a considerable improvement in energy consumption as the size of the exchanged messages is reduced. Moreover, the constrained nodes are involved in fewer messages compared to the full three round-trip Mikey-Ticket mode (see Figure 4-10 and Figure 4-15). Constrained nodes are thus less solicited as they take advantage of the shared pre-established context with the *KMS*. A more detailed analysis regarding energy consumption is provided in section 4.2.4.2

- *Resilience:* The resilience of our scheme is high. In fact, the loss of a node and thus its key affects only the corresponding sensor as each sensor only stores its shared key with the *KMS* (i.e. $K_{R,KMS}$) and eventually an established key with I (i.e. $K_{I,R}$). The *KMS* maintains a different key with each constrained node either for the pre-shared context or for the generated shared key.

- *Extensibility and scalability:* Our network model allows new sensors as well as new remote entities to be added (e.g. we can imagine a physician prescribing the implantation of a new sensor for medical reasons). An offline dealer will have to establish a shared context between the new entities and the *KMS*. No extra operation is required from existing constrained nodes or remote entities when new nodes join them. As a result, high scalability is ensured which is particularly required for constrained environments.

- *Storage:* Smart objects now provide considerable amounts of storage space due to recent advances in flash memory technology [131]. Moreover, our new exchange mode does not add further credentials to be stored in the constrained nodes. The amount of data to be stored is limited, as only two keys (i.e. $K_{R,KMS}$ and $K_{I,R}$) have to be stored. Storage space will therefore not limit the deployment of our scheme.

4.2.4.1.2 Protocol behaviour against e-health well-known attacks E-health applications are subject to several attacks that threaten the establishment of secure channels [79] [64] [80]. In this section, we analyze the behaviour of our protocol against these attacks. We focus on the attacks that occur in the network and transport layers of the OSI (Open System Interconnection) model.

Ensuring key freshness is an important concern with regards to our new Mikey-Ticket exchange mode. Indeed, to provide the perfect forward secrecy property, the involved entities have to be able to detect replayed messages. In particular, e-health applications might be more vulnerable compared to other types of applications as an outdated information could lead to inadequate and serious medical consequences. To overcome this issue, we have introduced the use of nonces in the different exchanged messages. In fact, these nonces are implemented using one of the following strategies according to the network segment, and to the constrained node capabilities:

- Random numbers

- Sequence numbers

- Timestamps

Random numbers might constitute a solution in our e-health scenario. The constrained node (i.e. the Initiator) maintains a list of the previous received random values in its internal memory. Upon receiving a new message, the initiator checks if the nonce has already been received. As a result, replayed messages are detected. This solution brings a drawback ; the constrained node has to maintain a list of the received nonces in its internal memory. This issue can be attenuated by the storage capacity of new developed nodes [131]. The second solution is based on sequence numbers, which does not require any data storage. Sequence numbers provide a sequential counter in the exchanged messages. In case where a message is replayed, its counter will be smaller or equal to the current one. Thus, the message will be dropped. However, if the KMS goes down (e.g. reboot, hardware failure, etc.), this protection is no longer effective. In fact, the KMS will lose track of the current counter value. Besides, to ensure message freshness, timestamps could also be used. This solution is not suitable for constrained devices as it consumes a lot of energy. In fact, synchronized clocks have to be maintained between the KMS, the remote server, and the constrained nodes.

Taking into account our network specifications, we discuss the feasibility of the precedent solutions. It is obvious that maintaining clock synchronization between KMS and the constrained nodes is not feasible. However, this solution is adopted to protect the unconstrained part of the network model, namely the channel linking the KMS with the remote server (Seg2). In fact, the Responder and the KMS are not able to challenge each other and they are considered as non-constrained entities that are able to maintain clock synchronization between them. Hence, the nonces are implemented as timestamps. By doing so, the KMS and the remote server will easily prevent replay attacks.

Regarding Seg1 communication part, our proposed exchange mode allows the Initiator to challenge the KMS about the nonce. In addition, the constrained node is not able to maintain clock synchronization with the KMS. Consequently, the solution based on random numbers (or sequence numbers) is adopted. If the storage capacity of smart objects is very limited, the solution based on sequence numbers is prefered at the expense of ensuring a highly reliable entities with small probabilities of failure. If storage capacity is not a concern, the solution based on random numbers can be adopted. In brief, protecting our new exchange mode against replayed messages is achieved through the combination of the above discussed strategies according to the network model specificities.

Denial of Service (DoS) attacks could seriously threaten the availability of our e-health application. In fact, the gathered health related data should always be available even if the system is under a DoS attack. Like the basic version of Mikey-Ticket, our new exchange mode is protected against DoS attacks by using the same techniques. In particular, the KMS does not establish any internal state before authenticating both the remote server and the constrained nodes. The different parties share a long-term key with the KMS. Each exchanged message is authenticated before being processed. Besides, classical countermeasures such as rate-limiting and ACL (Access Control List) could also be implemented . Any malicious message would lead

to an abortion of the protocol execution. Node redundancy could be another option. Whenever an entity is made unavailable due to a DoS attack, the protocol execution carries on with the redundant backup node. We refer to [18] and [84] for a more detailed analysis of Mikey-Ticket behaviour regarding DoS attacks.

Sybil attacks [40] [64] where a node claims multiple fake identities could lead to harmful consequences in the context of e-health applications. Using these attacks, an intruder could use feigned identities to send false information. As a result, either genuine emergency situations are skipped, or ceaseless false emergency situations are thrown. Our protocol is protected against Sybil attacks. There is no way for a malicious node to perform a Sybil attack, unless the KMS (assumed to be a trusted entity) is corrupted. In fact, long term keys are shared between the KMS, the Initiator (i.e. sensor), and the Responder (i.e. remote server). Any exchanged message with the KMS contains the identity of the sender, and is authenticated using the pre-shared long term keys. In addition, before any further processing, the KMS checks its access control policy regarding the sender.

Another point of interest regarding the threat model in e-health applications is the attacks that aim to drain the energy power of sensors, and therefore make them unavailable or force them to enter a sleep mode. For instance, the De-synchronization attack targets the sequence number of the exchanged messages. Actually, this will lead to infinite retransmissions which waste both energy and bandwidth resources. Providing message integrity is the main security concern that hinders this type of attacks. In fact, MAC messages are computed and checked for each exchanged message ensuring that the included data has not been altered.

E-health applications are subject to several routing attacks. Our key management protocol is not involved in securing the routing process, instead, it aims to establish a secure channel upon which the gathered data can be securely transmitted. In fact, we rely on other mechanisms regarding this aspect. Countermeasures usually involve

the introduction of Intrusion Detection Systems (IDS) [109] [77].

4.2.4.1.3 Formal validation The formal validation of our protocol was carried out using the same Avispa tool to prove that our new exchange mode does not violate the required security properties, in particular, confidentiality, authentication, delivery proof and replay protection. The actions of the different entities are specified in a module called *basic role*, while their interactions are defined by composing multiple *basic roles* together into a *composed role*. In addition, the security goals of the analyzed protocol are specified in the *goal section* before launching the analysis. Besides, Avispa uses four different automatic protocol analysis techniques to validate the analyzed protocol against the specified security goals: on-the-fly model-checker (OFMC), constraint-logic based attack searcher (CL-AtSe), SAT-based model checker (SATMC), and tree automata based on automatic approximations for the analysis of security protocols (TA4SP).

In our modeling, we have first specified a *basic role* to describe the actions of the different entities involved. Then, we have specified how the participants interact with each other in a *composed role*. For clarity reasons, we present our modeling using Alice-Bob ($A-B$) notation, where:

- *A: Constrained node*

- *B: Remote entity*

- *S: KMS*

The rest of the notations used are the same as those presented in Table 4.4.

- $A->S:\{I_{ID}, R_{ID}, KMS_{ID}, N_I\}_{eK_{I,KMS}}, MAC$

- $S->A:\{I_{ID}, R_{ID}, KMS_{ID}, aK_{I,R}, eK_{I,R}, N_I\}_{eK_{KMS,A}}, MAC$

- $S->B:\{I_{ID}, R_{ID}, KMS_{ID}, aK_{I,R}, eK_{I,R}, N_I, N_{KMS}\}_{eK_{KMS,B}}, MAC$

```
user@instant-contiki:~/NewMikeyMode$ avispa  Mikey.hlpsl --ofmc
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
```

Figure 4-16: Avispa output (OFMC)

```
user@instant-contiki:~/NewMikeyMode$ avispa  Mikey.hlpsl --cl-atse

SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
```

Figure 4-17: Avispa output ($CL - AtSe$)

- $B- > A : \{I_{ID}, R_{ID}, N_I\}_{eK_{I,R}}, MAC$

Upon completing modeling our exchange mode, we have checked its correctness using a protocol animation tool called SPAN [47] that has been introduced to help protocol developers in writing AVISPA specifications. The security goals were subsequently evaluated by executing the four Avispa's backends (i.e. OFMC, $CL - AtSe$, SATMC and TA4SP). Besides, we have used the default Dolev-Yao intruder model which allows to simulate an intruder that has full control over the network. All messages sent and received by the different entities might be intercepted, analyzed, modified (as far as the keys are known), or sent to other entities.

The results of the analysis were indicated in reports for each backend model produced by Avispa tool. Our new exchange mode is "SAFE" against OFMC (Figure 4-16), $CL-AtSe$ (Figure 4-17) and SATMC (Figure 4-18). However, against TA4SP database, the result was "INCONCLUSIVE". According to Avispa user manual [1], an inconclusive result does not imply that an attack has been detected (Figure 4-19). Consequently, based on the obtained results, we can affirm that our protocol is safe regarding the specified security goals. It is impossible for an attacker to violate any

```
user@instant-contiki:~/NewMikeyMode$ avispa  Mikey.hlpsl  --satmc
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
```

Figure 4-18: Avispa output (SATMC)

```
user@instant-contiki:~/NewMikeyMode$ avispa  Mikey.hlpsl  --ta4sp
SUMMARY
   INCONCLUSIVE
```

Figure 4-19: Avispa output (TA4SP)

of the specified security properties and disrupt the functioning of the protocol.

Following our formal validation, we focus, in the next section, on the energy cost savings achieved through our new exchange mode and our header compression scheme. The results are compared with the performances of the basic version of Mikey-Ticket.

4.2.4.2 Performance analysis

As explained above, our contribution focuses on tailoring Mikey-Ticket to the constrained environment of e-health applications. To this end, we propose a new header compression scheme along with a new exchange mode to reduce both the size of the exchanged messages and their number. In this subsection, we provide a performance analysis of our enhancements and compare energy consumption with the basic Mikey-Ticket. First, we describe the energy model upon which our estimations are based. Then, we evaluate the communication and computational costs regarding both versions of Mikey-Ticket (i.e. basic version and tailored version). The analysis is concluded with a discussion of the total energy cost highlighting the obtained energy savings.

4.2.4.2.1 Energy model and assumptions Authors, in [89], have presented an energy evaluation of Wireless Sensor Nodes (WSN) regarding the communication cost. This latter is composed of the costs of transmission, reception and listening. Besides, the energy consumption of AES encryption algorithm and SHA-1 hash algorithm on WSN nodes have been also assessed in [68]. Both implementations were processed on tiny nodes with few MHz of computational power, several kilobytes of RAM and several tens of kilobytes of ROM.

In our evaluation, we consider the total energy cost as the sum of the communication cost and the computational cost. This latter is composed of encryption primitives based on AES and authentication primitives based on SHA-1 as specified in RFC 3830 [18]. Based on the energy measurements presented in [89] and [68], we estimate the energy consumption of tiny nodes regarding both communication and computational aspects. The deduced values, summarized in Table 4.6, are used as an energy model of the different operations on constrained nodes. Transmission, reception, listening and cryptographic operations costs are considered for the evaluation of the total energy cost.

A set of assumptions is defined before diving into the details of our evaluation:

- Our evaluation only covers energy consumption of the constrained nodes as remote entities are not affected by resources scarcity. Hence, the efforts of reducing energy consumption are focused on the constrained part of the network model.

- In the estimation of message sizes, we only take into consideration the header part on which our compression scheme is applied. The other parts of the exchanged messages are constant regarding the two versions of Mikey-Ticket.

- Mikey specification has left the CS ID map info variable in length. In order to carry out our evaluation, we assume a 2 bytes long field.

Table 4.6: Estimated energy costs on constrained nodes (TelosB)

Operation	**Cost**
Transmit 1 bit	0.72 µJ
Receive 1 bit	0.81 µJ
Listen for 1 ms	0.29 µJ
AES-128 128-bits computation	28.11 µJ
SHA-1 128-bits MAC computation	23.9 µJ

Table 4.7: Different compression rates

Compression rate (%)	**Compressed fields**	**Gained space (bits)**
0	None of the fields are compressed	0
16.4	V, DT	13
32.9	V, DT, PRF, MT	26
51.9	V, DT, PRF,♯ CS ,MI	41
72.1	V, DT, PRF, MT, CSB	57
83.5	V, DT,♯ CS, MI, CSB	66
100	All the fields are compressed	72

- In order to evaluate the gains in energy savings of our compression scheme, we propose several levels of compression rates. These rates simulate different applications, each one defines a subset of fields to be compressed using our proposed 6LoWPAN-NHC-HDR. Table 4.7 presents the different compression rates along with the corresponding compressed fields.

4.2.4.2.2 Communication cost

- *Sending cost:* the sending cost is estimated by computing the overall size of the messages sent from the constrained node for both Mikey-Ticket's versions. The cost is then computed for different levels of compression rate using the proposed energy model. Table 4.8 summarizes the results.

- *Receiving cost:* the receiving cost is estimated by computing the overall size of the messages sent to the constrained node for both Mikey-Ticket's versions.

Table 4.8: Sending cost

	Compression (%)	**Size (Bits)**	**Messages**	**Cost (μJ)**
Basic Mikey-Ticket	0	96	02	138.24
Tailored Mikey-Ticket	16.4	83	01	59.76
Tailored Mikey-Ticket	32.9	70	01	50.4
Tailored Mikey-Ticket	51.9	55	01	39.6
Tailored Mikey-Ticket	72.1	39	01	28.08
Tailored Mikey-Ticket	83.5	30	01	21.6
Tailored Mikey-Ticket	100	24	01	17.28

The cost is then computed for different levels of compression rate using the proposed energy model. Table 4.9 summarizes the results.

