Distributed Security Paradigm for Resource-constrained Wireless Sensors in the Context of Internet of Things (IoT)

A Dissertation

Presented to the

Graduate Faculty of the

University of Louisiana at Lafayette

In Partial Fulfillment of the

Requirements for the Degree

Doctor of Philosophy

Muhammad Aamir Iqbal

Spring 2017

Distributed Security Paradigm for Resource-constrained Wireless Sensors
in the Context of Internet of Things (IoT)

Muhammad Aamir Iqbal

APPROVED:

Magdy A. Bayoumi, Chair
Professor of Computer Engineering
The Center for Advanced Computer Studies

Dmitri Perkins
Professor of Computer Science
The Center for Advanced Computer Studies

Ashok Kumar
Associate Professor of Computer Science
The Center for Advanced Computer Studies

Khalid Elgazzar
Assistant Professor of Computer Science
The Center for Advanced Computer Studies

Mary Farmer-Kaiser
Dean of the Graduate School

DEDICATION

To my parents and wife

ACKNOWLEDGMENTS

This work would not have been possible without the support and assistance of many people. First, I would like to express my deepest gratitude to my advisor, Dr. Magdy A. Bayoumi, a worldwide-known researcher, for his outstanding supervision and constant support during my graduate studies. Besides the research and his valuable insights on science and technology, I am impressed by his leadership abilities with great administrative and communication skills that will certainly help me to become a confident, successful research scientist.

I would like to acknowledge my dissertation committee members, Dr. Dmitri Perkins and Dr. Ashok Kumar, for their valuable suggestions and feedback that helped me to improve my dissertation and the defense presentation. Also, I want to thank Dr. Khalid Elgazzar for his suggestions and research work directions input that led to some significant optimizations. I would like to thank all graduate research students in the VLSI lab, CACS for the technical assistance, research discussions, computing help, tools setups, and other numerous activities in which we remained involved. The staff at CACS, the Graduate School, the Office of International Affairs at the University, and all other friends in Lafayette also helped and contributed to making this journey enjoyable, memorable, and possible.

I am deeply obliged to my parents. Their unconditional love and generous sacrifices made me courageous to pursue my dreams throughout my life. Last, but certainly not least, I would also like to thank my other family members and wife, Ayesha Aamir, for their encouragement, inspiration, and persistent care.

Finally, thanks to the One above all of us, the omnipresent God, for giving me the strength to overcome challenges and for answering my prayers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Introduction

1.1 Introduction

The world is undergoing a dramatic, rapid transformation from isolated systems to ubiquitous Internet-based-enabled “things” capable of interacting with each other and generating data that can be analyzed to extract valuable information. This highly interconnected global network structure, known as Internet of Things, will enrich everyone’s life, increase business productivity, improve government efficiency, and the list goes on. However, this new reality (IoT), built on the basis of the internet, contains new kinds of challenges, especially from a security and privacy perspective. IoT devices communicate among themselves with little human interaction; mutual authentication is a crucial aspect of the paradigm. In such an intelligent, vibrant system, sensors are connected to send useful information and control instructions via distributed sensor networks. Wireless sensors have an easy deployment and better flexibility of devices, contrary to wired setup. To facilitate these connections, research and industry have come up with a number of low powers physical and transport network protocols such as Zigbee, Bluetooth, IEEE 802.15.4, and, more recently, low-power Wi-Fi and 6LoWPAN. These developments help towards integrating smart things into a network of internet. The future internet will be an IPv6 network interconnecting traditional computers and a large number of smart objects or networks such as Wireless Sensor Networks (WSNs). This Internet of Things (IoT) will be the foundation of many services, and our daily lives will depend on its availability and reliable operations. Therefore, among many other issues, the challenge of implementing secure communication in IoT must be addressed.

Figure 1-1 Smart things having communication capabilities [63]

The traditional internet has established and tested ways of securing networks. IoT is a hybrid network of the internet and resource-constrained networks, and it is therefore reasonable to explore the options of using security mechanisms standardized for the internet in IoT. The common vision of smart systems today is by and large associated with one single concept, the Internet of Things (IoT), where the whole physical infrastructure is linked with intelligent monitoring and communication technologies through the use of wireless sensors. In such an intelligent, vibrant system, sensors are connected to send useful information and control instructions via distributed sensor networks. Wireless sensors have an easy deployment and better flexibility of devices, contrary to wired setup. With the rapid technological development of sensors, wireless sensor networks (WSNs) will become the key technology for IoT and an invaluable resource for realizing the vision of the Internet of Things (IoT) paradigm. Today's transition from legacy WSN systems to the Internet of Things (IoT) can be in a first approach summarized as an extension of the internet boundaries up to the leaf devices. Instead of stopping at the sink node, as was the case in WSNs, internet protocols can now run between any two IoT nodes. Accordingly, the architectures and communication types in IoT are becoming closer to those of legacy internet. Decentralization is appearing within once-monolithic, sink-centric subsystems whose end nodes are now able to be involved in peer-to-peer, bidirectional communications with any remote internet peer. The cost of sensors, processors, and transmitters is becoming less and their computational and processing powers becoming higher, allowing to put them into any object of our daily life (i.e., food, clothes, medicine, and so on). The technological advances also enhance this connectivity by adding one more dimension to it—connecting anything. The adoption rate of IoT is trending to be at least five times faster than the adoption of electricity and telephony.

1.1.1 Internet of Things Definition

In 1999, the term Internet of Things was devised by Kevin Ashton, co-founder and executive director of Auto-ID Center at MIT, and refers to uniquely identifiable objects and their virtual representations in an "internet-like" structure. IoT is defined as *"A pervasive and ubiquitous network which enables monitoring and control of the physical environment by collecting, processing, and analyzing the data generated by sensors or smart objects."*

IoT not only has the same security issues as sensor networks, mobile communications networks and the internet, but also has its specialties such as privacy issues, different authentication and access control network configuration issues, information storage and management, and so on. Data and privacy protection is one of the application challenges of IoT [1]. In IoT, RFID systems, WSNs sensors perceive the end of information technology, which protects the integrity and confidentiality of information by password encryption technology. There are many ways to encrypt data and information, such as random hash lock protocol (hash function), hash chain protocol, extract key from an infinite channel, encrypted identifier, and so on. Risk of IoT security from itself and others comes from the related technology of construction and implementation of the network functions. IoT itself is the integration of multiple heterogeneous networks; it should deal with compatibility issues between different networks that are prone to security issues; for example, it is difficult to establish the junction of the relationship as the relationship of trust between nodes that are constantly changing, but this can be solved by key management and routing protocols.

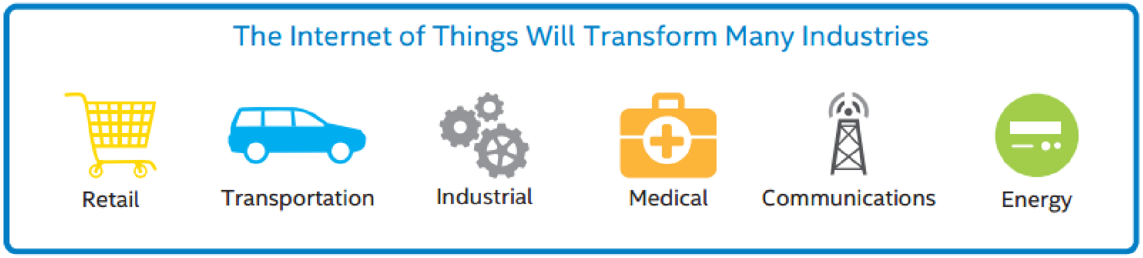


Figure 1-2 Internet of Things Applications [65]

Security issues such as DOS/DDOS attacks, forgery/middle attack, heterogeneous network attacks, application risk of ipv6, and WLAN application conflicts also affect the transport security of IoT. In the core network, due to the large amount of data during the transmission, it is easy to cause network congestion. We should give full consideration to the capacity and connectivity issues, such as address space, reference network redundancy, and security standards. The application of IoT directly connects with people's everyday lives to ensure technology security and to strengthen human security awareness and norms of human behavior at the same time. Meanwhile, people associated CPS (cyber-physical systems) and pervasive computing security have also been researched. Just to give an example how IoT would affect our daily lives: You enter the supermarket and receive your fridge's text message, "You are out of milk." In the dairy section, sensors signal your grocery cart that you’ve taken a milk carton. As you walk towards the pharmacy, your fitness wristband vibrates as it takes your vitals and streams the results to your doctor to adjust your prescription. When you’re finished shopping, you simply walk out the door. Your credit card is charged when you exit the supermarket’s geofence. As you drive home, your car communicates with other cars on the roadway to prevent accidents.