- *Listening cost:* We consider the constrained node listening for a period of time equal to the sum of packets propagation delay (Δ), packets computation time (Comp), transmission latency (T) and reception latency (R). We assume the *KMS* being at one hop from the constrained node and 150 ms propagation delay needed for routing packets from the *KMS* to the remote entity. Moreover, we assume both *KMS* and *R* being 100 times more powerful than the tiny node *I* for the estimation of computational time. Furthermore, we consider, for the estimation of communication latency, an effective data rate of 75 *kbps* for a tiny node (e.g. TelosB) [89]. As an example, in the basic Mikey-Ticket exchange mode, between the sending of $REQUEST\_INIT$ message and the reception of $REQUEST\_RESP$ message, the constrained node (CN) remains in the listening mode during the following period of time:

$$T_{listening} = R(KMS) + Comp(KMS) + T(KMS) + \Delta(KMS -> CN).$$

Where:

Table 4.9: Receiving cost

	Compression (%)	**Size (Bits)**	**Messages**	**Cost (µJ)**
Basic Mikey-Ticket	0	96	02	155.52
Tailored Mikey-Ticket	16.4	83	02	134.46
Tailored Mikey-Ticket	32.9	70	02	113.4
Tailored Mikey-Ticket	51.9	55	02	89.1
Tailored Mikey-Ticket	72.1	39	02	63.18
Tailored Mikey-Ticket	83.5	30	02	48.6
Tailored Mikey-Ticket	100	24	02	38.88

Table 4.10: Listening cost

	Compression (%)	**Time (mS)**	**Cost (µJ)**
Basic Mikey-Ticket	0	155.23	45.01
Tailored Mikey-Ticket	16.4	153.32	44.5
Tailored Mikey-Ticket	32.9	152.72	44.3
Tailored Mikey-Ticket	51.9	152.1	44.1
Tailored Mikey-Ticket	72.1	151.5	43.9
Tailored Mikey-Ticket	83.5	151.2	43.8
Tailored Mikey-Ticket	100	150.9	43.7

- *R(KMS): Reception latency of KMS*
- *Comp(KMS): Computational time of KMS*
- *T(KMS): Transmission latency of KMS*
- $\Delta(KMS->CN)$*: Packets propagation delay from KMS to CN*

The cost is computed for different levels of compression rate, Table 4.10 summarizes the results. We notice a slight difference between the energy consumption of the two versions of Mikey-Ticket. This is due to the fact that the listening time at the constrained node (i.e. I) is based on the time spent by the unconstrained nodes (i.e. KMS and R) to compute and communicate Mikey-Ticket messages. In fact, their unconstrained resources make our tailoring's impact less visible.

Table 4.11: Cryptography cost

	Compression(%)	Size (Bits)	Messages	Cost (µJ)
Basic Mikey-Ticket	0	96	04	84.33
Tailored Mikey-Ticket	16.4	83	03	54.68
Tailored Mikey-Ticket	32.9	70	03	46.11
Tailored Mikey-Ticket	51.9	55	03	36.23
Tailored Mikey-Ticket	72.1	39	03	25.69
Tailored Mikey-Ticket	83.5	30	03	19.76
Tailored Mikey-Ticket	100	24	03	15.81

4.2.4.2.3 Computational cost

- *Cryptography cost:* the cryptography cost is estimated by computing the overall size of the encrypted messages exchanged with the constrained node for both Mikey-Ticket's versions. The cost is then computed for different levels of compression rate using the proposed energy model. Table 4.11 summarizes the results.

4.2.4.2.4 Discussion Upon energy cost evaluation regarding both communication and computational aspects, we have estimated the overall energy cost considering both versions of Mikey-Ticket. The results are synthesized in Table 4.13. As shown in Figure 4-20, we have already noticed a marked decrease at the first compression rate (i.e. 16,4%) due to the introduction of both new exchange mode and compression scheme which reduces the size and the number of the exchanged messages. Energy consumption keeps decreasing with the augmentation of compression rate. In fact, nearly 74% less energy is required to perform a full key exchange in the best case of our compression scheme.

The obtained results were expected as the reduction of both size and number of messages leads to a decrease in the energy spent either in the processing or in the communication of data. Nevertheless, an additional processing overhead is

Table 4.12: Authentication cost

	Compression(%)	**Size (Bits)**	**Messages**	**Cost (μJ)**
Basic Mikey-Ticket	0	96	04	71.7
Tailored Mikey-Ticket	16.4	83	03	46.49
Tailored Mikey-Ticket	32.9	70	03	39.21
Tailored Mikey-Ticket	51.9	55	03	30.80
Tailored Mikey-Ticket	72.1	39	03	21.84
Tailored Mikey-Ticket	83.5	30	03	16.80
Tailored Mikey-Ticket	100	24	03	13.44

- *Authentication cost:* the authentication cost is estimated by computing the overall size of the messages exchanged with the constrained node on which a MAC is appended. The estimation is done regarding both Mikey-Ticket's versions. The cost is then computed for different levels of compression rate using the proposed energy model. Table 4.12 summarizes the results.

Table 4.13: Total energy cost

	Compression(%)	Comm	Comp	Total
Basic Mikey-Ticket	0	338.77	156.03	494.8
Tailored Mikey-Ticket	16.4	238.72	101.17	339.89
Tailored Mikey-Ticket	32.9	208.1	85.32	293.42
Tailored Mikey-Ticket	51.9	172.8	67.03	239.83
Tailored Mikey-Ticket	72.1	135.16	47.53	182.69
Tailored Mikey-Ticket	83.5	114	36.56	150.56
Tailored Mikey-Ticket	100	99.86	29.26	129.11

expected due to the compression/decompression operations of 6LoWPAN packets. As we consider the KMS being unconstrained, we can safely assume that the generated overhead will be supported by the KMS acting as a 6LoWPAN Border Router (6BR).

Additionally, we have compared the energy cost of several rekeying operations regarding different compression rates (Figure 4-20). Frequent updates are likely to be performed in order to avoid long term attacks. The results show a considerable gain in the energy consumption that increases with the increase of rekeying operations. It is worth noticing that the gain is more important with the increase of rekeying operations which is critical for tiny nodes with highly constrained resources (e.g. increasing battery lifetime).

The analysis study allowed us to validate our proposition from two perspectives. First of all, we have provided a theoretical analysis regarding the different security properties required in our network scenario. The properties analysis has been validated using Avispa tool. Furthermore, we have proceeded with a quantitative analysis to highlight energy savings resulting from our tailoring of Mikey-Ticket. Analysis study showed the viability of the proposed solutions on e-health environments that are based on highly constrained sensor nodes. In a nutshell, our proposed solutions make Mikey-Ticket more lightweight while its

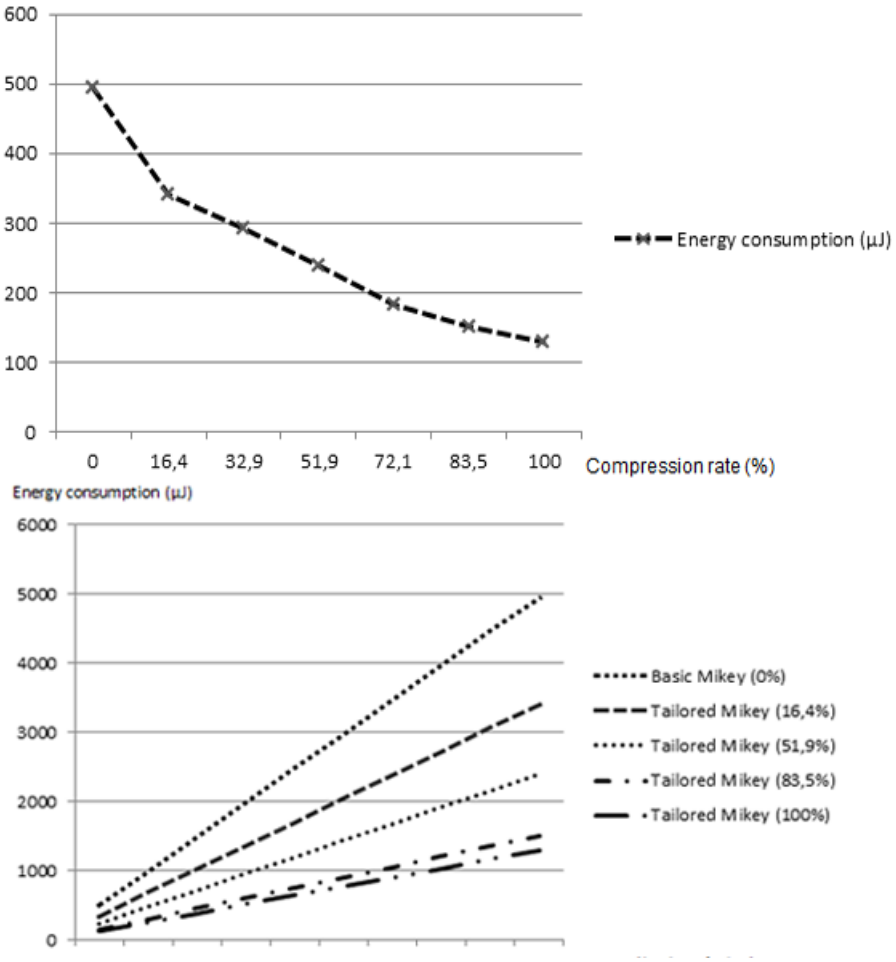

Figure 4-20: Total energy consumption on a constrained node (TelosB) for basic and tailored Mikey-Ticket regarding different compression rates and several rekeying operations

security properties are preserved.

4.2.5 Conclusion

We have introduced a tailoring mechanism for Mikey-Ticket to adapt it to low-power and constrained environment of e-health devices and applications. To this end, we have proposed a new header compression scheme to reduce the size of messages from 12 Bytes to 3 Bytes in the best compression case. In addition, we have introduced a new exchange communication mode to reduce the number of exchanged messages from six to four. We have evaluated our new solutions with respect to security and energy saving aspects. The results demonstrate that our approach keeps Mikey-Ticket safe while considerable amount of energy is saved at the constrained node side. Hence, we can claim that our adjustments of Mikey-Ticket protocol are well-adapted to IoT constrained environments such as e-health applications. However, Mikey-Ticket aims to establish a secret between entities with a pre-established shared context. Indeed, in IoT applications, nodes will likely need to establish a secure channel with any remote entity, eventhough no pre-shared context have been established beforehand. To address this issue, we introduce, in the next chapter our third protocol. In fact, in this protocol, nodes are able to establish a shared secret with any remote entity in an end to end way.

Chapter 5

Decentralized approaches

In this chapter, we present two decentralized protocols. The first protocol is entitled "An end-to-end secure key management protocol for e-health applications" [9] [12]. This protocol is designed to allow constrained entities establishing a secure communication channel with any other remote entity in the context of e-health applications. The second protocol entitled "A Decentralized Batch-based Group Key Management Protocol for Mobile Internet of Things (DBGK)" [13] aims to manage key establishment in multicast communication scenarios in the context of IoT. In the following, we present in detail these two protocols, along with their validation.

5.1 An end-to-end secure key management protocol for e-health applications

In this contribution, we propose a new distributed key management protocol. This protocol is based on collaboration to establish a secure end-to-end communication channel between a highly resource constrained node and a remote entity. The secure channel allows the constrained node to transmit captured data while ensuring confidentiality and authentication. To achieve this goal, we

propose offloading highly consuming cryptographic primitives to third parties. As a result, the constrained node obtains assistance from powerful entities. To assess our protocol, we conduct a formal validation regarding security properties. In addition, we evaluate both communication and computational costs to highlight energy savings. The results show that our protocol provides a considerable gain in energy while its security properties are ensured. To the best of our knowledge, no previous scheme has been introduced in the literature for e-health applications with the following properties:

- Keeping the constrained nodes only involved in simple operations (i.e. symmetric cryptography)

- Allowing the constrained nodes to dynamically establish a shared key with any remote entity with which no previous shared knowledge is established. To achieve this goal, third parties are dedicated to support the constrained nodes in this process.

- Guaranteeing the end-to-end principle as no entity has the knowledge of the exchanged secret apart from the constrained nodes and the remote entity.

- The third parties are not necessarily trusted entities. Indeed, each one of them is only in possession of a part of the secret. Thus, the only way to corrupt the exchanged secret is the colluding of all the third parties.

- Our protocol is not specific to e-health applications. In fact, it could be applied to a wide range of applications. In addition, it does not alter any cryptographic primitive, and thus allowing a high level of backward compatibility. As a result, any existing standard based cryptographic algorithm might easily be integrated.

5.1.1 The proposed protocol

In this section, we present our lightweight end-to-end key management protocol. Firstly, we present the network model and a set of assumptions. Afterwards, we provide a broad overview of our protocol along with a summary of the used notations. Finally, we describe in detail the different phases of our protocol.

5.1.1.1 Network Model

We consider in our network model four main components: mobile and contextual sensors (constrained nodes), third parties, a remote server, and a certification authority (See Figure 5-1).

- *Mobile and contextual sensors:* the sensors are planted in, on or around a human body to collect health-related data (e.g. blood pressure, blood glucose level, temperature level, etc.).

- *Third Parties:* the third parties represent a key component in our protocol. A third party could be any entity able to perform high consuming computations on behalf of the sensor nodes. In fact, the resource constrained sensors rely on them by offloading high consuming cryptographic primitives in a cooperative way. The set of third parties can be provided by an external service provider as part of a cloud infrastructure.

- *Remote server:* the remote server receives the gathered data for further processing. A remote server could be used by caregiver services in order to take appropriate decisions according to patient's data.

- *Certification authority:* the certification authority is required to guarantee authentication between the third parties and the remote server by

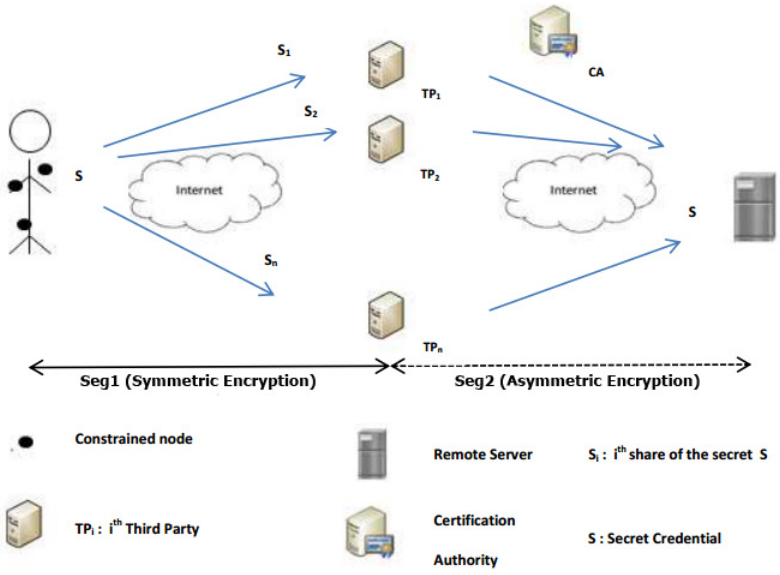

Figure 5-1: Network Model

delivering authenticated certificates.

The network is thus heterogeneous. It combines nodes with various capabilities both in terms of computing power and energy resources. In this network, we distinguish two categories of entities:

- Highly resource constrained nodes (mobile and contextual sensors), which are unable to perform public key cryptographic operations.

- Nodes with high energy, computing power, and storage capabilities (the third parties and the remote server).

Due to the high sensitivity of gathered data, we assume end-to-end secured communications between sensor nodes and the remote server. Hence, a key exchange protocol is required between the two entities to secure these communications. The protocol has to deal with resources capabilities of the involved entities, along with the fact that the remote server (i.e. remote entity) has not established any prior knowledge neither with the third parties nor with the constrained nodes (i.e. sensors).