Deployments of numerous devices with limited processing and memory capabilities can increase the threat space of IoT applications. IoT inherits the drawbacks of the current internet on an infinitely larger but more invisible scale. Every single connection could make networks vulnerable. An attacker can exploit a weakness in IoT device with limited capability to penetrate connected IoT application, which is supposedly considered more secure. The early years of the Internet of Things (IoT) started with Machine to Machine (M2M) Communication. M2M communication indicates two machines communicating with each other, usually without human involvement. The communication platform is not defined and can be both wireless and wired communication. The term M2M stems from telephony systems. In these systems, different endpoints needed to exchange information between each other, such as the identity of the caller. This information was sent between the endpoints without a human being needed to initiate the transmission. The M2M term is still very much in use, especially in the industrial market, and is commonly regarded as a subset of IoT [2].

1.1.2 Internet of Things Layers

In order to analyze the security issues of IoT in more detail, IoT layers are divided into four layers named the perception layer (Sensing and Identification layer), Network Construction layer, Information Processing layer, and Integrated Application layer. Perception layer can further be divided into perception nodes and perception network.

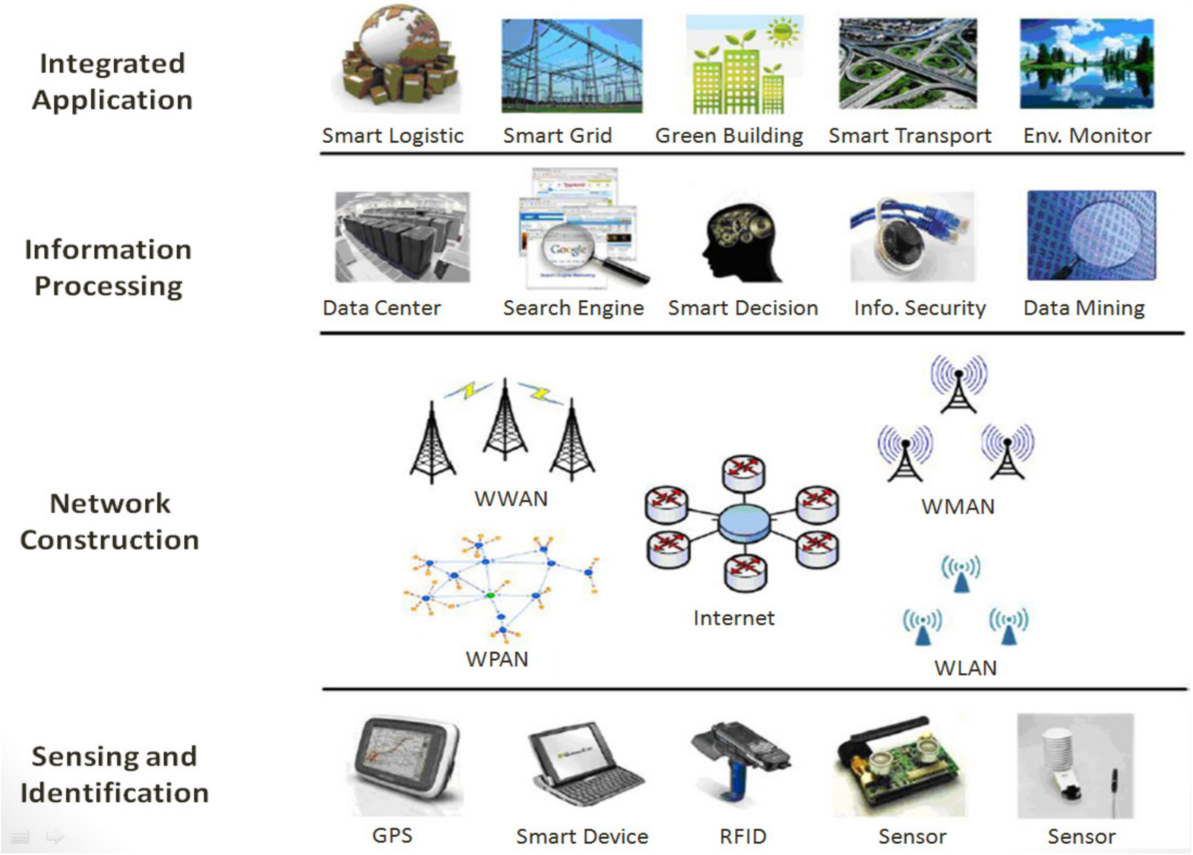


Figure 1-3 Internet of Things Layers [64]

Each layer has a corresponding technical support; these technologies at all levels play irreplaceable roles, but these techniques are more or less related to the existence of the range of problems that can cause insecurity, privacy, and other security issues of data.

IoT must ensure the security of all layers. In addition, IoT security should also include the security of the whole system crossing the perception layer, transportation layer, and application layer.

- At the bottom there is the Sensing and Identification layer that contains all the sensors and devices like WSN RFID, etc.

- The Network Construction layer includes access network security, core network security, and local network security. There are 3G access network security, Ad-Hoc network security, Wi-Fi security, and so on for these sub layers. Different network transmission has different technology.
- The Information processing layers include the data processing and get useful information out of data. They are mainly for information collection, object perception, and object control. The perception network is responsible for communicating with the transportation network.
- The application layer includes the application support layer and specific IoT applications. The security in the support layer includes middleware technology security, cloud computing platform security, and so on. IoT applications in different industries have different requirements.

The Network Construction layer is also called the perception layer and is used for data acquisition and data control; the perception network sends collected data to the gateway or sends control instruction to the controller. Perception layer technologies include RFID, WSNs, RSN, GPS, etc.

1.2 Design Requirements for IoT-Oriented Mechanisms

Deployments of numerous devices with limited processing and memory capabilities can increase the threat space of IoT applications. IoT inherits the drawbacks of current internet on an infinitely larger but more invisible scale. Every single connection could make networks vulnerable. An attacker can exploit a weakness in an IoT device with limited capability to penetrate a connected IoT application, which is supposedly considered more secure.

Markets won't invest in the right level of security, as today "time to market" is a bigger driver than the level of security or privacy. IoT devices communicate among themselves with little human interactions; mutual authentication is a crucial aspect of the paradigm. Technological standards are still fragmented; a common set of standards between companies, educational systems, and nations are required.

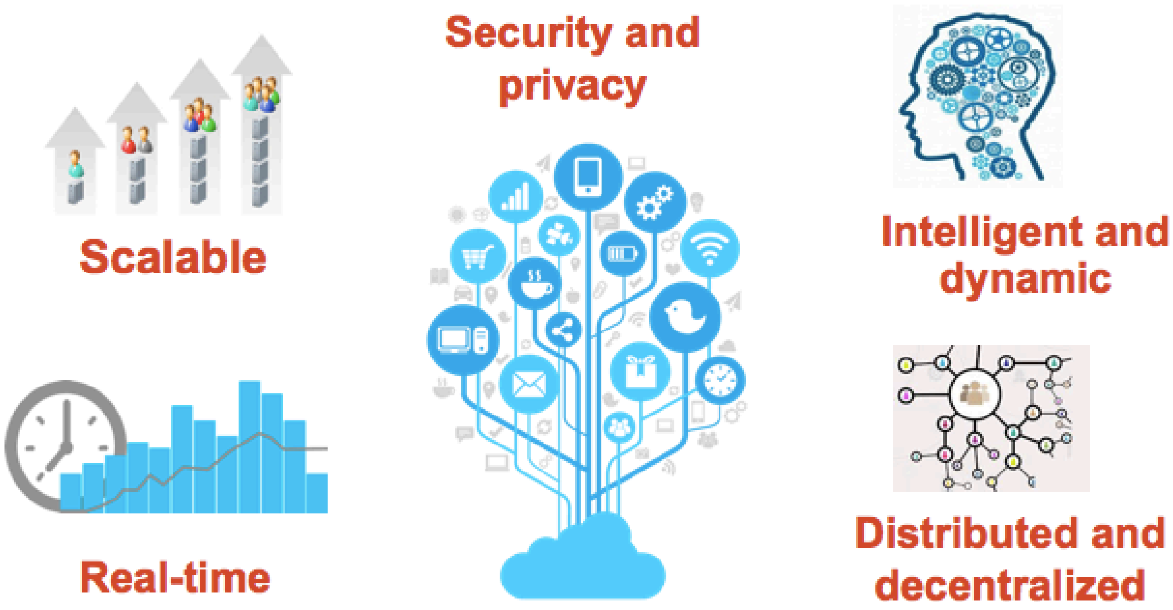


Figure 1-4 Internet of Things key requirements [66]