5.1.1.2 Assumptions

For the implementation of our protocol, we assume that:

- Sensor nodes are able to perform symmetric encryption.
- Third parties are able to perform asymmetric cryptographic operations (either public or private).
- The remote server is powerful enough to support asymmetric encryption.
- The third parties are not necessarily trusted. In fact, one or several third parties can be corrupted without compromising the exchanged secret.
- The certification authority is a trusted entity. It delivers authenticated cryptographic credentials to the third parties, and to the remote server.
- Each sensor node is able to keep a list of remote third parties, which is pre-established during the initialization phase.
- Each sensor node shares pairwise keys with each third party. These keys are generated during the initialization phase.
- Both third parties and the remote server own a pair of public/private keys.

5.1.1.3 Overview of the proposed protocol

We provide a broad overview of our protocol before considering a formal description in the next section. Once a resource constrained sensor is willing to establish a shared secret with a remote server, it initiates our protocol. This latter goes through successive phases. First, the constrained node generates a secret key, which is then randomly split to several secret parts. The number of parts corresponds to the number of third parties involved. In the next step, the constrained node encrypts and sends each secret part to the corresponding third party. The encryption of each part is based on symmetric algorithms (less

Notation	Description
CN	Constrained Node (the sensor)
UN	Unconstrained Node (the remote server)
TP_i	Third Party
CA	Certification Authority
N_x	Nonce generated by node X
$K_{x,y}$	Shared pairwise key between X and Y
K_x	Public key of node X
K_x^{-1}	Private key of node X
$[data]_K$	Data encrypted with the key K
$SIGN_X$	X's digital signature
S	Secret credential used to secure communications between CN and UN

Table 5.1: Terminology Table

resource consuming than asymmetric ones) using pre-shared keys. In addition, MAC (Message Authentication Code) messages are used to ensure authentication. Once the third party receives its corresponding secret part, it encrypts and delivers it to the remote server. Using encryption, each third party secures the delivery. This encryption is based on asymmetric algorithms, which use the remote server's public key. Once the remote server receives the secret part, it uses digital signatures to ensure authentication. Upon successful authentication and decryption of the different secret parts, the remote server reassembles the shared secret, which will be used to derive further keying materials. In our protocol, we also make sure that each third party proves to the remote server that it is a legitimate entity, authorized by the constrained node to act on its behalf. The following section describes in detail each phase of our protocol.

5.1.1.4 Formal description

After an initialization phase, where each constrained node is pre-loaded with a set of third parties IDs along with pre-shared keys, our protocol proceeds with successive phases. Table 5.1 summarizes the notations used to present the

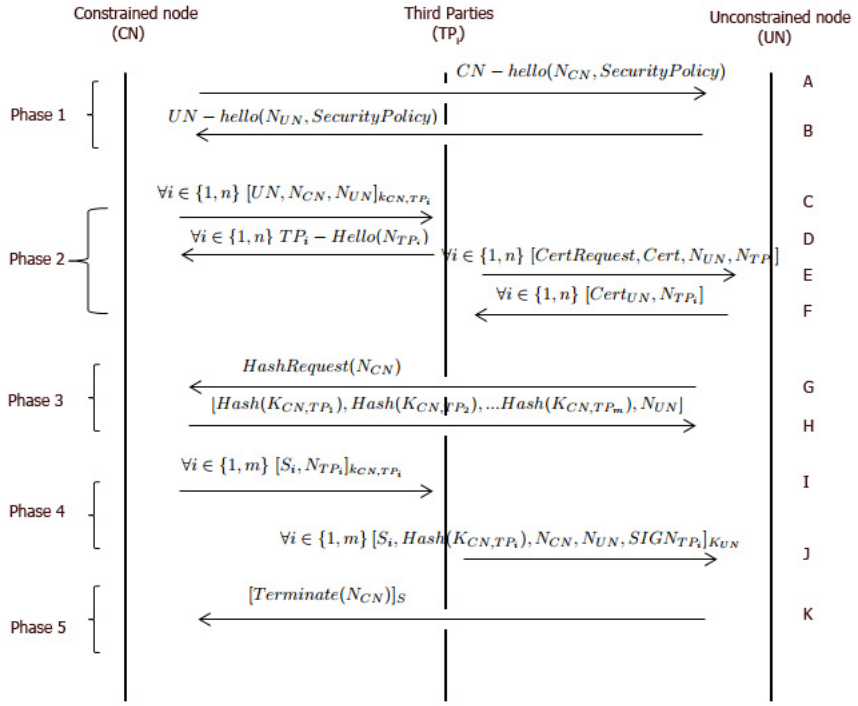

Figure 5-2: Illustration of the different phases and message exchanges of our protocol

exchanged messages, and Figure 5-2 illustrates the succeeding phases of our protocol.

- *Phase 1 (Initial exchange):* node CN initiates the exchange by sending a CN_HELLO message (A) to UN. This message informs UN about the security policies (e.g. encryption algorithm, HMAC algorithm, key life time, etc.), and the cooperative key establishment process it supports. If UN agrees, it selects one of the proposed security policies, and responds with a UN_HELLO message. Nonces are also included in the exchanged messages to prevent replay attacks.

- *Phase 2 (Securing connection between parties):* this phase follows the successful connection between CN and UN. It aims to establish a secure channel either between CN and TP_i or between TP_i and UN.

During this phase, CN sends message C to the third parties to inform them about UN identity. The message includes a Message Authentication Code (MAC), and is encrypted using K_{CN,TP_i}. The third parties express their willingness to be part of the key exchange protocol through message D. It is worth noting that not all the asked third parties respond with message D. For instance, this might be due to possible resource exhaustion. Hence, we consider that only m TP_i (m <= n) have responded with message D expressing their willingness to take part in the key exchange process. In message E, each TP_i provides UN with its own certificate containing its public key (delivered by CA). In addition, it requests the certificate of UN. This latter verifies that the third party has supplied a valid public key. It, then, responds with message F that contains the requested certificate. We highlight that all messages contain nonces against replay attacks.

- *Phase 3 (Proving third parties's representativeness of CN to UN):* this phase aims to prove to UN the representativeness of CN by the third parties. To this end, in message G, UN requests the pairwise keys shared with TP_i. In response, CN applies a hash function on each key to keep it confidential, and sends it to UN through message H. The authentication will occur later after receiving message J from the third parties, which should contain the same keys's hashes.

- *Phase 4 (Secret generation and delivery):* upon successful preparation of the involved entities, CN generates a premaster secret S, which will be used later to generate further keying materials at both CN and UN sides. The secret is split into m parts S_1, S_2, ..., S_m. Each part S_i is securely sent to the appropriate TP_i in message I. The communication is secured using the symmetric key K_{CN,TP_i}. Upon receiving message I, each TP_i

uses UN's public key to encrypt message J. This message contains the secret part S_i, K_{CN,TP_i}'s hash, and TP_i's signature that covers all the fields of the message. After the decryption of message J, UN verifies the authenticity of each message using TP_i's public key. If all the messages are authenticated, UN verifies TP_i's representativeness of CN. The verification is done by comparing the hashes received in message H and those received in message J. If the hashes match, the TP_i act on behalf of CN as they pretend. After having received all the packets, UN then reconstructs the secret S. This secret along with the previous exhanged nonces are used to derive further key credentials. Both messages I and J contain nonces to avoid replay attacks.

- *Phase 5 (Termination phase):* this phase concludes the exchanges through message K by proving to CN the knowledge of the secret S.

The derivation process is ensured by a hash function agreed upon during the first phase. Both parties are then able to derive state connection keys for encryption and authentication of the exchanged data. A secure end-to-end channel is hence created between highly constrained nodes and remote unconstrained servers.

5.1.2 Analysis

In this section, we provide a detailed analysis of our proposed key management protocol both in terms of security properties and energy consumption. Firstly, we propose a theoretical analysis of our protocol regarding its security properties. In addition, we highlight the resistivity of our protocol against the well-known attacks that could hinder the establishment of a secure channel in an e-health environment. Our analysis is then validated using Avispa [1]. Secondly, we focus on energy cost savings obtained through our protocol. To achieve this goal, we use energy models to estimate the total energy cost composed of both computational and communication costs.

5.1.2.1 Security analysis

5.1.2.1.1 Key exchange properties Security features of our protocol have been assessed based on the properties presented in [112]. We have added an analysis concerning integrity and confidentiality as we consider them being critical in an e-health application. For the following discussion, we consider our communication channel split into two segments: Seg1) from CN to TP_i and Seg2) from TP_i to UN (See Figure 5-1)

- *Confidentiality:* the exchanged data between the different entities involved in our protocol are kept confidential. For Seg1, symmetric encryption is used based on the pre-shared keys set during the initialization phase. We recommend the use of the AES-CCM mode that defines AES-CBC for MAC generation with AES-CTR for encryption [42]. Nowadays, more and more tiny sensors include AES hardware coprocessor, which would help to decrease the computational overhead. Regarding Seg2, communications are secured using Public Key Encryption (e.g. RSA algorithm). The CA delivers the required certificates to the involved entities. Our protocol can be run periodically to update the established keys in order to strengthen confidentiality, and prevent long term attacks.

- *Authentication and integrity:* through the use of MACs in Seg1 and digital signatures in Seg2, our protocol makes sure that the exchanged data are genuine. The aim is to ensure that data have not been altered, and have been sent from legitimate nodes. Our protocol also ensures that the involved TP_i prove their authenticity to UN. This is done through the comparison of the secret shared between CN and TP_i. (we refer to section 5.1.1.4 for more details). In addition, to avoid any replay attacks, we include nonces (e.g. time-stamps, random values, etc.) in the different exchanged messages.

- *Distribution:* the distribution of security credentials in Seg1 is performed by an off-line dealer during the initialization phase. However, in Seg2, through the use of Public Key Encryption, the involved entities establish a secure channel in an online mode. Thus, upon key's distribution in Seg1, our protocol can be run without any external intervention, allowing updates to be processed in an automatic way.

- *Overhead:* the computation overhead is relatively low. In fact, through the different handshakes of our protocol, constrained nodes are only involved in symmetric encryption primitives, which are much less resource consuming than asymmetric ones. All asymmetric operations are offloaded to third parties that are much more powerful. Doing so, our protocol limits the computational requests for the constrained nodes. As a result, it decreases their power consumption, and thus increases their battery lifetime.

- *Resilience:* the resilience of our protocol is high. Indeed, as S is split into several parts, an attacker has to take control of all third parties to compromise the exchanged secret S. Unless all TP_i are compromised, it is nearly impossible to recover the secret.

- *Extensibility and scalability:* our network model allows new sensors to be integrated (e.g. we can imagine a physician prescribing the implantation of a new sensor for medical reasons). The new sensor has to pass through an initialization phase. Then, the sensor will receive a set of TP_i 'IDs to rely on along with the pairwise keys shared respectively with each of them. This phase is performed by the network administrator. No operation is required concerning any TP_i or the remote servers that will be involved later in the protocol. Upon successful initialization phase, the new sensor

can establish an end-to-end secure channel with any remote entity.

- *Storage:* due to the recent advances in flash memory technology [131], smart objects now provide vast amounts of storage space. We rely on this space in our protocol to make the constrained nodes able to store the TP_i's ID list along with the corresponding shared keys. We also consider that the number of TP_i will not exceed a certain threshold defined by the network administrator. Therefore, storage space will not hinder our protocol's deployment.

5.1.2.1.2 Resistivity against e-health well-known attacks E-health applications are subject to several attacks that threaten the establishment of secure channels [79] [80]. In this section, we highlight the resistivity of our protocol against these attacks. We focus on the attacks that are positioned in the network and transport layer of the OSI (Open System Interconnection) model where our protocol is performed.

Ensuring key freshness is an important concern with regards to our protocol. Indeed, to provide the perfect forward secrecy property, the involved entities have to be able to detect replayed messages. In particular, e-health applications might be more vulnerable to this kind of attacks compared to other application scenarios; an outdated information could lead to inadequate medical interventions. To overcome this issue, we have introduced the use of nonces in the different exchanged messages. In fact, these nonces could be implemented using one of the following strategies against replayed messages:

- Random numbers
- Sequence numbers

- Timestamps

Random numbers might constitute a solution in our e-health scenario. The constrained node maintains a list of the previous received random values in its internal memory. Upon receiving a new message, the node checks if the nonce has already been received. As a result, replayed messages are detected. This solution brings a drawback; the constrained node has to maintain a list of the received nonces in its internal memory. Nevertheless, due to recent advances in flash memory technology [131], smart objects now provide vast amounts of storage space, which attenuates the storage issue. The second solution is based on sequence numbers, which do not require any data storage. Indeed, sequence numbers provide a sequential counter in the exchanged messages. In case where a message is replayed, its counter will be smaller or equal to the current one. Thus, the message will be dropped. However, if one of the involved third parties goes down (e.g. reboot, hardware failure, etc.), this protection is no longer effective. In fact, the third party will lose track of the current counter value. Besides, to ensure message freshness, timestamps could also be used. This solution is highly energy consuming to be implemented for a constrained node. In fact, synchronized clocks have to be maintained between the third parties, the remote server, and the constrained nodes.

Taking into account our constrained networks scenario, we discuss the feasibility of the precedent solutions. It is obvious that maintaining clock synchronization between the third parties and the constrained nodes is not feasible. However, this solution might be considered to protect the unconstrained part of the network model, namely the channel linking the third parties with the remote server (Seg2). Doing so, the third parties and the remote server will have no difficulty to avoid replay attacks using timestamps. Besides, according to the storage capabilities of the constrained nodes, the solution based on random numbers might be adopted for the constrained part of the network (Seg1). Nevertheless,

in case where the constrained nodes are highly limited regarding their storage capacity, the solution based on sequence numbers would be preferred at the expense of ensuring highly reliable entities with small probabilities of failure. In a nutshell, protecting our protocol against replayed messages could be achieved through the combination of the above discussed strategies according to the network model specificities.

Denial of Service (DoS) attacks could seriously threaten the availability of our e-health application. In fact, the gathered health related data should always be available even if the system is under a DoS attack. Indeed, if any of the involved nodes is made unavailable, in the sense that it is no longer able to gather or process data, this situation would engender disastrous consequences. For instance, we assume that a sensor is planted in the body of a patient suffering from a heart condition. In case where a heart related value that indicates an impending heart attack is registered, it should immediately be transmitted to health care services. Any delay due to a DoS attack could be fatal.

Several mechanisms are implemented in our protocol to mitigate DoS attacks regarding the involved entities. In fact, the constrained nodes share a long-term key with the third parties. Each exchanged message is authenticated upstream of any processing effort. In the same way, the third parties do not establish any internal state before authenticating both the remote server and the constrained nodes. Authenticating the constrained nodes is achieved through the long-term shared keys using MAC messages. Furthermore, authenticating the remote server is achieved through the exchange of certificates during phase 2 of our protocol. Besides, classical countermeasures could also be implemented such as rate-limiting and ACL (Access Control List). Any abnormal message would lead to an abortion of the protocol execution. In addition, based on the sensitivity of e-health applications, we advocate the use of redundancy. When-

ever a node is made unavailable by a DoS attack, the protocol execution carries on with the redundant node.

Sybil attacks, where a node claims multiple fake identities could be highly harmful in the context of e-health applications. Through these attacks, an intruder could use feigned identities to send false information. As a result, either an actual emergency situation is skipped or ceaseless false emergency situations are thrown. In addition, a third party could feign as being several third parties. Doing so, it increases its chances of retrieving more shares of the secret sent by the constrained nodes. In our protocol, sybil attacks are mitigated differently according to the segment of our network model. Regarding Seg1, the exchanged messages between the third parties and the constrained nodes contain the identity of the sender, and are authenticated using the pre-shared long-term keys. As a result, the third parties are not able to use multiple identities encrypted with the same shared key. Regarding Seg2, sybil attacks are mitigated through the use of trusted certification to make sure that each entity is assigned exactly one identity.

Another point of interest regarding the threat model of e-health applications is the attacks that aim to exhaust sensors energy making them unavailable. For instance, the De-synchronization attack targets the sequence number of the exchanged messages. This will lead to infinite retransmissions, which waste both energy and bandwidth. Providing message integrity is the main security property that hinders this type of attacks. In fact, MAC messages are computed and checked for each exchanged message ensuring that the included data have not being altered.

E-health applications are subject to several other attacks. In particular, routing attacks that can quickly hinder their functioning to the point of making them

unavailable [70]. Our key management protocol is not involved in securing the routing process. In fact, it aims to establish a secure channel upon which the gathered data can be securely transmitted. Indeed, we rely on other mechanisms to secure the routing process. Countermeasures usually involve the introduction of Intrusion Detection Systems (IDS)[93].

5.1.2.1.3 Formal validation In our modeling, we have first specified a *basic role* to describe the actions of the different entities involved. Then, we have specified how the participants interact with each other in a *composed role*. For clarity reasons, we present our modeling using a high level Alice-Bob ($A-B$) notation, where:

- *A: constrained node*
- TP_i*: third party*
- *B: remote entity*
- $A->B:\{N_A, SecurityPolicy\}$
- $B->A:\{N_B, SecurityPolicy\}$
- $A->TP_i:\{B, N_A, N_B\}_{K_{A,TP_i}}, MAC$
- $TP_i->A:\{N_{TP_i}\}_{K_{A,TP_i}}, MAC$
- $TP_i->B:\{CertRequest, Cert_{TP_i}, N_B, N_{TP_i}\}$
- $B->TP_i:\{Cert_B, N_{TP_i}\}$
- $B->A:\{HashRequest, N_A\}$
- $A->B:\{Hash(K_{A,TP_i}), N_B\}$
- $A->TP_i:\{S_i, N_{TP_i}\}_{K_{A,TP_i}}, MAC$
- $TP_i->B:\{S_i, Hash(K_{A,TP_i}), N_A, N_B, SIGN_{TP_i}\}_{K_B}$

- $B->A:\{N_A\}_S$

The rest of the notations used are the same as those presented in Table 5.1.

Upon complete modeling of our protocol, we have checked its correctness using a protocol animation tool called SPAN. This tool has been introduced to help protocol developers in writing AVISPA specifications. The security goals of our protocol were subsequently evaluated by executing the four Avispa's backends (i.e. OFMC, $CL-AtSe$, SATMC and TA4SP). Besides, we have used the default Dolev-Yao intruder model, which allows simulating an intruder that has full control over the network. In this model, all messages sent and received by the different entities might be intercepted, analyzed, modified (as far as the keys are known), or sent to other entities.

The results of the analysis were indicated in reports for each backend model produced by Avispa tool. These reports show that our protocol is "SAFE" against OFMC (See Figure 5-3), $CL-AtSe$ (See Figure 5-4) and SATMC (See Figure 5-5). However, against TA4SP database, the result was "INCONCLUSIVE" (See Figure 5-6). According to Avispa user manual [1], an inconclusive result does not imply that an attack has been detected. Consequently, based on the obtained results, we can safely affirm that our protocol is safe regarding the specified security goals. It is impossible for an attacker to violate any of the specified security properties, and disrupt the functioning of the protocol.

Following our formal validation, we focus in the next section on energy cost savings achieved through our protocol.

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl  --ofmc
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
```

Figure 5-3: Avispa output (OFMC)

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl --cl-atse

SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
```

Figure 5-4: Avispa output ($CL - AtSe$)

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl  --satmc
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
```

Figure 5-5: Avispa output (SATMC)