Some of the key design requirements of IoT oriented mechanisms are as follows with reference to resource-constrained sensors integrated into this future enabling technology of IoT:

1.2.1 Security and Privacy Problems

Although internet security architecture has been very mature, there are still many means of attack. For example, a large number of malicious nodes sending data at the same time might lead to DoS attack. There is a trade-off between security and cost; simple cryptographic functions such as logical addition, XOR, and various hash functions are easy to implement (require fewer resources) but easy to break in terms of their security. Also, devices may be distributed in public areas unprotected, thus are vulnerable to physical attacks, and preventing sensor tempering and system misuse is another important challenge.

Lightweight encryption algorithm and security authentication protocols are required for resource-constrained environments integrated into IoT mechanisms. Middleware security is required for heterogeneous components to form a cohesive whole in IoT mechanisms. Some of the other requirements in security are authorized access of data, authentication, and device and identity management. Sensor networks generally exist without IoT, but IoT cannot exist with sensor networks.

1.2.2 Privacy and Policy

Privacy policies for IoT applications are different than traditional privacy. It's easy for any person to get involved in IoT even without his knowledge, as ubiquitous devices monitor everything, causing privacy concerns; data from them can be stored indefinitely. Legislative and ethics issues need to be considered. Moreover, the definition of privacy by regulatory bodies can be quite different among different geo-political zones, making it more difficult to have the same policies everywhere; the requirements of enabling the reuse of IoT data gathered by one IoT application towards other applications are mainly contradicting with privacy-by-design.

1.2.3 Scalability

The vast amount of interconnected things in IoT demands highly scalable protocols. This also has an influence on security mechanisms as well. For instance, centralized approaches, e.g., hierarchical Public Key Infrastructures (PKIs), as well as some distributed approaches, for instance, pairwise symmetric key exchange schemes, cannot scale with IoT.

Along with that, there are addressing issues for unique and extensible identifiers for billions of devices, so transition to IPv6 is being made. Moreover, with these huge number of devices connected to the internet, data is everywhere; acting upon that data is dangerous since you don't know its source. Factors needed to be considered are

- Understanding Big Data
- Accuracy of data
- Data needs to be contextual (Temporal, spatial, or thematic)

1.2.4 Distributed and Decentralized

With the exponential growth of devices being connected to the internet, it's difficult to manage a large number of devices. Existing internet infrastructure was originally designed to connect computers, phones, printers, servers, and, more recently, mobile devices mainly operated by users. However, now we are facing huge growth not only in connecting devices but also in the type of devices, so it should be a distributed and decentralized approach. Another requirement is to be able to integrate already deployed things, WSNs, and devices into IoT. In today's systems, every LEGO brick comes from a different source, and they all must snap together. Factors needed to be considered are

- Technology Neutrality
- IPv4 and IPv6 must effectively cohabitate
- Data flow securely and efficiently
- Demand for API access and interoperability

1.2.5 Intelligent and Dynamic

IoT being an integrated part of Future Internet is defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols. IoT mechanisms are required to be intelligent and dynamic to support the visions of pervasive or ubiquitous computing concepts. These mechanisms must be self-sustainable and viable with self-configuring capabilities as well as having the openness for future extensions, ideas, and technologies. Also, the understanding of new network traffic patterns by the IoT mechanism are an important basis for design of network infrastructure and protocols.

1.2.6 Real Time

IoT mechanisms are required to work and are available in real time, anywhere and anytime. The concepts "anywhere" and "anytime" need not necessarily refer, respectively, to "globally" and "always." The "anywhere" mainly refers to the concept of where it is needed, and the "anytime" similarly refers to when it is needed. Due to the involvement of sensors/actuators in the IoT system, they are required to perform the sensing/actuation in real time with stable network connectivity and constant presence; hence, IoT mechanisms must support the mobility of the devices or the users in IoT applications as well even for the Limited-function Embedded Devices.

For sensor networks in IoT, the diversity of resource and the network heterogeneity make security research much more difficult. Some of the security measures for IoT might be data encryption, key management schemes, secure efficient routing protocols, better intrusion detection algorithms, better physical security design, etc.

1.3 Security and Privacy of Internet of Things as a whole

The ability for any two nodes of exchanging information with one another is not sufficient for a networked architecture being deployed in proximity of the physical world (either sensed or acted upon) and therefore vulnerable to malicious attacks on nodes and/or communications channels. Security is an essential service that has to be provided. There are different kinds of IoT applications such as Intelligent Transportation, Smart Home, Intelligent Urban Management, Intelligent Medical, Smart Green, and Smart Grid. For different applications, we have different security requirements. For example, the security of data privacy would be of great importance for Intelligent Transportation and Intelligent Medical. But to Intelligent Urban Management and Smart Green, data authenticity would be more important. In order to get the best security, we may need to give them different weight from different applications.

As we know, some security needs cannot be fulfilled by only using one specific technology in a single layer. For example, for a system with a weak application layer, no matter how much effort we have made for data privacy security in the perception layer, it would be easy for a cracker to get all private data. So in this situation, we need to have some cooperation between different layers. Thus, we need to design corresponding technologies for cross-layer usage.

We not only should deal with single-layer heterogeneous issues, but also need to deal with cross-layer heterogeneous integration issues. We need to find out new technologies to build system autonomy and a heterogeneous integration model to meet the cross-layer requirement so that we can get uniform data across different layers. Common to WSN and RFIDs are their severely limited power resources, which classify them as ultra-low power devices. RFID Technology uses low-cost transponders that contain item specific information to avert removal reapplication attacks. If Transponders hold unique and cryptographically secured data that uniquely binds a given instance of product to a given tag, this makes duplication or re-application of tags difficult.

Conventional security primitives cannot be applied due to the heterogeneous nature of sensors (either implanted, on-body, or wearable), low resources, and the system architecture of IoT based healthcare systems. Physiological data measurements are collected and transmitted to remote servers to analyze the medical data and to intervene in case of an emergency. Any unauthorized use of a patient's data or privacy concerns may restrict people to utilize IoT-based healthcare applications. To mitigate these security and privacy threats, strong network security infrastructures are required. Peer authentication and End-to-End data protection are crucial requirements to prevent eavesdropping on sensitive data or malicious triggering of harmful actuating tasks [3]. Symmetric key cryptography such as AES provides fast and lightweight encryption and decryption on smart devices, and their integrated hardware supports it as well. However, when the number of devices connected becomes high, exchanging symmetric keys becomes a challenging task; therefore, an efficient scalable key establishment protocol is required. Asymmetric key cryptography is another method for key establishment at two ends, but it involves high computational overheads, which are the main concerns for resource constrained devices. Although internet security architecture has been very mature, there are still many means of attack. For example, a large number of malicious nodes sending data at the same time might lead to a DoS attack. There is a trade-off between security and cost; simple cryptographic functions, such as logical addition, XOR, and various hash functions, are easy to implement (require fewer resources) but easy to break in terms of their security. Also, devices may be distributed in public areas unprotected; thus, they are vulnerable to physical attacks, and preventing sensor tempering and system misuse is another important challenge.