```
user@instant-contiki:~/OurProtocolModeling$ avispa  OurProtocol.hlpsl  --ta4sp
SUMMARY
   INCONCLUSIVE
```

Figure 5-6: Avispa output (TA4SP)

5.1.2.2 Performance analysis

As explained above, our contribution focuses on proposing a viable key management protocol to the constrained environment of e-health applications in the context of IoT. To this end, we base our solution on offloading heavy computational operations (i.e. asymmetric operations) to third parties. This makes the constrained nodes only solicited for symmetric operations, which are much less resource consuming than asymmetric ones. In addition, our protocol does not require the third parties to be trusted. Thus, the end-to-end principle is not broken. In this section, we provide a performance analysis of our protocol, and measure energy savings according to the number of third parties. The results allow us, in one hand, to emphasize the impact of the offloading on energy consumption, and in the other hand, to analyze the variation of energy consumption regarding the number of third parties involved in the protocol. We first describe the energy model on which our estimations are based. Then, we evaluate the communication and computational costs of our protocol according to the number of third parties. The analysis is concluded with a discussion of the total energy cost highlighting the obtained energy savings.

5.1.2.2.1 Energy model and assumptions Authors, in [89], have presented an energy evaluation of Wireless Sensor Nodes (WSN) regarding the communication cost. This latter is composed of the costs of transmission, reception, and listening. Besides, the energy consumption of AES encryption algorithm and SHA-1 hash algorithm on WSN nodes have been assessed in [68]. In addition, authors in [89], have also assessed the energy consumption of both ECC (Elliptic Curve Cryptography) encryption algorithm (160-bit keys) and ECDSA (Elliptic Curve Digital Signature Algorithm) signature algorithm. These implementations were processed on tiny nodes with few MHz of computational power, several kilobytes of RAM, and several tens of kilobytes of ROM

(i.e. TelosB nodes).

In our evaluation, we consider the total energy cost as the sum of the communication cost and the computational cost. This latter is composed of AES encryption primitives and SHA-1 authentication primitives. Based on the energy measurements presented in [89] and [68], we estimate the energy consumption of tiny nodes regarding both communication and computational aspects. The deduced values, summarized in Table 5.2, are used as an energy model for the different operations of our protocol. For the evaluation of the total energy cost, we consider the cost of transmission, reception, listening, and cryptographic operations.

Before presenting the details of our evaluation, we set the following assumptions:

- Our evaluation only covers energy consumption of the constrained nodes, as remote entities (i.e. third parties and the remote server) are not limited by resources scarcity. Hence, the efforts of reducing energy consumption are focused on the constrained part of the network model. In fact, we only consider the energy consumption generated by the exchanged messages with the constrained nodes.

- Based on our formal description depicted in section 5.1.1.4, we estimate the size in bytes of each exchanged message (See Table 5.3).

- In order to assess the impact of offloading resource consuming operations to third parties, we firstly consider a basic version of our protocol. In this version, tiny nodes do not rely on any third party to share secrets with remote entities. In fact, each tiny node generates the secret, encrypts it, and sends it to the remote entity. This basic version of our protocol uses classical PKI.

Operation	**Cost**
Transmit 1 bit	0.72 µJ
Receive 1 bit	0.81 µJ
Listen for 1 ms	0.29 µJ
AES-128 128-bits computation	28.11 µJ
SHA-1 128-bits MAC computation	23.9 µJ
ECC-160 point-mult	17 mJ
ECDSA-160 sign	15 mJ

Table 5.2: Estimated energy costs on a constrained node (TelosB)

- Symmetric algorithms are used to secure communications between the constrained nodes and the third parties. We assume the use of AES-CCM mode that defines AES-CBC for MAC generation and AES-CTR for encryption. Besides, we consider the use of asymmetric algorithms to secure communications between the third parties and any remote entity. We also consider the use of ECC for encryption and ECDSA for authentication, which are more energy saving compared to RSA based algorithms for the same level of security [129]. In addition, in case where no third entity is involved, asymmetric algorithms are used to secure communications between the constrained nodes and any remote entity.

- The constrained nodes are IP-enabled through a lightweight IP stack (i.e. 6LowPAN). Thus, using a gateway (i.e. 6LowPAN border router), 6lowPAN headers are compressed and decompressed. The gateway role might be played by a patient smartphone, or any other device placed near a patient. Hence, we suppose in our simulation that the messages sent from the constrained nodes are transmitted through short distances. We also assume 150 ms as a propagation delay needed for transmitting the messages with an effective data rate of 75 *kbps* for the constrained nodes (e.g. TelosB) [89].

- We consider that our network scenario is clear from interferences.

Exchanged Message	size (bytes)
A	104
B	104
C	88
D	68
G	14
H	64
I	100
K	4

Table 5.3: Size in bytes for each exchanged message with a constrained node

Number of third parties	**Size (Bytes)**	**Energy Consumption** (µJ)
0	318	1831.68
2	544	3133.44
4	920	5299.2
6	1296	7464.96
8	1672	9630.72
10	2048	11796.48

Table 5.4: Sending cost

5.1.2.2.2 Communication cost

- *Sending cost:* sending cost is estimated by computing the overall size of the messages sent by a constrained node. The cost is then computed according to the number of third parties involved using the proposed energy model. Table 5.4 summarizes the results.

- *Receiving cost:* similarly to sending cost, receiving cost is computed according to the overall size of the messages sent to a constrained node. The cost is then estimated based on the number of third parties involved using the proposed energy model. Table 5.5 summarizes the results.

- *Listening cost:* we consider the constrained node listening for a period of

Number of third parties	**Size (Bytes)**	**Energy Consumption** (μJ)
0	212	1373
2	258	1671.84
4	394	2553.12
6	530	3434.4
8	666	4315.68
10	802	5196.96

Table 5.5: Receiving cost

time equal to the sum of packets propagation delay (Δ), packets computation time (Comp), transmission latency (T), and reception latency (R). We suppose 150 ms propagation delay needed for routing packets between the different involved entities. Moreover, for the estimation of computational time, we suppose the third parties and the unconstrained remote entity (UN) being 100 times more powerful than the constrained nodes. Furthermore, for the estimation of communication latency, we consider an effective data rate of 75 *kbps* for a tiny node (e.g. TelosB) [89]. As an example, between the sending of message A and the receipt of message B, the constrained node (CN) remains in the listening mode during the following period of time:

$T_{listening} = \Delta(CN->UN)+R(UN)+Comp(UN)+T(UN)+\Delta(UN->CN)$.

Where:

- $\Delta(CN->UN)$*: Packets propagation delay from CN to UN*
- *R(UN): Reception latency of UN*
- *Comp(UN): Computational time of UN*
- *T(UN): Transmission latency of UN*

Number of third parties	**Listening Time** (mS)	**Energy Consumption** (µJ)
0	942	273.18
2	2409.16	698.6
4	4216.2	1222.6
6	6023.24	1746.7
8	7830.28	2270.78
10	9637.32	2794.82

Table 5.6: Listening cost

Number of third parties	**Size (Bytes)**	**Energy Consumption** (µJ)
0	36	34000
2	132	231.88
4	260	456.76
6	388	681.64
8	516	906.52
10	644	1131.4

Table 5.7: Cryptography cost

- $\Delta(KMS->CN)$*: Packets propagation delay from UN to CN*

Listening cost is computed according to the number of third parties involved. Table 5.6 summarizes the results.

5.1.2.2.3 Computational cost

- *Cryptography cost:* Cryptography cost is estimated by computing the overall size of the encrypted messages exchanged with a constrained node. The cost is then computed according to the number of third parties involved using the proposed energy model. Table 5.7 summarizes the results.

- *Authentication cost:* authentication cost is estimated by computing the overall size of the exchanged messages with a constrained node on which

Number of third parties	Size (Bytes)	Energy Consumption (μJ)
0	36	15000
2	132	197.2
4	260	388.4
6	388	579.6
8	516	770.8
10	644	962

Table 5.8: Authentication cost

Number of third parties	Communication cost	Computational cost	Total energy cost
0	3477.86	49000	52477
2	5503.88	429.08	5932.96
4	9074.92	845.16	9920.08
6	12646.1	1261.24	13907
8	16217.18	1677.32	17894.5
10	19788.26	2093.4	21881.66

Table 5.9: Total energy cost

a MAC or a digital signature has been calculated. The cost is then computed according to the number of third parties involved using the proposed energy model. Table 5.8 summarizes the results.

5.1.2.2.4 Discussion Upon energy cost evaluation regarding communication aspect, we have noticed an increase in energy consumption when the number of third parties augments. This is due to the increase of exchanged messages between the constrained node and the third parties. Besides, upon energy cost evaluation regarding the computational aspect, we have noticed high energy consumption in the case where no third party is used. This result is the consequence of using asymmetric cryptographic primitives by the constrained node. Indeed, these cryptographic primitives are much more resource consuming than symmetric primitives.

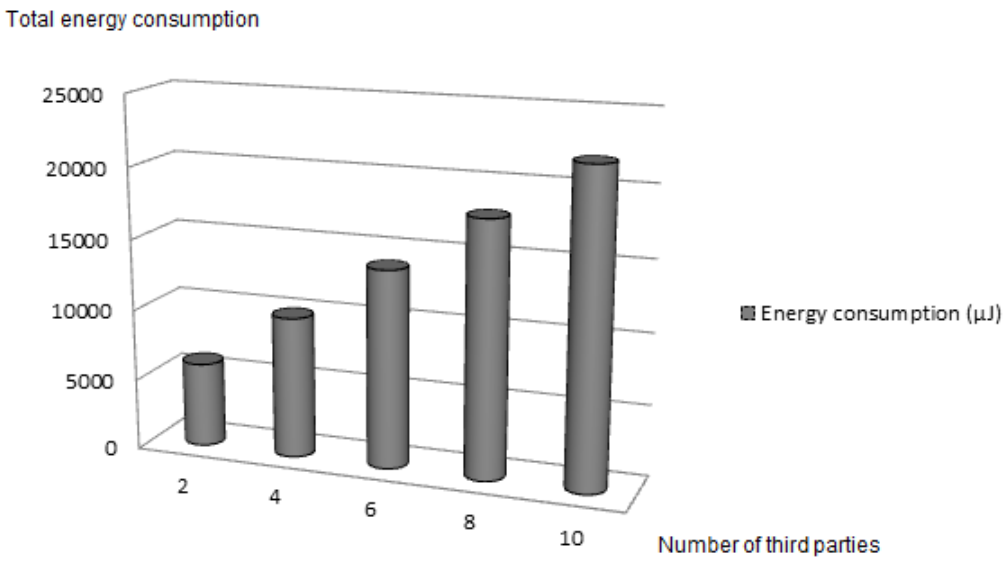

Figure 5-7: Energy consumption evolution on a constrained node (TelosB) considering several third parties

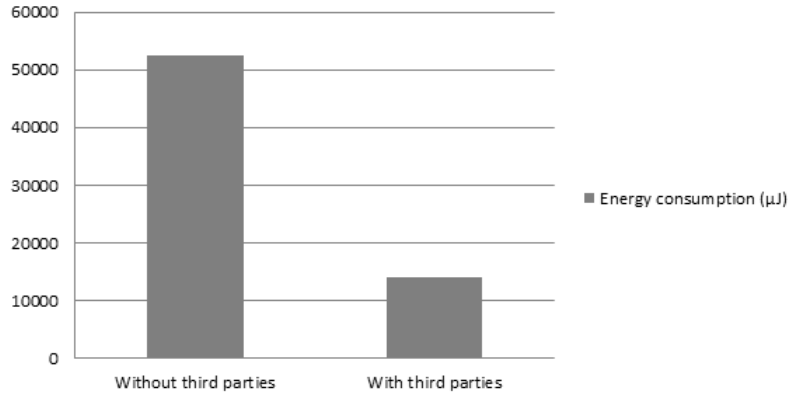

Figure 5-8: Comparison between the average energy consumption of several third parties and the energy consumption when no third party is used

Based on the energy evaluation regarding both communication and computational aspects, we have estimated the overall energy cost considering different numbers of third parties. The results are synthesized in Table 5.9 and plotted in Figure 5-7. We notice an increase in energy consumption with the introduction of more third parties. This is due to the communication overhead caused by the increase of the number of exchanged messages. However, the average energy consumption including several third parties ranging from 2 to 10 third parties is still considerably lower compared to the case with no third party involved (Figure 5-8). This is due to the offloading of asymmetric operations. Indeed, the constrained node is only involved in symmetric operations.

The analysis study has allowed us to validate our protocol from different perspectives. First of all, we have provided a theoretical analysis regarding the different security properties required in our network scenario. Additionally, we have analyzed the resistivity of our protocol against well-known attacks, which can be led against e-health applications. The security properties of our protocol have then been formally validated using Avispa tool. Furthermore, we have proceeded with a quantitative analysis to highlight energy savings. Our estimation has showed the viability of the proposed protocol for e-health applications based on highly constrained nodes. Additionally, as the third parties are not required to be trusted, the end-to-end principle is kept safe. In fact, no intermediate entity is able to retrieve the exchanged secrets. Besides, it is worth noting that as the number of third parties increases, security is consolidated. Indeed, an eventual attacker would need to compromise a more important number of targets. Nevertheless, a reasonable tradeoff needs to be found between the number of third parties and the communication overhead.

5.1.3 Conclusion

The proposed protocol is based on offloading heavy cryptographic primitives to third parties. Highly resource-constrained nodes are then able to establish a shared end-to-end secret with any remote entity, making use of symmetric cryptography. This is achieved through simple message exchanges with third parties, which are much less energy consuming than the actual use of asymmetric cryptographic primitives. We have evaluated our protocol regarding both security aspect and energy cost savings. The results have shown that security properties of our protocol are safe. In addition, considerable energy gains according to the number of used third parties are achieved. Besides, as the number of third parties grows, security is strengthened. Nevertheless, energy consumption increases due to the communication overhead. A viable tradeoff needs to be found between security level and energy consumption. In our last three contributions, we targeted the establishment of a shared secret between two entities. However, it is anticipated that constrained devices in IoT will often operate in groups to achieve a collective monitoring or management task. Hence, securing multicast communications in the IoT is crucial. Our fourth contribution is therefore focused on group key management protocols.

5.2 A Decentralized Batch-based Group Key Management Protocol for Mobile Internet of Things (DBGK)

To secure group communications, several group key management protocols have been introduced. However, the majority of the proposed solutions are not adapted to the IoT and its strong processing, storage, and energy constraints. In this context, we introduce a new decentralized batch-based group key management protocol called DBGK to secure multicast communications in the context

of IoT. Our protocol is designed to take into account resources scarcity and the mobility of IoT devices. To mitigate the single point of failure issue, we opt for a decentralized architecture. In addition, to reduce the *1-affects-n* phenomenon, we consider that each sub group of the network is secured with a different group key. Moreover, we use a time-driven approach where a group key is used in each time slot or interval. Members only request the required keys for each particular interval. As a consequence, less memory is required to store the keys, and only active members are involved in the rekeying process, which further reduces the *1-affects-n* issue. By using a different group key for each area, our protocol allows to handle mobile members. To assess our protocol, we analytically evaluate its security properties and its performances compared to similar schemes proposed in the literature. In addition, we validate this analysis through simulations using Cooja simulator [2]. Our proposed protocol brings the following contributions:

- Mitigates the *1-affects-n* issue. In fact, in addition to the adoption of a decentralized architecture, our protocol only involves the active members in the rekeying process. Doing so, it allows the other members to remain in a sleeping mode, therefore, saving energy.

- The joining process does not require from a new member to store a high number of keys. Instead, based on its storage capabilities and its future behaviour in the group, the new member asks only for the required keys.

- Mobility is inherently handled without any assumption on the node authenticity after the handover.

- The rekeying process is not based on the prediction of the handover or departure time as we assume a highly dynamic IoT network with unpredictable leave and join events.

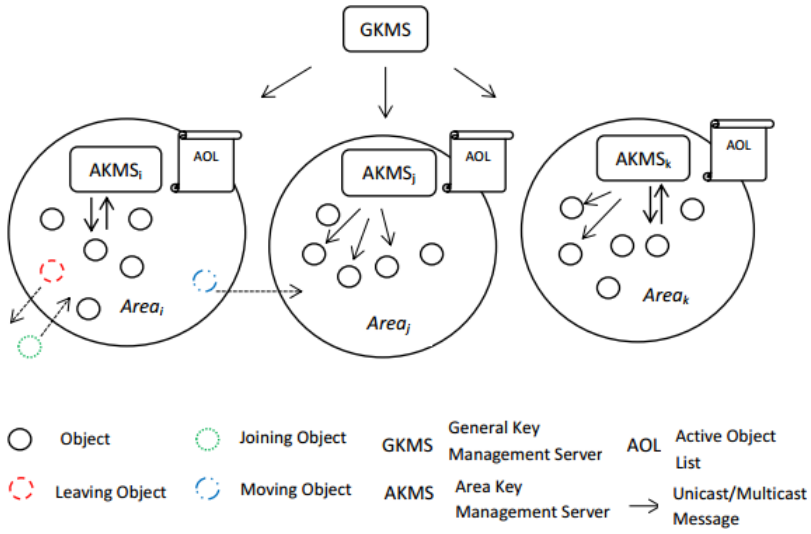

Figure 5-9: Network Model: a decentralized architecture based on an independent group key per area

5.2.1 Network Model

Our network model is divided into several areas. Each area covers a number of objects. An object might be any entity of our daily life able to process, store, and communicate data to other entities and to the Internet. Each area is managed by one Area Key management Server (AKMS) (see Figure 5-9). The AKMS establishes a Traffic Encryption Key (TEK) for each object in the area. Each area has its own TEK that is different from the TEK of other areas. Using the TEK, the objects secure their communications inside the area. In case of an event, the AKMS is responsible of updating the TEK. The event can be triggered by a new object joining the area, an existing object leaving the area, or a moving object between the different areas. In addition, the AKMS maintains a list called Active Object List (AOL), which stores the delivered credentials to the objects for each time slot. These credentials are used during the rekeying process. A General Key Management Server (GMKS) manages the different AKMS, and sets the security policy for the entire group. In particular, it ensures the appropriate access control policy of each area. Moreover, it may act as a backup of a given AKMS if it fails or it is overloaded (e.g. hardware

failure, DoS attack). Besides, different modes of communication (i.e. unicast, multicast, and anycast) are used depending on the protocol execution stage. Our network model is positioned in the category of decentralized architectures with a different TEK for each subgroup. The network is heterogeneous and contains two types of entities with various capabilities both in terms of computing power and energy resources. We distinguish two categories of entities:

- highly resource constrained entities (i.e. the objects), with few MHz of computational power, several kilobytes of RAM, and several tens of kilobytes of ROM.

- entities with high energy, computing power, and storage capabilities (i.e. GKMS and AKMS).

Figure 5-9 provides a general overview of the involved entities along with their interactions. In addition, the used notations are presented in Table 5.10. Before presenting our protocol, we define a set of assumptions.

- the constrained objects are IP-enabled and run 6LoWPAN adaptation layer. Thus, we assume the existence of a gateway (i.e. 6LoWPAN border router) through which 6loWPAN headers are compressed and decompressed.

- we consider the different $AKMS$ and the $GKMS$ as trusted entities.

- we assume that the different entities have successfully established the Key Encryption Keys between them.

- a long term key SK is shared inside each area. It is distributed to each new joining node, and used in the generation of the TEK.

Notation	Description
$GKMS$	General Key Management Server
$AKMS_i$	Area Key Management Server of the area i
AOL_i	Active Object List of the area i
O_i	An object i
$TEK_{i,t}$	Traffic Encryption Key for the area i and the time slot t
$KEK_{O_i,AKMS_j}$	Key Encryption Key shared between O_i and $AKMS_j$
$KEK_{O_i,GKMS}$	Key Encryption Key shared between O_i and $GKMS$
$KEK_{AKMS_i,GKMS}$	Key Encryption Key shared between $AKMS_j$ and $GKMS$
SK	Long term Secret Key shared between the members of each area
$Slot_t$	Interval of time between time t and time $t+1$
$T_{i,t}$	Ticket issued in the area i for the slot t
ID_{O_i}	The identity of the object O_i
$Data_{O_i}$	Information on O_i's storage and processing capabilities
$NSlot$	Number of requested tickets (corresponding to slots)
F	One-way function

Table 5.10: Terminology table

- time is split into several slots of an established length. Time slots units could be seconds, minutes, days, months, etc. A different ticket is associated to each time slot. A ticket is a piece of data used in the generation of the different TEK.

- although an object is able to join the group asynchronously, it has to wait the beginning of the next slot before becoming an active object. In the same way, the leave of a legitimate object should take place at the end of a slot. Hence, we assume a synchronous batch rekeying protocol. This category of protocols introduces a delay, but provides a reduced number of rekeying operations.

- We consider that our network model is clear from interferences.

5.2.2 The proposed protocol

In this section, we detail the different exchanged messages along with their structure. The exchanges are triggered upon the following events (see Figure

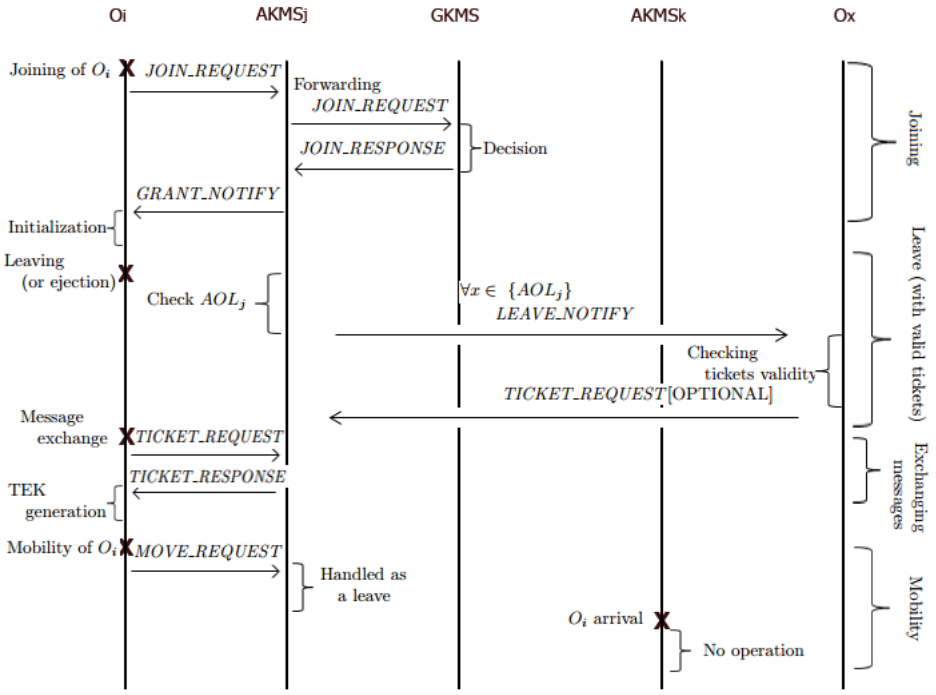

Figure 5-10: The signaling flow of our protocol

5-10).

- *Join:* an object O_i willing to join the group sends a $JOIN\_REQUEST$ message to the anycast address of $AKMS$. The nearest $AKMS$ treats this message. The message has the following structure: $< ID_{O_i}, Data_{O_i} >$. Let us assume that $AKMS_j$ handles the request. This latter forwards $JOIN\_REQUEST$ to $GKMS$. Based on the requesting object specificities, its trust level (if the object has previously been a member of the group), and its access policy, the $GKMS$ decides on whether to deny or grant the access to O_i. $GKMS$ sends $JOIN\_RESPONSE$ message in unicast to $AKMS_j$ containing the decision. If O_i is accepted to join the group, $AKMS_j$ sends $GRANT\_NOTIFY$ to O_i. We assume the occurrence of an initialization phase where O_i securely receives $KEK_{O_i,AKMS_j}$, $KEK_{O_i,GKMS}$, and SK. In fact, an offline dealer is in charge of O_i's initialization. Upon the initialization phase, O_i is considered as a valid and

authenticated member of area j.

- *Exchanging messages:*

  To secure its communications within the group, O_i will have to derive the corresponding $TEK_{j,t}$. This latter is computed as follows:

$$TEK_{j,t} = F(SK, T_{j,t}) \tag{5.1}$$

  F could be implemented as a cryptographic one-way hash function such as SHA-1 [7]. To retrieve the corresponding tickets required to derive $TEK_{j,t}$, O_i sends $TICKET\_REQUEST$. This message contains the following information: the identity of the object, its storage and processing capabilities, and the number of requested tickets. In fact, O_i can request several tickets according to its expected activity during the future slots. $TICKET\_REQUEST$ message is sent in unicast to the corresponding $AKMS_j$. The message has the following structure: $< ID_{O_i}$, $Data_{O_i}$, $NSlot >$. Based on the requesting object specificities, its trust level (if the objects has previously been a member of the group) and its access policy, $AGMS_j$ decides on whether to deny, or grant the access to O_i. Moreover, it decides on the number of tickets to be granted. When $AGKMS_j$ accepts the request, it sends a $TICKET\_RESPONSE$ message to O_i, which generates the corresponding $TEK_{j,t}$ of the different tickets using equation (1).

- *Leave:* in our protocol, the leaving (or ejection) event is handled based on whether the leaving node is in possession of valid tickets or not. In fact, if the leaving object O_i from the area j is not active, and thus not in possession of the current $T_{j,t}$ or the future $T_{j,t+k}$ $(k > 0)$, no rekeying operation is required. Indeed, forward secrecy is ensured as O_i will not be able to derive any $TEK_{j,t}$. Nevertheless, if O_i is in possession of valid tickets, $AKMS_j$ will have to handle a rekeying operation to ensure for-

ward secrecy in the area. $AKMS_j$ maintains a list (i.e. AOL_j). Each entry of the list has the following structure:

$\{T_{j,t} --> List_t\}$

$List_t$ is the list of the objects in possession of $T_{j,t}$. Upon O_i departure, $AKMS_j$ checks its AOL_j looking for the objects that are in possession of the same tickets as those delivered to O_i. Then, $AKMS_j$ sends a multicast $LEAVE\_NOTIFY$ notification to the concerned objects. The message has the following structure:

$< T_{j,t}, T_{j,t+1}, ... T_{j,t+k} >$.

The recipients of the notification will invalid the received tickets, and ignore the notification if the tickets are no longer used. However, if they are still using the tickets, the concerned objects will send a $TICKET\_REQUEST$ to the $AKMS_j$. Consequently, forward secrecy is ensured with a minimum overhead. Only active objects are involved in the rekeying process.

- *Mobility:* an object O_i is considered a mobile object when it moves from an area J to another area K. The process starts by O_i sending a $MOVE\_REQUEST$ message to $AKMS_j$. Let us assume that O_i arrives to area K during time slot x. Even though O_i is in possession of SK, it is not enough to derive $TEK_{k,t}$ with ($t < x$). Therefore, backward secrecy is inherently ensured. Forward secrecy regarding area J is handled as a leave event. Indeed, if O_i is not in possession of a valid ticket, no rekeying operation is required. It is worth mentioning that unlike our protocol, several approaches in the literature [46] [66] ensure forward secrecy based on the following assumption. The moving members remain valid and authentic members of the session even though they left area J. In fact, they are considered as already authenticated with $AKMS_J$.

Although this assumption allows to handle mobility with fewer operations, it constitutes a forward secrecy breach since O_i will still be able to access the communications after its departure. Moreover, if O_i returns to area J, backward secrecy property will also be violated. Indeed, O_i would get an access to the communications of area J that occurred before its joining.

- *$AKMS$ unavailability:* Area Key Manager Servers are in charge of maintaining the keying materials of their respective areas. In case where an $AKMS$ of a particular area is not available (e.g. hardware failure, DoS attack, etc.), the communications inside the group need to stay secured. One of the motivations behind our choice of a decentralized architecture is to ensure a mechanism to avoid the single point of failure issue. Each time an object O_i sends a message to its $AKMS$, it waits for a certain period of time. If no response is received, O_i assumes that its $AKMS$ is unavailable. Therefore, it sends a $SPARE\_REQUEST$ notification to $GKMS$. The structure of the message is as follows:

  $< ID_{O_i}, AKMS >$

  As $KEK_{O_i,GKMS}$ is shared between O_i and $GKMS$, this latter will be able to manage the requests from the affected area until its restoration. $GKMS$ will act as a back-up $AKMS$, hence mitigating the single point of failure issue.

- *Periodical update:* long term attacks could be carried out in an attempt to retrieve the shared long term key SK. To mitigate these attacks, we propose a periodical rekeying of SK for each area. In fact, the operation is similar to the widely treated classical rekeying operation in centralized architectures. We advocate the use of a tree-based hierarchical approach (e.g. LKH [138]) as this type of protocols achieve a reasonable computation and communication overhead (i.e. logarithmic complexity) without

Protocol	messages	keys to transmit
MARKS	1	$\leq 2(\log_2(n) - 1)$
LKH	h	h(h+1)/2-1
Veltri et al	1	$\leq 2(\log_2(n) - 1)$
DBGK	1	1

Table 5.11: Member joining

Protocol	messages	keys to transmit
MARKS	X	X
LKH	(d-1)(h-1)	(d-1)h(h-1)/2
Veltri et al	(d-1)(h-1)	(d-1)h(h-1)/2 + 1
DBGK	$M1 \leq (d-1)(h-1)$	$M2 \leq (d-1)h(h-1)/2$

Table 5.12: Member leaving with a valid ticket

compromising any aspect of security [29].

5.2.3 Analysis

We analysed and compared the performances of our protocol and its security features with MARKS[26], LKH [138], and Veltri et al [132] protocols with respect to security requirements and energy consumption.

5.2.3.1 Security requirements

- *Backward secrecy:* is ensured regarding both joining and mobility events. The joining member is not able to access the communications occurred before its arrival to the group. In fact, each $TEK_{i,t}$ is different from

Protocol	messages	keys to transmit
MARKS	0	0
LKH	(d-1)(h-1)	(d-1)h(h-1)/2
Veltri et al	0	0
DBGK	0	0

Table 5.13: Member leaving without a valid ticket

Protocol	**messages**	**keys to transmit**
MARKS	X	X
LKH	h +(d-1)(h-1)	h(h+1)/2-1 + (d-1)h(h-1)/2
Veltri et al	$1 + (d-1)(h-1)$	$\leq$ $2(\log_2(n)-1)+(d-1)h(h-1)/2+1$
DBGK	$2 + M1(M1 \leq (d-1)(h-1))$	$1 + M2(M2 \leq (d-1)h(h-1)/2)$

Table 5.14: Mobility with a valid ticket

Protocol	**messages**	**keys to transmit**
MARKS	1	$\leq 2(\log_2(n)-1)$
LKH	h +(d-1)(h-1)	h(h+1)/2-1 + (d-1)h(h-1)/2
Veltri et al	1	$\leq 2(\log_2(n)-1)$
DBGK	2	1

Table 5.15: Mobility without a valid ticket

the next $TEK_{i,t+1}$. Therefore, it is impossible for a joining member to decrypt stored messages that have been encrypted with precedent Traffic Encryption Keys. Likewise, upon the movement of a member from an area to another, previously exchanged messages of the destination area are not accessible to the mobile member.

- *Forward secrecy:* two distinct cases are considered in our protocol to ensure forward secrecy. In the first case, the leaving member has no valid ticket $T_{i,t+k}$ $(k \geq 0)$. In this case, forward secrecy is ensured without any rekeying operation. In fact, the decryption of future communications would require that the leaving member has access to a valid ticket, which is impossible after its departure from the group. In the second case, the leaving member is in possession of valid tickets. Forward secrecy is therefore ensured in our protocol by sending a notification message (i.e. $LEAVE\_NOTIFY$) informing the members in possession of the same tickets of their non-validity. The members that are still active ask the involved $AKMS$ for new tickets. Thus, the leaving (or ejected) member will not be able to access future communications in the group.

- *Key independence:* is ensured if a disclosure of a key does not compromise other keys [29]. In our protocol, Traffic Encryption Keys are generated using a one way function that has as an input a long term key SK and a valid ticket $T_{i,t}$. By definition, a one way function ensures that the data used as an input cannot be retrieved from the resulting output (in this case the $TEK_{i,t}$). Thus, the disclosure of one key does not give any extra information to an attacker to retrieve precedent, future, or group keys from other areas.

- *Minimal trust:* our protocol reduces the number of trusted entities to the minimum. In fact, each area is managed by a single trusted Area Key Management, and the whole group is managed by one trusted Key Management Server. The loss of a member due to a failure or an attack does not affect other members. In addition, our protocol relies on the General Key Management Server to handle the loss of an Area Key Manager Server in order to keep the area available.

5.2.3.2 Performances comparison

To compare the performances of our protocol with those of LKH[138], MARKS [26], and Veltri et al protocol [132] on a group of n members, we consider the following parameters.

- *Number of messages (M1)* exchanged from the Key Management Server or the members of the group as a result of an event.

- *Number of transmitted keys (M2)* contained in the exchanged messages following an event. This criteria also shows the amount of cryptographic materials to be stored by the different entities.

We consider a mobility event in the protocols that do not handle it explicitly as two separate events. The first event consists of a leaving from the source

area, and the second one consists of a joining to the destination area. Hence, the mobility cost is estimated as the sum of the two events. It is worth mentioning that MARKS [26] protocol does not handle unpredicted member leaving.

In our protocol, if a leaving member is in possession of valid tickets, all the members that hold the same tickets have to be notified about the non-validity of the involved tickets. Thus, the cost of such operation including both the total number of messages $M1$ to be sent and the number of keys $M2$ might be high. To mitigate this cost, we propose to superimpose the stat-of-the-art LKH [138] protocol to our protocol. As a result, our protocol renews the long-term SK using LKH instead of sending a notification to all the involved members. This solution has been adopted in Veltri et al protocol [132]. This latter uses LKH to rekey the group upon an unpredicted leaving instead of using a basic approach with unicast messages sent to all members. However, in our protocol, this solution is only used in the case where the notification process cost is higher than LKH cost. Doing so, we ensure that the cost of our protocol is less than or equal to LKH cost [138] in the worst case. The decision of using LKH or sending notifications is taken by the corresponding $AKMS$. Indeed, the cost of LKH regarding the number of messages to be sent is: (d-1)(h-1), where d represents the degree of LKH tree, and h represent its height. The $AKMS$ compares the cost of applying LKH with the cost of sending notifications to the involved members, and decides whether to adopt the first solution or the second one accordingly.

The performances of our protocol are compared to the performances of MARKS[26], LKH [138], and Veltri et al [132] protocols relying on the following set of events: joining, member leaving with valid tickets, member leaving without valid tickets, member mobility with valid tickets, and member mobility without valid tickets. Table 2 to table 6 show our analytical results.

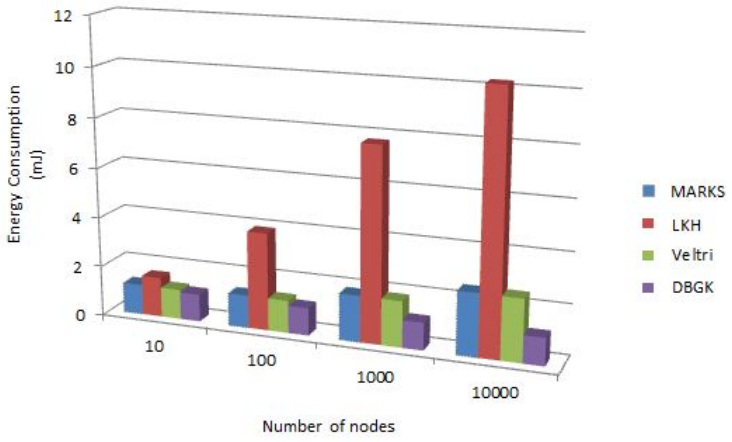

Figure 5-11: Energy consumption upon joining

Based on the obtained analytical results, we evaluate the energy consumption resulting from rekeying operations to compare the performances of our protocol to MARKS[26], LKH [138], and Veltri et al [132] protocols. To do so, we use Cooja simulator of Contiki OS 2.7 [2]. Contiki is an open source operating system for the Internet of Things. Contiki allows tiny, battery-operated low-power systems communicate with the Internet. Contiki is used in a wide variety of systems such as city sound monitoring, street lights, networked electrical power meters, industrial monitoring, and e-health applications. In our simulations, we use Tmote Sky nodes which are based on the CC2420 radio chip and the MSP430 microcontroller (10k RAM, 48k Flash). In addition, we activate the CC2420 AES hardware encryption to encrypt/decrypt the different exchanged messages. We measure the energy consumption using Powertrace tool [41]. This latter provides the time (in ticks) passed by different components of a node (e.g. CPU, transmitting, receiving, etc.) in the active state which allows evaluating the energy consumption when combined with other data (i.e. current draw, voltage) specific to the used platform. Relevant results are plotted in Figure 5-11, 5-12 and 5-13.

Figure 5-11 shows the evolution of the energy consumption resulting from a

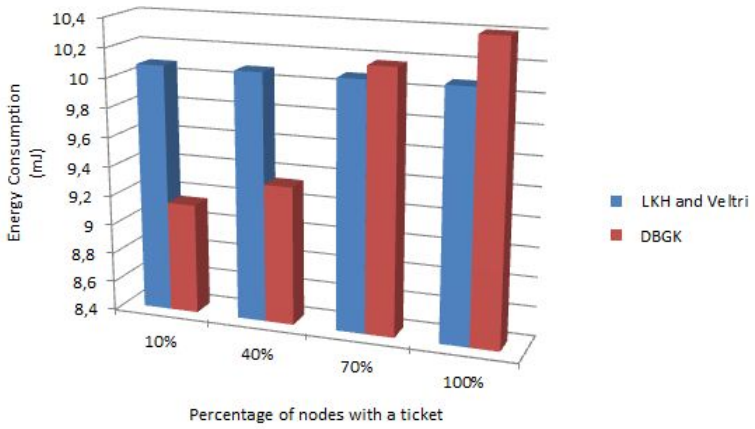

Figure 5-12: Energy consumption upon leave with a valid ticket

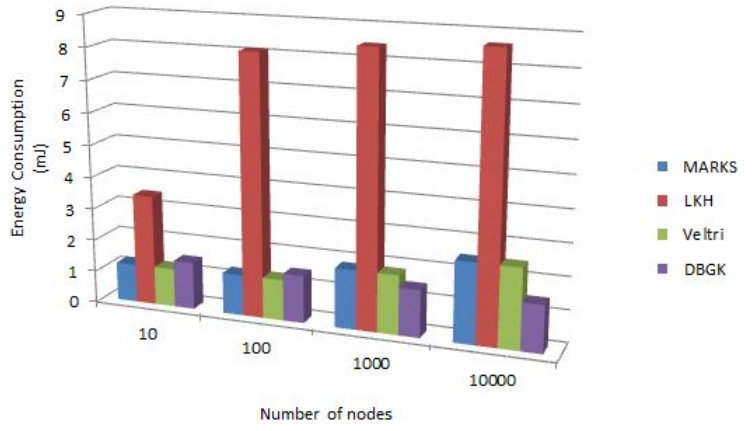

Figure 5-13: Energy consumption upon mobility without a valid ticket