Lightweight encryption algorithm and security authentication protocols are required for resource-constrained environments integrated into IoT mechanisms. Middleware security is required for heterogeneous components to form a cohesive whole in IoT mechanisms. Some of other requirements in security are authorized access of data, authentication, and device and identity management. Sensor networks generally exist without IoT, but IoT cannot exist with sensor networks. Security in IoT context involves End-to-End communications; an IoT node can be expected to act alternatively as a client and as a server, contrary to a wireless sensor network. This implies that providing security means only the two participants involved at the ends in the pairwise key exchange protocol should have access to the agreed generated key. By having mutual authentication, these two peers should also authenticate each other and link the generated keys to their respective identities [4]. Sensors installed or implanted on the body are supposed to have low (energy, computation) resources and are not considered to perform complex asymmetric cryptographic operations.

Privacy policies for IoT applications are different than traditional privacy. It’s easy for any person to get involved in IoT even without his knowledge, as ubiquitous devices monitor everything, causing privacy concerns, and data from them can be stored indefinitely. Legislative and ethics issues need to be considered. Moreover, the definition of privacy by regulatory bodies can be quite different among different geo-political zones, making it more difficult to have the same policies everywhere, and the requirements of enabling the reuse of IoT data gathered by one IoT application towards other applications is mainly contradicting with privacy-by-design.

1.4 Network Security

In the context of network security, we should consider existing security goals and be aware of traditional security treats in networks. Cryptographic primitives are in general utilized to comply with the main security goals for exchanged messages and the system itself. These security goals are: (i) confidentiality: the message is only disclosed to authorized entities, (ii) integrity: the original message is not tampered with, (iii) authenticity: the message is sent from a genuine entity, and (iv) availability: the system keeps serving its purpose and stays uninterruptedly available for legitimate entities.

Attack techniques are important to understand the rationale of security mechanisms in communication protocols. With respect to the secure Internet of Things (IoT), the following attacks are important:

1.4.1 Eavesdropping

Eavesdropping is the process of overhearing an ongoing communication, which is preliminary for launching the next two attacks. Eavesdropping on unprotected wireless communication is launched with less effort, since everyone in general has access to the medium (air), whereas in wired communication, physical access to the medium (cable) is restricted. Confidentially is a typical counter-measurement against eavesdroppers. However, if the keying material is not exchanged in a secure manner, the eavesdropper could compromise the confidentiality. Therefore, secure key exchange algorithms, such as Diffe-Hellman (DH), are used in the practice.

1.4.2 Impersonation

Impersonation is when a malicious entity pretends to be another mostly legitimate entity, for instance by replaying a genuine message in order to bypass the aforementioned security goals. A special form of this attack is the Man-In-The- Middle (MITM) attack.

1.4.3 Man-In-The- Middle (MITM) Attack

The MITM attack takes place when a malicious entity is on the network path of two genuine entities. Hence, it is capable of delaying, modifying, or dropping messages. MITM attack is interesting within the context of PKC. Then the malicious entity does not attempt to break the keys of involved parties but rather to become the falsely trusted man-in-the-middle. The malicious user achieves this by replacing the exchanged keys with its own. This way, each of the parties establishes a secure channel with the malicious user, who gains access to messages in plaintext.

1.4.4 Denial of Service (DoS) Attack

The DoS attack targets the availability of a system that offers services. This is achieved by exhaustingly consuming resources from the victim so that the offered services become unavailable to legitimate entities. A common way to launch this attack is to trigger expensive operations from the victim that consume resources, such as computational power, memory, bandwidth, or energy. This attack is critical for constrained devices, where existing resources are already scarce.

1.4.5 Physical Attacks

In this attack, an adversary takes advantages of the wireless nature of devices in IoT in order to disable sensors/devices temporarily or permanently. To permanently disable a device, he may remove the device from the place or from the network and replace it with a device with a low price. The other way is sending a kill command to erase the memory, removing the antenna, or giving a high energy wave to the sensor that will also destroy the sensor permanently. To disable the sensor temporarily, the attacker can use a Faraday cage like an aluminum foil-lined bag in order to block electromagnetic waves from it. In other cases, he may prevent sensors to communicate with the server or remote hosts by generating a signal in the same range as the server broadcasts, call active jamming.

1.5 Security issues comparison between IoT and traditional network

IoT and traditional network security issues are different in many ways. IoT is composed of RFID nodes and WSN nodes, whose resources are limited, while the internet is composed of PC, severs, and smart phones, whose resources are rich. In the internet, we use combinations of complex algorithms and lightweight algorithms to maximize security with less considerations of resource usage such as computation power. In IoT, in most cases, we can only use lightweight algorithms to find the balance between security and power consumptions. Connection between IoT nodes are always through slower, less secure wireless media, which results in easy data leakage, easily node compromising, and other insecure issues. Whereas with the internet, most communications are through faster, more secure wired or wireless communications. Even with the mobile internet, wireless connections are built on top of complex secure protocols that are almost impossible to implement for resource limited IoT nodes.

Although there are various devices in the internet, but with the abstraction of operating system, their data formats are almost the same with Windows family and Unix-like operating systems. However, in IoT, what we have is just a bare wireless node. There is no operating system, just a simple embedded program for the chip. With the diversity of nodes perception goal, there comes different chip hardware that results in heterogeneous data contents and data formats. There are all kinds of IoT applications in the application layer used in our everyday lives; they gather our private information every second automatically to make our lives easier. These applications can even control our everyday environment. It would be a great potential security problem if we lose control of the IoT system. While with the internet, if we do not provide our information ourselves, there is no way for attackers to get our information. And with the help of operating systems and plenty of security software, the environment is more secure. Therefore, the IoT system lives in a more dangerous environment with limited resources and less network guards, so we need to implement lightweight solutions to deal with this more dangerous environment.

1.6 Motivation

The Internet of Things is intended to connect smart objects surrounding us in our daily lives for smarter living conditions and improved quality of service in different application domains, such as smart transport systems, smart cities, and smart healthcare, through ambient communications and remote control facilities. It is believed that up to fifty billion machines and objects will be connected by 2020 [3]. The concept behind IoT is the pervasive presence of various wireless technologies around us such as RFID tags, sensors, actuators, or mobile phones, in which computing and communication systems are seamlessly embedded. Every single connection could make networks vulnerable. These projections depict the possibility of a smarter, efficient and safer world of interconnected devices [5], while some observers show concerns that the IoT represents a darker world of surveillance, privacy and security violations, and consumer lock–in. As all these things or devices are heterogeneous in their available computational, energy, bandwidth, and memory resources, there is a key requirement to have such security primitives to provide security and privacy of the whole network and user-data while taking into account the limited resources as well. Highly constrained sensors cannot provide enough resources required for heavy asymmetric cryptographic functions. Some of them might be placed on public places, making them vulnerable to physical attacks, too.

Secure End-to-End communication and end-users' privacy protection are major concerns in the development of IoT. Traditional security primitives cannot be applied to resource-constrained wireless sensors, so new security techniques and algorithm protocols are required to make things secure enough and efficiently. Secret key distribution for heterogeneous sensors becomes challenging due to the inconsistencies in their cryptographic primitives and computational resources in varying IoT applications. New security paradigms are needed for End-to-End secure key establishment protocols that are lightweight for resource-constrained sensors and secure through strong encryption and authentication so that huge number of resource-constrained sensors can also benefit from the same security functionalities that are typical of unconstrained domains without however having to execute computationally intensive operations.

1.7 Main Contributions

The summary of the key contributions of this thesis are as follows:

- Study the heterogeneous wireless sensors integration into the Internet of Things (IoT), new security challenges raised, and the security primitives that can be taken to protect their data over the internet in context of IoT. The question as to how sensor nodes should provide their services when connecting WSN to the internet is important, either directly or through the base station. The "thing" connected to the internet is required to be locatable and addressable via the internet, but this particular configuration might not be suitable for certain scenarios. There are different approaches in which WSNs are integrated into IoT depending on the application requirements and already deployed WSNs infrastructure. Some of these approaches are completely independent from the internet where WSN has its own protocol and sensor nodes communicate through a centralized device called the base station; the other is one in which sensor nodes implement TCP/IP stack (or a compatible set of protocols such as 6LoWPAN) so that any internet host can have direct communication with them and vice versa. Sensor nodes are no longer to use specific WSN protocols.