joining event in groups with various sizes. We notice that LKH [138] energy consumption increases with the increase of the number of nodes in a more important scale compared to MARKS[26] and Veltri et al [132]. In fact, both Veltri et al [132] and MARKS[26] are time-driven protocols that only involve one message for the joining member, while LKH [138] is an event-driven protocol that involves several messages in case of a joining event. The energy consumption of our protocol is constant and independent from the number of nodes.

In figure 5-12, we show the obtained results with respect to a member leaving with a valid ticket. To simulate a realistic scenario, we set the number of nodes to 1000 nodes. Then, we measure the energy consumption according to the proportion of nodes that are in possession of the same tickets as those hold by the leaving member. In addition, we consider that the notified members will respond with a request of 20 tickets (i.e. $NSlot = 20$). MARKS[26] is not included in the simulation as it does not handle unpredicted leave events. The results show that the energy consumption of LKH [138] and Veltri et al [132] is constant and it is not affected by the variation of the proportion of responding nodes. However, the energy consumption of our protocol increases with the increase of responding nodes. This is caused by the computation and the communication overheads caused by the exchange of an important number of messages. It is worth noting that the energy consumption of our protocol is lower in up to nearly a proportion of 50 % of nodes in possession of a valid ticket at the moment of the leaving event.

Figure 5-13 illustrates the energy consumption following a mobility event of a node without a valid ticket for groups with various sizes. It shows that the energy consumption of LKH [138] is high for the same reasons invoked in the joining event analysis. Furthermore, it highlights a progressive increase of the energy consumption of both Veltri et al [132] and MARKS[26]. In fact, both

protocols are based on the predicted leave moment of nodes. This requires several keys to be sent. On the other hand, the energy consumption of our protocol is constant and it is not affected by the number of involved nodes. This results in energy gains, in particular, where the size of the group grows.

Relying on security and performances evaluation results, we can prove that our protocol is more adapted to groups with a large number of members that can join the network for a long period of time (without being necessarily active), and leave the network unexpectedly. Unlike Veltri et al protocol that involves all the members during the rekeying operations, even if they have not been active for a long period of time, our protocol allows them to remain in a sleep mode without being interrupted by the rekeying operations. Thus, the energy consumption is lower in up to a proportion of 50 % members of the group with the same tickets hold by the leaving member. If the proportion exceeds 50 %, LKH is used. Hence, the energy consumption of our protocol is less or equal to other protocols in the worst case. Mobility is handled with fewer operations in particular in the case of a mobile node without valid tickets. In addition, forward secrecy is ensured for mobility events without any assumption on the source area.

Our protocol is not based on predicting the leaving moment of the members as IoT will likely contain highly dynamic nodes affected by unplanned events. Consequently, it does not impose a high number of keys to be stored for a joining member as it is the case in Veltri et al protocol. Our protocol offers a flexible mechanism that allows the member to ask for the number of keying materials that suits the most its storage capabilities (i.e. $NSlot$). As a result, we can safely consider the adoption of our protocol for the constrained and highly dynamic IoT applications.

5.2.4 Conclusion

We introduced a novel decentralized batch-based group key management protocol for the Internet of Things. Unlike existing protocols, our rekeying process does not rely on the predicted time of objects leaving. Instead, an on-demand based mechanism is used. Doing so, only active objects are concerned by the rekeying. Moreover, objects storage capabilities are considered, as joining objects are not compelled to store a high number of keys. In fact, objects request the amount of keying materials based on their storage space. Mobility is smoothly handled as backward secrecy is inherently ensured. In addition, forward secrecy is ensured without any rekeying in case where the leaving member is not in possession of valid tickets. Otherwise, it is handled as a leave event in the source area. As a result, unlike several other approaches in the literature, we ensure both backward and forward secrecy with few rekeying messages without any assumptions on the moving member validity in the source area. Furthermore, our protocol mitigates the *1-affects-n* issue. Indeed, not only each rekeying affects only the concerned area but also in each area only the active members with valid tickets are concerned. We evaluated our protocol regarding its security properties and energy cost savings using both analytical and simulation experiments. The obtained results show that our protocol performs better than other existing protocols in the literature, while keeping its security properties safe.

General Conclusion

Ensuring data confidentiality in IoT is an essential requirement for a large and successful deployment of its applications and services. Sensitive applications such as e-health are highly vulnerable to security attacks. In fact, any breach in the confidentiality of private data would seriously endanger the privacy of users and therefore threaten the adoption of IoT applications. Using classical key management protocols to secure IoT might quickly become challenging. Indeed, several IoT specificities hinder the deployment of existing approaches such its highly dynamicity, scalability issues, heterogeneity, and resources constraints. Addressing previously mentioned challenges has been the subject of this thesis.

Throughout this thesis, we have provided an overview on the IoT concept, its applications, its building blocks technologies along with the main proposed architectures, that are considered well-adapted to suit IoT requirements. We have also introduced a classification highlighting the suitability of the proposed architectures to IoT characteristics. In addition, we have underlined the main shortcomings of the current approaches and proposed our vision regarding the IoT's future architecture based on the current state of the art. Furthermore, we focused on security issues by clearly defining and highlighting the differences between the involved security concepts.

We categorized our contributions into two main categories of approaches, namely, the centralized approaches, and the decentralized approaches. In the centralized approaches category, we introduced our first protocol where Public Key primitives are only involved at the registration phase to establish a symmetric session key, which is used to establish a secure channel between the different nodes and the base station. Analysis results showed a gain in enregy consumption compared to other existing schemes. In our second protocol, we propose a

tailoring of Mikey-Ticket to adapt it to low-power and constrained environment of e-health devices and applications. Analysis results with respect to security and energy saving aspects demonstrate that our approach keeps Mikey-Ticket safe while considerable amount of energy is saved.

In the decentralized approaches category, we introduced our third protocol. This latter allows constrained entities to establish a secret with any remote entity in an end to end way. It is based on the offloading heavy cryptographic primitives to third parties. The obtained results showed that security properties of our protocol are safe. In addition, considerable energy gains according to the number of used third parties are achieved. In our fourth protocol, we address key management in multicast communication scenarios for IoT taking into account the mobility aspect. The analytical evaluation of the proposed protocol regarding its security properties and energy cost savings show interesting performances compared to existing schemes.

By the end of this thesis, we consider several new research directions as a continuation of our different contributions. Firstly, following our study on IoT architectures, we plan to design a suitable approach to deal with the different challenges of the IoT at each layer of the network. Secondly, following our approach on Mikey-Ticket protocol, we intend to study the applicability of our tailored version for group communication scenarios, and the eventual impact of mobility. Thirdly, following our distributed end to end key management protocol, we are investigating the possibility of developing a lightweight trust model to allow constrained nodes automatically select effective third parties. Fourthly, we aim to assess our proposed group key management protocol regarding different realistic mobility models. In fact, we intend to implement our approaches on real test beds in order to compare the obtained results with our analytical evaluations.

List of Publications

National communication

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui
- Title: Securing e-Health Applications in the Context of Internet of Things: Tailoring Mikey Ticket
- Conference: Conference sur l Ingenierie Informatique (C2i)
- Location/date: Ecole Militaire Polytechnique (Bordj El Bahri, Algiers), 17/12/2014
- Preceedings: Page 22

International communication 1

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui
- Title: A Lightweight Key Management Scheme for E-health applications in the context of Internet of Things
- Conference: International Conference on Next Generation Computing and Communication Technologies (ICNGCCT 2014)
- Location/date: Dubai (UAE), 24/04/2014
- Preceedings: Pages 47-52. ISBN: 978-93-83303-42-7

International communication 2

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui
- Title:Tailoring Mikey-Ticket to e-Health Applications in the Context of Internet of Things

- Conference: International Conference on advanced Networking, Distributed Systems and Applications (INDS 2014)
- Location/date: Bejaia (Algeria), 18/06/2014
- Preceedings: Pages 72-77 (Short Papers Proceedings)
- Link: https://inds-2014.hds.utc.fr/

International communication 3

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui
- Title: A Cooperative End to End Key Management Scheme for E-health Applications in the Context of Internet of Things
- Conference: ADHOC-NOW 2014 Workshops
- Location/date: Benidorm (Spain), 24/06/2014
- Preceedings: Pages 35-46 (LNCS 8629)
- Link: http://link.springer.com/chapter/10.1007

International communication 4

- Authors: Faiza Medjek, Djamel Tandjaoui, Mohammed Riyadh Abdmeziem, Nabil Djedjig
- Title: Analytical evaluation of the impacts of Sybil attacks against RPL under mobility
- Conference: The 12 th International Symposium on Programming and Systems (ISPS)
- Location/date: Algiers (Algeria), 29/05/2015

- Preceedings: Pages 1-9

- Link: http://ieeexplore.ieee.org/Xplore/home.jsp

International communication 5

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui, Imed Romdhani

- Title: A Decentralized Batch-based Group Key Management Protocol for Mobile Internet of Things (DBGK)

- Conference: The 14th IEEE International Conference on Ubiquitous Computing and Communications (IUCC-2015)

- Location/date: Liverpool (UK), 26/10/2015

- Preceedings: Pages 1109 - 1117

- Link: http://ieeexplore.ieee.org/Xplore/home.jsp

Book chapter

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui, Imed Romdhani

- Title: Architecting the Internet of Things: State of the Art

- Series Title: Studies in Systems, Decision and Control

- Book title: Robots and Sensor Clouds

- Pages: 55-75

- Link: http://link.springer.com/chapter/10.1007/978-3-319-22168-7_3

International journal

- Authors: Mohammed Riyadh Abdmeziem, Djamel Tandjaoui
- Title: An end-to-end secure key management protocol for e-health applications
- Journal: Computers & Electrical Engineering
- Special issue: Emerging Research in Internet of Things
- Volume: 44 (May 2015)
- Pages: 184-197
- Link: http://www.sciencedirect.com/science/article/pii/S0045790615001238

Bibliography

[1] Avispa a tool for automated validation of internet security protocols. *http://www.avispa-project.org*.

[2] The contiki operating system. *http://www.contiki-os.org*.

[3] Murphi model checker. *http://www.cs.utah.edu*.

[4] Prism - a probabilistic model checker. *http://www.prismmodelchecker.org*.

[5] Routing over low power and lossy networks (roll). IETF, http://datatracker.ietf.org/wg/roll/charter.

[6] Ieee std 802.15.4, 2003. IEEE Comp. Soc.

[7] Secure hash standard (shs). *Federal Information Processing Standards Publication*, pages FIPS 180–3, 2008. csrc.nist.gov.

[8] Sensei project, 2010. http://www.ict-sensei.org/.

[9] M.R. Abdmeziem and D. Tandjaoui. A cooperative end to end key management scheme for e-health applications in the context of internet of things. In *Ad-hoc Networks and Wireless*, pages 35–46. Springer, 2014.

[10] M.R. Abdmeziem and D. Tandjaoui. A lightweight key management scheme for e-health applications in the context of internet of things. In *International Conference on Next Generation Computing and Communication Technologies (ICNGCCT 2014)*, 2014.

[11] M.R. Abdmeziem and D. Tandjaoui. Tailoring mikey-ticket to e-health applications in the context of internet of things. In *International Conference on Advanced Networking, Distributed Systems and Applications (Short Papers)*, pages 72–77, June 2014.

[12] M.R. Abdmeziem and D. Tandjaoui. An end-to-end secure key management protocol for e-health applications. *Computers & Electrical Engineering*, 2015.

[13] M.R. Abdmeziem, D. Tandjaoui, and I. Romdhani. A decentralized batch-based group key management protocol for mobile internet of things (dbgk). In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 1109–1117, Oct 2015.

[14] M.R. Abdmeziem, D. Tandjaoui, and I. Romdhani. Architecting the internet of things: State of the art. In *Robots and Sensor Clouds*, pages 55–75. Springer, 2016.

[15] G. Adl, A. Szekely, and S. Tillich. The energy cost of cryptographic key establishment in wireless sensor networks (extended abstract). *ASIACCS 07*, pages 380–382, March 2007.

[16] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002.

[17] Moshaddique Al Ameen, Jingwei Liu, and Kyungsup Kwak. Security and privacy issues in wireless sensor networks for healthcare applications. *J. Med syst*, 36:93–101, 2012.

[18] J. Arkko, F. Lindholm, M. Naslund, and K. Norrman. Mikey: Multimedia internet keying. *RFC 3830, IETF*, 2004.

[19] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, pages 2787–2805, May 2010.

[20] L. Atzori, A. Iera, G. Morabito, and M. Nitti. The social internet of things (siot) when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks, Volume 56, Issue 16*, pages 3594–3608, November 2012.

[21] D. Balenson, D. McGrew, and A. Sherman. Key management for large dynamic groups: One-way function trees and amortized initialization. *Internet-Draft*, February 1999.

[22] A. Ballardie. Scalable multicast key distribution. *RFC 1949*, May 1996.

[23] N. W. Bergmann and P.J. Robinson. Server-based internet of things architecture. *The 9th Annual IEEE Consumer Communications and Networking Conference*, 2012.

[24] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi. Secure communication for smart iot objects: Protocol stacks, use cases and practical examples. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–7. IEEE, 2012.

[25] BRIDGE. Bridge: Building radio frequency identification solutions for the global environment. 2009. http://www.bridge-project.eu.

[26] B. Briscoe. Marks: Zero side effect multicast key management using arbitrarily revealed key sequences. *Networked Group Communication*, pages 301–320, 1999.

[27] N. Bui. Internet of things architecture. Technical report, Project co-funded by the European Commission within the Seventh Framework Program, 2011.

[28] CASAGRAS. Casagras project. 2009. http://www.rfidglobal.eu.

[29] Y. Challal and H. Seba. Group key management protocols: A novel taxonomy. *International Journal of Information Technology*, 2(1):105–118, 2005.

[30] A. Charu and T. Mathieu. Validating integrity for the ephemerizerâĂŹs protocol with cl-atse. pages 21–32, 2009.

[31] S. Cherukuri, K.K. Venkatasubramanian, and S.K.S. Gupta. Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. *Parallel Processing Workshops Proceedings, International Conference, 2003*, pages 432–439, October 2003.

[32] Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, J. Mantovani, and and L. Vigneron S. Modersheim. A high level protocol specification language for industrial security sensitive protocols. *Proc. SAPS 04. Austrian Computer Society*, 2004.

[33] G.H. Chiou and W.T. Chen. Secure broadcasting using the secure lock. *IEEE Transactions on Software Engineering*, 15(8):929–934, 1989.

[34] C. Chun, H. Daojing, C. Sammy, B. Jiajun, G. Yi, and F. Rong. Lightweight and provably secure user authentication with anonymity for the global mobility network. *International Journal of Communication Systems*, 24(3):347–362, 2011.

[35] EM. Clarke, O. Grumberg, and DA. Peled. Model checking. *MIT Press: Cambridge*, 1999.

[36] Schurgers Curt and M.B. Srivastava. Energy efficient routing in wireless sensor networks. *IEEE Military Communications Conference MILCOM. Communications for Network-Centric Operations: Creating the Information Force*, 1:357–361, 2001.

[37] B. Daghighi, M.L.M. Kiah, S. Shamshirband, and M.H.U. Rehman. Toward secure group communication in wireless mobile environments: Issues, solutions, and challenges. *Journal of Network and Computer Applications*, 50:1–14, 2015.

[38] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier. The internet of things for ambient assisted living. In *Information Technology: New Generations (ITNG)*, pages 804–809, April 2010.

[39] D. Dolev and C.C. Yao. On the security of public key protocols. *FOCS, IEEE, 1981*, page 350âĂŞ357, 1981.

[40] J.R. Douceur. The sybil attack. *Peer-to-peer Systems*, 2002.

[41] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. 2011.

[42] M. Dworkin. Recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality. *SP-800-38c, NIST, US department of commerce*, 2007.

[43] N. Modadugu E. Rescorla. Datagram transport layer security version 1.2. *RFC 6347, IETF*, 2011.

[44] L. Eschenauer and V.D. Gligor. A key management scheme for distributed sensor networks. *Ninth ACM conference on computer and communications security*, pages 41–47, 2002.

[45] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk. Server-based certificate validation protocol(scvp). *Internet proposed standard RFC*, 5055, 2007.

[46] S. Gharout, A. Bouabdallah, Y. Challal, and M. Achemlal. Adaptive group key management protocol for wireless communications. *J. UCS*, 18(6):874–898, 2012.

[47] Y. Glouche and T. Genet. Span âĂŞ a security protocol animator for avispa âĂŞ user manual. *http ://www.irisa.fr/lande/genet/span/, 2006*, 2006. IRISA/UniversitÃľ de Rennes 1.

[48] J. Granjal, E. Monteiro, and J. Sa Silva. Enabling network-layer security on ipv6 wireless sensor networks. *Proc. of IEEE GLOBECOM, 2010*, 2010.

[49] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems, Volume 29, Issue 7*, page 24, 2007.

[50] N. Gura, A. Patel, A. Wander, H. Elberle, and S.C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. *Proceedings of the Sixth Workshop on Cryptographic Hardware and Embedded Systems (CHESâĂŹ04)*, pages 119–132, 2004.

[51] H. Hada and J. Mitsugi. Epc based internet of things architecture. *IEEE International Conference on RFID-Technologies and Applications*, pages 527–532, 2011.

[52] Y. Hanna, H. Rajan, and W. Zhang. Slede: A domain specific verification framework for sensor network security protocol implementations. *Proceeding of the ACM Conference on Wireless Network Security (WiSecâĂŹ08)*, pages 109–118, 2008.

[53] H. Harney and C. Muckenhirn. Group key management protocol (gkmp) architecture. *RFC 2093*, July 1997.

[54] Mohammad Mehedi Hassan, Biao Song, and Eui-Nam Huh. A framework of sensor-cloud integration opportunities and challenges. *ACM, ICUIMC 09*, January 2009.

[55] Michael Healy, Thomas Newe, and Elfed Lewis. *Smart Sensors and Sensing Technology*, chapter Analysis of Hardware Encryption Versus Software Encryption on Wireless Sensor Network Motes, pages 3–14. Springer Berlin Heidelberg, 2008.

[56] Guoqiang Hu, Wee Peng Tay, and Yonggang Wen. Cloud robotics: Architecture, challenges and applications. *IEEE Network*, June 2012.

[57] J. Hui and P. Thubert. Compression format for ipv6 datagrams over ieee 802.15.4-based networks. *RFC 6282, IETF, 2011*, 2011.

[58] R. Hummen, J. Hiller, M. Henze, and K. Wehrle. Slimfit - a hip dex compression layer for the ip-based internet of things. *WiMob, IEEE*, pages 259–266, 2013.

[59] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle. 6lowpan fragmentation attacks and metigation mechanisms. *Proc. 6th ACM Conf. Security Privacy Wireless Mobile Networks*, pages 55–66, Apr 2013.

[60] R. Hummen, H. Wirtz, J.H. Ziegeldorf, J. Hiller, and K. Wehrle. Tailoring end-to-end ip security protocols to the internet of things. In *21st International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, 2013.

[61] R. Hummen, J.H. Ziegeldorf, H. Shafagh, S.Raza, and K. Wehrle. Towards viable certificate-based authentication for the internet of things. *HotWiSec '13 Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*, pages 37–42, 2013.

[62] I. Ishaq, J. Hoebeke, I. Moerman, and P. Demeester. Internet of things virtual networks. *IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing*, pages 293–300, 2012.

[63] R. Istepanian, A. Jara, A. Sungoor, and N. Philips. Internet of things for m-health applications (iomt). *AMA-IEEE medical technology conference on individualized healthcare, Washington*, 2010.

[64] S.S. Javadi and M.A. Razzaque. Security and privacy in wireless body area networks for health care applications. *Wireless Networks and Security*, pages 165–187, 2013.

[65] J.Vasseur and A. Dunkels. *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.

[66] S. Kamat, S. Parimi, and D.P. Agrawal. Reduction in control overhead for a secure, scalable framework for mobile multicast. *IEEE International Conference on Communications, ICC'03*, 1:98–103, 2003.

[67] K. Kamei, S. Nishio, and N. Hagita. Cloud networked robotics. *IEEE Network*, may/june 2012.

[68] J.P. Kaps and B. Sunar. Energy comparison of aes and sha-1 for ubiquitous computing. *Emerging Directions in Embedded and Ubiquitous Computing, Lecture Notes in Computer Science*, pages 372–381, 2006.

[69] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. *Second ACM Conference on Embedded Networked Sensor Systems*, pages 162–175, november 2004.

[70] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, 1(2):293–315, 2003.

[71] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen. Internet key exchange protocol version 2 (ikev2). *RFC 5996, IETF*, 2010.

[72] S.L. Keoh, S.S. Kumar, and H. Tschofenig. Securing the internet of things: A standardization perspective. *IEEE INTERNET OF THINGS JOURNAL*, 2014.

[73] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, XI:161–191, January 1883.

[74] Y. Kim, A. Perrig, and G. Tsudik. Tree-based group key agreement. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):60–96, 2004.

[75] M. Kovatsch, S. Mayer, and B. Ostermaier. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things. In *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pages 751–756. IEEE, 2012.

[76] T. Landstra, S. Jagannathan, and M. Zawodniok. Energy-efficient hybrid key management protocol for wireless sensor networks. *International Journal of Network Security*, 9(2):121–134, September 2009.

[77] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo. 6lowpan: a study on qos security threats and countermeasures using intrusion detection system approach. *International Journal of Communication Systems*, 25(9), 2012.

[78] P. Lee, J. Lui, and D. Yau. Distributed collaborative key agreement and authentication protocols for dynamic peer groups. *Networking, IEEE/ACM Transactions on*, 14(2):263–276, 2006.

[79] Ming Li and Wenjing Lou. data security and privacy in wireless body area networks. *Wireless Technologies for E-healthcare*, February 2010.

[80] S. Lim, T.H. Oh, Y.B Choi, and T. Lakshman. Security issues on wireless body area network for remote healthcare monitoring. *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), IEEE International Conference*, pages 327 – 332, February 2010.

[81] D. Malan, M. Welsh, and M. Smith. A public-key infrastructure for key distribution in tiny os based on elliptic curve cryptography. *First annual IEEE communications society conference on sensor and ad hoc communications and networks, 2004*, pages 71–80, 2004.

[82] M. Mana, M. Feham, and B. A. Bensaber. Trust key management scheme for wireless body area networks. *International Journal of Network Security, 2011*, 12(2):71–79, March 2011.

[83] M. Marino and U. Caterina. Formal analysis of facebook connect single sign-on authentication protocol. 11:22–28, 2011.

[84] J. Mattsson and T. Tian. Mikey-ticket: Ticket-based modes of key distribution in multimedia internet keying (mikey). *RFC 6043, IETF, 2011*, 2011.

[85] C. M. Medaglia and A. Serbanati. An overview of privacy and security issues in the internet of things. *The internet of things*, pages 389–395, 2010.

[86] F. Medjek, D. Tandjaoui, M.R. Abdmeziem, and N. Djedjig. Analytical evaluation of the impacts of sybil attacks against rpl under mobility.

In *International Symposium on Programming and Systems*. IEEE, April 2015.

[87] A. Mehdizadeh, F. Hashim, and M. Othman. Lightweight decentralized multicast–unicast key management method in wireless ipv6 networks. *Journal of Network and Computer Applications*, 42:59–69, 2014.

[88] S.S.M. Meingast, T. Roosta, and Elfed Lewis. Security and privacy issues with health-care information technology. *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5453–5458, August 2006.

[89] G. De Meulenaer, F. Gosset, F.X Standaert, and O. Pereira. On the energy cost of communication and cryptography in wireless sensor networks. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communication*, pages 580–585, 2008.

[90] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, pages 1497–1516, April 2012.

[91] S. Moedersheim and P.H. Drielsma. Avispa project deliverable d6.2: Specification of the problems in the high-level specification language. http://www.avispa-project.org, 2003.

[92] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of ipv6 packets over ieee 802.15.4 networks. *RFC 4944, IETF, 2007*, 2007.

[93] N. Nasser and Y. Chen. Seem: Secure and energy-efficient multipath routing protocol for wireless sensor networks. *Computer Communications*, 30(11):2401–2412, 2007.

[94] B. Nath, F. Reynolds, and R. Want. Rfid technology and applications. *IEEE Pervasive Computing*, 5(1):22–24, 2006.

[95] H. S. Ng, M.L. Sim, and C.M. Tan. Security issues of wireless sensor networks in healthcare applications. *BT Technology Journal*, 24(2):138–144, 2006.

[96] P. Papazoglou and D. Georgakopoulos. Service oriented computing. *Communications of the ACM*, 46(10), October 2003.

[97] M. Patel and J. Wang. Applications, challenges, and prospective in emerging body area networking technologies. *Wireless Commun*, 17:80âĂŞ88, 2010.

[98] Y. Piao, J. Kim, U. Tariq, and M. Hong. Polynomial-based key management for secure intra-group and inter-group communication. *Computers & Mathematics with Applications*, 65(9):1300–1309, 2013.

[99] C.C.Y. Poon, Yuan-Ting Zhang, and Shu-Di Bao. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *Communications Magazine, IEEE*, 4:73–81, April 2006.

[100] E.D. Poorter, I. Moerman, and P. Demeester. Enabling direct connectivity between heterogeneous objects in the internet of things through a network-service-oriented architecture. *Journal on Wireless Communications and Networking, volume 2011, Issue 1*, August 2011.

[101] Madhuri Prashar and Rajeev Vashisht. Survey on pre-shared keys in wireless sensor network. *Internationl Journal for Science and Emerging Technologies with Latest Trends*, 4(1):42–48, 2012.

[102] S.M.K. Raazi, Z. Pervez, and S. Lee. Key management schemes of wireless sensor networks: A survey. *In A.K. Pathan (ed). Security of Self-Organizing Networks: MANET, WSN, WMN, VANET, CRC Press, 2011*, pages 297–316.

[103] K. Raeburn. Advanced encryption standard (aes) encryption for kerberos 5. *RFC 3962*, 2005.

[104] S. Rafaeli and D. Hutchison. Hydra: a decentralized group key management. *11th IEEE International WETICE: Enterprise Security Workshop*, June 2002.

[105] S. Rafaeli and D. Hutchison. A survey of key management for secure group communication. *ACM Computing Surveys (CSUR)*, 35(3):309–329, 2003.

[106] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig. Securing communication in 6lowpan with compressed ipsec. *in Proc. of IEEE DCOSS, 2011*, 2011.

[107] S. Raza, S. Duquennoy, and G. Selander. Compression of ipsec ah and esp headers for constrained environments. *draft-raza-6lowpanipsec-00 (WiP), IETF, 2013*, 2013.

[108] S. Raza, D. Trabalza, and T. Voigt. 6lowpan compressed dtls for coap. *in Proc. of IEEE DCOSS*, 2012.

[109] S. Raza, L. Wallgren, and T. Voigt. Svelte: Real-time intrusion detection in the internet of things. *Ad hoc networks*, 11(8), 2013.

[110] Shahid Raza, Thiemo Voigt, and Vilhelm Jutvik. Lightweight ikev2: A key management solution for both compressed ipsec and ieee 802.15.4 security. *IETF/IAB workshop on Smart Object Security*, 2012.

[111] E. Rescorla. Diffie-hellman key agreement method. *RFC2631*, 1999.

[112] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos. key management systems for sensor networks in the context of internet of things. *Computers and Electric Engineering*, 37:147–159, 2011.

[113] Rodrigo Roman, Pablo Najera, and Javier Lopez. Securing the internet of things. *IEEE Computer*, 44:51–58, september 2011.

[114] I. Romdhani, M. Kellil, L. Hong-Yon, A. Bouabdallah, and H. Bettahar. Ip mobile multicast: Challenges and solutions. *Communications Surveys Tutorials, IEEE*, 6(1):18–41, First 2004.

[115] Antonio Ruiz-Martínez, C. Inmaculada Marín-López, Laura Baño-López, and AF Skarmeta. A new fair non-repudiation protocol for secure negotiation and contract signing. page 16, 2006.

[116] Somia Sahraoui and Azeddine Bilami. Efficient hip-based approach to ensure lightweight end-to-end security in the internet of things. *Computer Networks*, 91:26–45, 2015.

[117] Y. B. Saied and A. Olivereau. D-hip: A distributed key exchange scheme for hip-based internet of things. *in Proc. of IEEE WoWMoM, 2012*, 2012.

[118] Y. B. Saied and A. Olivereau. Hip tiny exchange (tex): A distributed key exchange scheme for hip-based internet of things. *in Proc. of ComNet, 2012*, 2012.

[119] Y. B. Saied and A. Olivreau. (k, n) threshold distributed key exchangefor hip based internet of things. *in Proc. of ACM MobiWac, 2012*, 2012.

[120] C. Schroth. The internet of services: Global industrialization of information intensive services. In *In Proceedings of 2nd IEEE International Conference on Digital Information Management, ICDIM '07.*

[121] S. Setia, S. Koussih, S. Jajodia, and E. Harder. Kronos: A scalable group re-keying approach for secure multicast. *Proceedings IEEE Symposium on Security and Privacy*, pages 215–228, 2000.

[122] C.E. Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[123] Z. Shelby and C. Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011.

[124] Z. Shelby, K. Hartke, and C. Bormann. The constrained application protocol (coap). *RFC 7252*, June 2014.

[125] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *Wireless Communications, IEEE*, 20(6):91–98, 2013.

[126] R Shirey. Rfc 4949: Internet security glossary. 2007.

[127] Robert Shirey. Rfc 2828: Internet security glossary. *The Internet Society*, page 13, 2000.

[128] J. Song, S. Han, K.A. Mok, D. Chen, M. Lucas, and M. Nixon. Wirelesshart: Applying wireless technology in real-time industrial process control. In *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS'08. IEEE*, pages 377–386. IEEE, 2008.

[129] P. Szczechowiak, L.B Oliveira, M. Scott, M. Collier, and R. Dahab. Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks. In *Proceedings of the 5th European conference on Wireless Sensor Networks*, pages 305–320, 2008.

[130] L. Tobarra, D. Cazorla, F. Cuartero, G. Diaz, and E. Cambronero. Model checking wireless sensor network security protocols: Tinysec + leap + tinypk. *Telecommunication Systems*, 40(3-4):91–99, 2009.

[131] N. Tsiftes and A. Dunkels. A database in every sensor. *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 316–332, 2011.

[132] L. Veltri, S. Cirani, S. Busanelli, and G. Ferrari. A novel batch-based group key management protocol applied to the internet of things. *Ad Hoc Networks*, 11(8):2724–2737, 2013.

[133] A. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Third IEEE International Conference on Pervasive Computing and Communications. PerCom 2005*, pages 324–328. IEEE, 2005.

[134] R. Want. An introduction to rfid technology. *Pervasive Computing, IEEE*, 5(1):25–33, 2006.

[135] R.H. Weber. Internet of things, new security and privacy challenges. *Computer Law and Security Review, 2010*, 26:23–30, January 2010.

[136] Y. Wei and B. Blake. Service-oriented computing and cloud computing. *IEEE INTERNET COMPUTING*, 2010.

[137] Suh Sang Won, Aaron M. Hamby, and Raymond A. Swanson. Hypoglycemia, brain energetics, and hypoglycemic neuronal death. *Glia*, 55(12):1280–1286, 2007.

[138] C.K. Wong, M. Gouda, and S.S. Lam. Secure group communications using key graphs. *Networking, IEEE/ACM Transactions*, 8(1):16–30, 2000.

[139] M. Wu, T.J. Lu, F.Y. Ling, J. Sun, and H.Y. Du. Research on the architecture of internet of things. In *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. IEEE, 2010.

[140] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. *INFOCOM. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1567–1576, 2002.

[141] M. Yun and B. Yuxin. Research on the architecture and key technology of internet of things (iot) applied on smart grid. *Advances in Energy Engineering (ICAEE)*, pages 69 – 72, 2010.

[142] L. Zhang and Z. Wang. Integration of rfid into wireless sensor networks: Architectures, opportunities and challenging problems. In *Grid and Cooperative Computing Workshops. GCCW'06.*, pages 463–469. IEEE, 2006.

[143] J. Zhou, T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, C. Yu, H. Jin, and L.T. Yang. Cloudthings: A common architecture for integrating the internet of things with cloud computing. In *17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 651–657. IEEE, 2013.