- Investigate the feasibility of implementing Localized Encryption and Authentication Protocol (LEAP+), a distributed symmetric based key management over the ZigBee stack. First, a qualitative security analysis was conducted of LEAP+, and the current ZigBee key management scheme then implemented LEAP+ on the ZigBee platform using the QualNet 5.0.2 simulator. Experimental results show that a distributed key management scheme such as LEAP+ provides improved security and offers good scalability as well. ZigBee specification is an emerging wireless technology designed to address the specific needs of low-cost, low-power WSNs with an area of concern with its centralized approach introducing the issues of limited scalability and a single point of vulnerability. Also, it uses a public key infrastructure. Due to these limitations, replacing ZigBeee key management with LEAP+ that is decentralized, scalable, and designed to support multiple types of keys based on the message type being exchanged is suggested.

- Design of a secure End-to-End cooperative key establishment system answering the constraints and characteristics of heterogeneous Machine-to-Machine or the Internet of Things environments. Wireless sensors considered might perform as node or server depending on the specific application scenario. The main idea behind this proposed security protocol is based on the cooperation by offloading heavy cryptographic operations of resource-constrained sensors to the neighboring trusted nodes or devices. The proposed security paradigm aims to establish shared secret keys in a secure and efficient way to provide confidentiality and authentication while exchanging data. Moreover, the proposed protocols are able to identify the neighboring nodes that don't cooperate, failing to deliver its assigned shared part during the key establishment process. Security analysis and performance evaluation results show that the proposed protocols are secure and sufficiently energy efficient, especially for resource-constrained sensors.
- Design and development of secure End-to-End Authentication and Key Agreement Protocol for a Body Area Network scenario, where the assisting nodes are verified by correlating their accompanying accelerometers data with the base station accelerometer data to detect if these assisting nodes are installed on the body. Security of the authentication and key agreement protocol by offloading heavy cryptographic primitives in such a cooperative way is checked for formal security threats such as DOS, resilience, MITM, and Sybil attacks. Moreover, our proposed protocols are able to identify the compromised participating devices so a mechanism can be implemented to revoke their pair-wise symmetric keys and to initiate the process of re-keying for them.

1.8 Thesis Organization

In this chapter, the introduction addresses the Internet of Things (IoT), security and privacy for devices in IoT as a whole, and how these are different as compared to traditional networks. It also briefly describes the secure communication requirements and challenges for heterogeneous wireless sensors and their limitations in performing secure communications. Additionally, a number of attacks threatening the Internet of Things and the sensors or devices connected are presented. Motivation of the research work carried out to design and develop key establishment protocols for End-to-End secure communication is described as well. The solution approaches provide secure enough communication while conserving the energy of resource-constrained wireless sensors. After this introduction on the Internet of Things and its security requirements in first chapter, the reminder of this dissertation is organized as follows:

CHAPTER 2 presents the related work, how wireless sensors are integrated into the Internet of Things, and the concept of cryptography designated for resource constraint wireless sensors designs in the context of IoT. At the end, the existing security solution approaches available are described to compare with our work.

In CHAPTER 3, we have considered these wireless sensors in the form of Wireless Sensor Network (WSN), their security, and their privacy while taking into consideration the scalability of the network. For this, LEAP+ key management scheme that is distributed in nature and also has key revocation and re-keying mechanism is implemented to ZigBee stack, replacing its centralized key management scheme. Key management scheme routines are implemented in C++, and the simulations are run through Qualnet simulator by a PERL file.

CHAPTER 4 gives details of key establishment protocol where these wireless sensors might have to perform as node or server depending on the specific application scenario. The main idea behind these new proposed security protocols is based on the cooperation by offloading heavy cryptographic operations of resource-constrained sensors to the neighboring trusted nodes or devices. The proposed security paradigm aims to establish shared secret keys in a secure and efficient way to provide confidentiality and authentication while exchanging data. We have taken the healthcare monitoring scenario to implement our proposed protocols where medical sensors are implanted and worn on the body of patients of different capabilities and resources. Among all this heterogeneous equipment operating in the vicinity, the implanted sensors on human body are highly resource-constrained nodes and might be located in inaccessible places inside the body (i.e., replacing batteries is impossible, needs surgery). Therefore, preserving their energy resources becomes critically important with the requirement to have an End-to-End secure communication to protect their data. Moreover, the proposed protocols are able to identify the neighboring nodes that don't cooperate, failing to deliver its assigned shared part during the key establishment process. Security analysis and performance evaluation results show that the proposed protocols are secure and sufficiently energy efficient, especially for resource-constrained sensors.

CHAPTER 5 describes the secure End-to-End Authentication and Key Agreement Protocol for Body Area Network scenario, where the assisting nodes are verified by correlating their accompanying accelerometers data with the base station accelerometer data to detect if these assisting nodes are installed on the body. Security of the authentication and key agreement protocol by offloading heavy cryptographic primitives in such cooperative ways is checked for formal security threats such as DOS, resilience, MITM, and Sybil attacks. Moreover, our proposed protocols are able to identify the compromised participating devices so a mechanism can be implemented to revoke their pair-wise symmetric keys and to initiate the process of re-keying for them.

CHPATER 6 summarizes the work accomplished in this dissertation by providing concluding remarks and presenting further research perspectives.

CHAPTER 2: Background and Related Work

With the rapid technological development of sensors, wireless sensor networks (WSNs) will become the key technology for IoT and an invaluable resource for realizing the vision of the Internet of things (IoT) paradigm. Pervasive and ubiquitous computing has a long-lasting tradition of looking into the integration of physical objects with the digital world. Recent developments in the field of embedded devices have led to smart things increasingly populating our daily lives, slowly but steadily forming interconnected networks of physical objects. Sensor nodes are networked together to create environmental monitoring applications, making cities smarter and dynamically adapting to their context. Home appliances such as TVs, alarm clocks, digital picture frames, and Hi-Fi systems can communicate with each other to offer integrated services such as cross-devices multimedia experiences, smarter HVAC (Heating, Ventilating, and Air Conditioning) systems, or more energy aware and efficient homes. RFID-tagged objects in stores and along supply chains allow manufacturers, suppliers, and service providers to optimize their operations. Wireless sensors have an easy deployment and better flexibility of devices, contrary to wired setup. With the rapid technological development of sensors, wireless sensor networks (WSNs) will become the key technology for IoT and an invaluable resource for realizing the vision of the Internet of Things (IoT) paradigm. It is also important to consider whether the sensors of a WSN should be completely integrated into IoT or not. New security challenges arise when heterogeneous sensors are integrated into the IoT. Security needs to be considered from a global perspective, not just on a local scale. Traditional security primitives cannot be directly applied to IoT technologies due to the different standards and communication stacks involved. Along with scalability and heterogeneity issues, some major parts of IoT infrastructure consist of resource constrained devices such as RFIDs and wireless sensor nodes. Therefore, a flexible infrastructure is required that is capable of dealing with security and privacy issues in such a dynamic environment. This paper presents an overview of sensor integration into IoT, major challenges in IoT, existing security solutions that can be taken to protect their data over the internet, and identifying some open issues for future research.

In this chapter, the integration approaches of wireless sensors into the Internet of Things are described, and characteristics of a lightweight cryptosystem will be defined. Then, a selection of well-known security solution approaches and a literature survey will be presented as a related work. This related work covers recent secure proposed protocols and the security solution approaches in general for resource-constrained wireless sensors in the context of the Internet of Things (IoT).

2.1 Sensor Networks in a Globally Connected Network

The things to be connected to the internet largely vary in terms of characteristics. This ranges from very small and static devices (e.g., RFIDs) to large and mobile devices (e.g., vehicles). Such heterogeneity induces complexity and stipulates the presence of an advanced middleware that can mask this heterogeneity and promote transparency. Among other technologies, radio frequency identification (RFID) and wireless sensor network (WSN) represent two of the most promising technologies enabling the implementation of IoT infrastructure. RFID is a low-cost, low-power technology consisting of passive or battery-assisted passive devices (tags) that are able to transmit data when powered by the electromagnetic field generated by an interrogator (reader). Since passive RFID tags do not need a source of energy to operate, their lifetimes can be measured in decades, thus making the RFID technology well-suited in a variety of application scenarios, including the industrial and healthcare ones [6]. The main challenges for RFID are non-uniform encoding, conflict collision, and RFID privacy protection.

On the other hand, WSNs are basically self-organizing ad hoc networks of small, cost-effective devices (motes) that communicate/cooperate in a multi-hop fashion to provide monitor and control functionalities in critical applications, including industrial, military, home, automotive, and healthcare scenarios [7]. Currently, most WSN motes are battery-powered computing platforms integrating analogue/digital sensors and an IEEE 802.15.4 radio enabling up to 100m outdoor communication range (single hop). Unlike other networks, WSNs have the particular characteristic of collecting sensed data (temperature, motion, pressure, fire detection, voltage/current, etc.) and forwarding it to the base station or gateway. Even though most WSN protocols were not designed for two-way communications, as illustrated in IMS research, they should also be able to receive information and send it to the sensors (a command, for example) and react on behalf of the commander/user, e.g., automating home appliances.

Integration of Wireless Sensor Networks (WSN) into IoT is not mere speculation; a number of big technology companies are supporting and developing their IoT infrastructure around WSN. Noteworthy examples are IBM's "A Smarter Planet," a strategy that considers sensors as fundamental pillars in intelligent water management systems and intelligent cities and the CeNSE project by HP Labs, focused on the deployment of a worldwide sensor network in order to create a "central nervous system for the Earth" [8]. The question of how sensor nodes should provide their services when connecting WSN to the internet is important, either directly or through the base station. The "thing" connected to the internet is required to be locatable and addressable via the internet, but this particular configuration might not be suitable for certain scenarios. Some specific scenarios, for instance, in SCADA systems, a sensor node does not need to provide its services directly, and other scenarios are where a sensor node should be completely integrated into the internet.

There are different approaches in which WSNs are integrated into IoT depending upon the application requirements and already deployed WSNs infrastructure. Some of these approaches are described below.

- Completely independent from the internet where WSN has its own protocol and sensor nodes communicate through a centralized device called the base station. Any query coming from the internet host is traversed through the base station that collects and holds all data from the sensor nodes. If the base station acts as an applications layer gateway, then the internet hosts and sensor nodes are able to address each other and exchange information without establishing a true direct connection. WSN is still independent of the internet, and all queries go through the gateway.
- Sensor nodes implement TCP/IP stack (or a compatible set of protocols such as 6LoWPAN) so that any internet host can have direct communication with them and vice versa. Sensor nodes are no longer to use specific WSN protocols.
- There is another topology-based integration approach in which the level of integration depends on the actual location of the nodes; nodes can be dual sensors (base stations) located on the root of the WSN or full-fledged backbone of devices that allow sensing nodes to access the internet in one-hop (access point). WSN becomes an unbalanced tree with multiple roots; leaves are normal sensors nodes, and other elements are internet-enabled nodes.

The evolution of IoT has its origin in the convergence of wireless technologies, advancements of MEMS, and digital electronics, where, as a result, miniature devices with the ability to sense and compute are communicating wirelessly. However, having IP connectivity does not mean that every sensor node should be directly connected to the internet. There are many challenges that must be carefully considered, and one of those challenges is security [8]. In the era of IoT, the interaction or relationship between humans and machines needs to be considered more seriously as machines are getting smarter and starting to handle more human tasks. A thing might be a patient with a medical implant to facilitate real-time monitoring in a healthcare application or an accelerometer for movement attached to the cow in a farm environment. In such situations, humans are required to trust the machines and feel safe about them [9] [7].

2.2 Major Challenges in the Internet of Things

Deployments of numerous devices with limited processing and memory capabilities can increase the threat space of IoT applications. IoT inherits the drawbacks of the current internet on an infinitely larger but more invisible scale. Every single connection could make networks vulnerable. An attacker can exploit a weakness in IoT device with limited capability to penetrate connected IoT application, which is supposedly considered more secure. Recent attacks of smart light bulb passwords, hacks of Foscam baby monitors, and Belkin home automation systems are just the beginning. Markets won't invest in the right level of security, as today "time to market" is a bigger driver than the level of security or privacy.

IoT devices communicate among themselves with little human interactions; mutual authentication is a crucial aspect of the paradigm. Some of the challenges this future enabling technology of IoT has:

2.2.1 Scalability

The vast amount of interconnected things in IoT demands highly scalable protocols. This also has an influence on security mechanisms. For instance, centralized approaches, e.g., hierarchical Public Key Infrastructures (PKIs), as well as some distributed approaches, for instance, pair wise symmetric key exchange schemes, cannot scale with the IoT. Along with that, there are addressing issues for unique and extensible identifiers for billions of devices so transition to IPv6 is being made. Moreover, with these huge number of devices connected to the internet, data is everywhere; acting upon that data is dangerous since the users don't know its source. Factors needed to be considered are:

- Understanding Big Data
- Accuracy of data
- Data needs to be contextual (Temporal, spatial, or thematic)

2.2.2 Lack of Standardization in the IoT Market

Technological standards are still fragmented; a common set of standards between companies, educational systems, and nations are required.

2.2.3 Highly Distributed Nature

With the exponential growth of devices being connected to the internet, it is difficult to manage a large number of distributed devices. Also, sensors and devices may be distributed in public areas unprotected and thus, are vulnerable to physical attacks; preventing sensor tempering and system misuse is another important challenge.

2.2.4 New Network Traffic Patterns to Handle

Characteristics of smart objects traffic in IoT are still unknown, making them more vulnerable to internet attacks. The internet will be traversed by a large number of data generated by sensors deployed for heterogeneous purposes. These new network traffic patterns are important bases for design of network infrastructure and protocols in the context of IoT.

2.2.5 Lack of Integration of Already Deployed Things, Devices, Sensor Networks

In today's systems, every LEGO brick comes from a different source, and they all must snap together; factors needed to be considered are

- Technology Neutrality
- IPv4 and IPv6 must effectively cohabitate
- Data flow securely and efficiently
- Demand for API access and interoperability

2.2.6 Limited-function Embedded Devices

Not all the devices which are being connected to the internet have enough resources due to the heterogeneity of the sensors or devices in IoT; they might have constraints on their resources available such as power, computation capability, storage, etc. Most of the communications are wireless, which makes attacks (e.g. eavesdropping, jamming) simple. Some types of devices (e.g. passive RFID tags) are unable to provide authentication or data integrity. Low-power CPU requires more time for expensive cryptographic operations than unconstrained devices, affecting the responsiveness of the device and the energy consumption/life-time of the node.

2.2.7 Support for Mobility

Stable network connectivity and constant presence cannot be expected in such an environment; hence, IoT must support the mobility of the devices or the users in IoT applications.

2.2.8 Privacy and Policy

Privacy policies for the Internet of Things applications are different than traditional privacy. It's easy for any person to get involved in IoT even without his knowledge, as ubiquitous devices monitor everything, causing privacy concerns; data from them can be stored indefinitely. Two kind of challenges are there to be considered: legislative issues and ethics issues. Moreover, the definition of privacy by regulatory bodies can be quite different among different geo-political zones, making it more difficult to have the same policies everywhere, and the requirements of enabling the reuse of IoT data gathered by one IoT application towards other applications is mainly contradicting with privacy-by-design.

2.2.9 Lightweight Security Protocols for Constrained Environments

Processor performance of sensor nodes is lower in IoT; lightweight encryption algorithm and security authentications are required.

2.2.10 Trust and Ownership Issues

A priori trusted relationships are unlikely for the large amount of devices interacting with each other and users. Thus, automated mechanisms to measure and manage trust of things, services, and users are crucial for the IoT [10]. Prior technology trends, e.g., cloud computing and big data, are likely to share security requirements with the IoT. Big data solutions, for instance, are designed to scale and deal with heterogeneity of data sources, not with an uncontrolled environment and constrained resources. Likewise, cloud computing by design is supposed to scale and overcome challenges of constrained resources, but it hardly deals with mobility of devices and physical accessibility of sensors. For sensor networks in IoT, the diversity of resource and the network heterogeneity makes security research much more difficult. Some of the security measures for IoT might be data encryption, key management schemes, secure efficient routing protocols, better intrusion detection algorithms, better physical security design, etc.

2.2.11 Trade-off between Security and Cost

Simple cryptographic functions such as logical addition, XOR, and various hash functions are easy to implement (require fewer resources) but easy to break in terms of their security.

2.2.12 Middleware Security

Middleware security is required for heterogeneous components to form a cohesive whole. For instance, for an API, add the levels of abstraction needed between transport layer and the application. This enables a degree of modularity that potentially brings systems closer to plug-and-play.

Some of the other challenges in security are authorized access of data, authentication, device, and identity management. Sensor networks are usually designed, developed, and used for specific application purposes such as environmental monitoring, agriculture, medical care, or event detection. Sensor Networks traffic characterization strongly depends on the application scenario. Sensor networks generally exist without IoT, but IoT cannot exist with sensor networks. For IoT purposes, sensor networks need to have a middleware, addressing issues of abstraction support, dynamic topology, application knowledge, programming paradigm, adaptability, scalability, and security. Also, efficient solutions for QoS support are needed.

2.3 Cryptography

Cryptography is studying different techniques concerned with keeping communication between two parties private in the presence of third parties. An encryption scheme has five ingredients: plaintext, encryption algorithm, secret key, ciphertext, and decryption algorithm. In these techniques, a message called plaintext will be converted at the sender party by a secret key and an algorithm or mathematical procedure such that the result, called ciphertext, appears non-sense for all parties. The used algorithm for encryption and decryption is available for all parties, while the secret key is shared only between the sender and the receiver. To protect data and systems against adversaries, the following four requirements are essential:

Confidentiality: Only the sender and the intended recipient of a communication can see the content of that communication. This concept is accomplished through encryption.

Data Integrity: It guarantees that the data received at the reception party is original and was received exactly as it was sent by the sender party. If the content of a communication is compromised, it must be detectable by either communicating party. Data integrity can be threatened either by environmental hazards, such as heat, dust, and electrical surges, or by attackers.

Authenticity: The sender and the recipient should be able to verify each other's identity. Any impostor needs to be either detected or identified.

Non-repudiation: It means preventing an entity from denying previous actions. In other words, the sender of the message cannot deny having sent the message.

Among these four services, confidentiality is the primary service, and all security algorithms are required to provide it, while other services are arbitrary.

2.4 Lightweight Cryptography

Lightweight cryptography is an innovative approach that concerns solutions to meet the challenge of developing fast and efficient security mechanisms for harsh resource constrained environments. These solutions include new design in cryptographic primitives and protocols in addition to adapting and modifying contemporary cryptosystems [11].

To design a lightweight cryptography, there are three things that are required to be optimized: security, performance, and cost. Security is measured with the number of bits of key. By increasing the size of the key, the provided security will be higher. Performance is considered in terms of the total number of clock cycles to complete an operation that is proportional with throughput and energy. Cost like power and area depends on the utilized architecture. Among these three objects, there is a trade-off that makes optimizing all of them together in one design very difficult (Figure 2-1). For example, security is in a trade-off with performance and cost. Having high security requires increasing either the number of rounds or cost. Performance and cost are two other vertexes of this triangle. Serialized architecture yields lower power and area, while it results in lower performance.

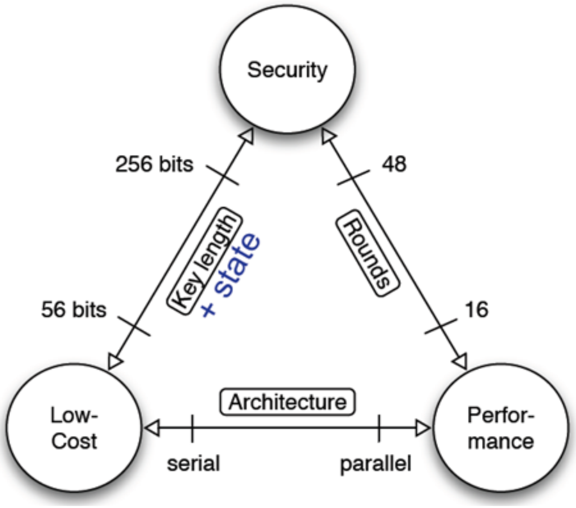


Figure 2-1 Design trade-offs for lightweight cryptography [11]

To have a more precise definition of lightweight cryptography, it is required to define the boundaries of cost and performance. Power consumption of the security implementation has to be reduced to 10s of microwatts, and for EEPROM read operation, this limitation should not exceed it unless the tag read range requirements cannot be preserved [12]. Performance is mainly limited by user requirements and air interface protocols. However, it is recommended to be 10s to 100s clock cycles.

In the following section, some insight into asymmetric and symmetric cryptography is given.

2.4.1 Asymmetric Key Encryption

Asymmetric key encryption algorithms, also called public key algorithms, are very strong in terms of security. They provide confidentiality, integrity, reliability, availability, and non-repudiation altogether. In this cryptography, two different keys are used: public key, which is published on the network, and private key, which is kept secret by user. To encrypt a plaintext, a public key is enough, but to decrypt the ciphertext, a corresponding private key is required. Thus every part can encrypt a message while only the party who has the private key can recover the message. Public-key constructions are typically based on some mathematical problem, such as factoring, which is assumed to be a hard problem in a computational sense. For example, in factoring, the private key can consist of two large prime numbers, and the corresponding public key is their product. Obtaining the private key from the public is possible in theory, but in practice, huge resources are required, e.g., time is required to compute it.

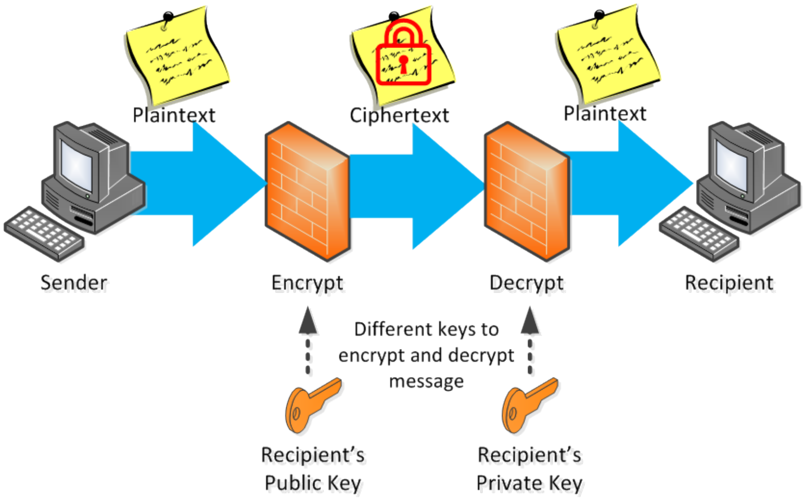


Figure 2-2 Asymmetric key encryption [11]

One of the advantages of Asymmetric key cryptosystems is distributing key among parties. Since it is not required for all parties to keep the encryption key in private, no key is required to be exchanged among involved parties.

2.4.2 Symmetric Key Encryption

Symmetric key encryption is the oldest and best-known technique to provide security in communications. In this technique, the sender and the receiver both share a secret key, which they have already agreed on. The shared key is used for both encryption and decryption (Figure 2-3). This setting, referred also to as private key cryptography, is considered to be confidential if only eligible parties whose have access to the shared secret key can recover the plaintext from the ciphertext.

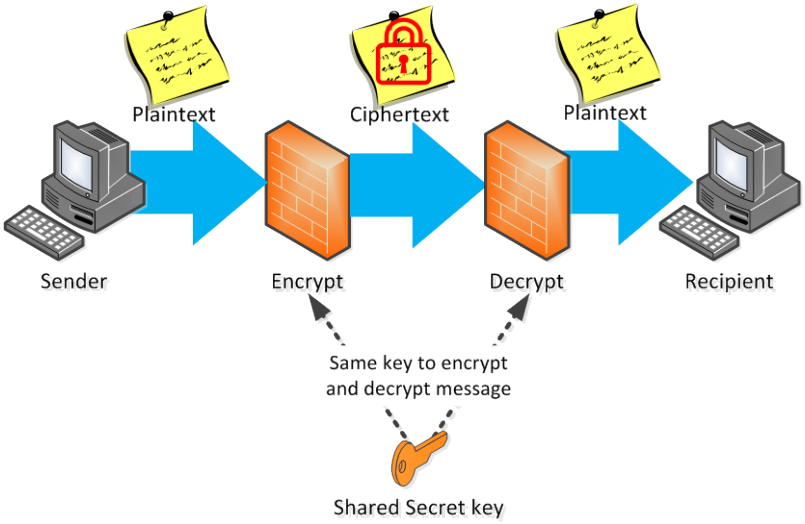


Figure 2-3 Symmetric key encryption [11]

There are several drawbacks that make symmetric key algorithms less interested in some applications. One of the obvious problems is distributing private keys among authorized parties. Moreover, keeping one secret key for each party makes managing keys more difficult by increasing the number of parties. Symmetric encryption algorithms cannot provide integrity and authentication alone. To provide these services, they need other algorithms to be integrated with them. Not supporting non-repudiation service is another problem of these cryptosystems. Despite all of these drawbacks, there are efficient software and hardware implementations for private key algorithms that make them suitable for restricted resource applications. Therefore, since Public key algorithms still have significant challenges for resource-constrained wireless sensors' implementation, recent research has been directed towards symmetric encryption while making them secure.

There are traditionally two classes of symmetric encryption algorithms: block ciphers and stream ciphers. Recently, a new class called hybrid cipher, which is a combination of these two ciphers, has been introduced as well.

2.5 Security Solution Approaches in Internet of Things

Different approaches are being employed for secure E2E communication in WSNs and IoT; they can be classified into major research directions as follows:

- Centralized Approaches
- Protocol-based Extensions and Optimizations
- Alternative Delegation Architectures
- Solutions that Require Special Purpose Hardware Modules

2.5.1 Centralized Approaches

Centralized security solution approaches are considered as efficient and suitable for the resource-constrained sensor networks, but the common issue is the scalability of the key management; the node must be pre-configured with shared keys of all entities before deployment. Some of the common centralized based approaches are SPINS (A centralized architecture for securing uni- and multicast communication in constrained networks, composed of two security protocols, SNEP and μTESLA) and the Polynomial-based scheme (Polynomial schemes aim at simplifying the key agreement process in distributed sensor networks; the main idea is to assign every node n a polynomial share F(n; y) derived from a secret symmetric bi-variate polynomial F(x; y). This allows any possible pair of nodes with a polynomial share to be able to establish a common secret) [13].

2.5.2 Protocol-based Extensions and Optimizations

Approaches such as compression aim at optimizing the protocol without breaking the security properties. There are several compression schemes proposed such as the compression of IPV6 header, extension headers, and UDP (User Datagram Protocol) header now standard in 6LoWPAN. Some of these approaches are Abbreviated DTLS Handshake (allows for a shorter handshake that reuses the state information from the previous session in order to resume the session). TLS Session Resumption without Server-Side State is when the server does not hold any state required to resume a session; rather, the server's encrypted state is offloaded during the handshake towards the client and in caching. TLS Cached Information extension allows for omitting cached information such as the large certificate chains from the handshake. Compression of header information is an approach to reduce the transmission overhead of packets in constrained environments; 6LoWPAN already defines header compression mechanism for IP packets.

2.5.3 Alternative Delegation Architectures

Delegate computationally intensive tasks, such as public-key-based operations involved in session establishments, to more powerful devices. Some important approaches are:

Server-based Certificate Validation Protocol (SCVP), which enables a client to delegate the complex task of certificate validation or certificate path construction to a trusted server. SCVP server should be trusted.

Another delegation approach: by Bonetto [14]. It delegates the public-key-based operations to a more powerful device, such as the Gateway (GW). They describe the procedure for IKE session establishment, where the GW intercepts session establishment and pretends to be the end-point. After calculation of the session key, this key is handed over to the constrained device, and both peers can directly protect their communication with the session key. But in the vision of IoT, a trusted GW is not always present e.g. in the home automation scenario, constrained devices of different manufacturers might be present in the constrained network.

Tiny 3-TLS [15]: It requires a strong trust level between the constrained resource device and the GW, offloads expensive public-key-based operations to the GW. The constrained resource device trusts the GW and the unconstrained device authenticates itself to the GW; hence, GW trusts the unconstrained device.

Consequently, Tiny 3-TLS assumes that by means of transitive trust, the constrained device could trust the unconstrained device. Tiny 3-TLS distinguishes between partially and fully trusted GWs.

Sizzle [16] implements a complete SSL-secured HTTP web server for constrained devices with support for ECC-based authentication. This approach, in contrast to previous delegation-based architectures, delegates only the task of adapting the underlying transport-layer protocol. This is achieved by terminating the incoming TCP connection at the GW and sending the payload via a UDP-based reliable protocol to the constrained device. Sizzle only allows for certificate-based authentication towards powerful clients and does not implement certificate handling for constrained devices.

Peer authentication and End-to-End data protection are crucial requirements to prevent eavesdropping on sensitive data or malicious triggering of harmful actuating tasks in the context of the Internet of Things (IoT). Symmetric key cryptography such as AES provides fast and lightweight encryption and decryption on smart devices, and their integrated hardware supports it as well. However, when the number of devices connected becomes high, exchanging symmetric keys becomes a challenging task, and an efficient scalable key establishment protocol is required. Asymmetric key cryptography is another method for key establishment at two ends, but it involves high computational overheads, which are the main concerns for resource-constrained devices [17]. Sensors with low resources (energy, computation) are not meant to perform complex asymmetric cryptographic operations.

Key establishment protocols are used to provide shared secrets between two or more parties, typically for subsequent use as private keys for a variety of cryptographic objectives [18]. These objectives are in turn used as security primitives for enabling various security protocols such as source authentication, integrity protection, or confidentiality [19]. To afford interoperable network security between endpoints from independent network domains, variants of traditional End-to-End IP security protocols have recently been proposed for resource-constrained devices and the networks formed by them [17].

- Protocol variants such as Datagram Transport Layer Security (DTLS) [1], HIP-DEX [21], and minimal IKEv2 [22] consider public-key cryptography in their protocol design. As public-key cryptography acquires significant computational processing and transmission overheads in resource-constrained network environments, research and standardization currently focuses to reduce the public-key related overheads during the protocol handshake.
- Another interesting approach has been suggested in [23] and [19]. In these papers, a proxy-based solution is proposed to delegate the heavy cryptographic operations from a resource-constrained device to less constrained nodes. A similar approach might be found in [24] for ambient-assisted living and also in [25], where communication is made from one resource-constrained node to another resource-constrained sensor node. These approaches have assumed the sensor nodes to be trustworthy and the mechanism in case if nodes are compromised, misbehave, authentication fails, or nodes fail to deliver the assigned share. Still, the risk involved is there for the secret

shared key to be revealed by the attacker from the compromised nodes. Selection criteria are described for these assisting nodes to evaluate their abilities before they are assigned computational tasks to work as proxies.

Other approaches proposed include session resumption mechanisms [17] and caching of static handshake information such as certificates [26]. However, the considerable RAM and ROM requirements make the use of public-key cryptography unsuitable for a wide range of constrained devices [17]. One such implementation of two-way authentication scheme for the IoT based on DTLS protocol is described in [27]. This approach even generates considerable overheads to the network traffic due to the utilization of X.509 certificates and RSA public keys with DTLS handshake. Both these X.509 certificate and RSA public key with DTLS handshake involve heavy computations for the low performing and high resource-constrained sensor nodes.

2.5.4 Solutions that Require Special Purpose Hardware Modules

A class of security solutions relies on additional hardware security modules, such as TPMs. A Trusted Platform Module (TPM) is tamper-proof hardware that provides support for cryptographic computations, especially public-key-based cryptographic primitives. TPMs can hold keys, such as RSA private keys, in a protected memory area. Furthermore, the cryptographic accelerator of TPMs is capable of computing the cryptographic computations with a higher performance. In contrast, ECC provides the same level of security with considerably smaller key sizes [13]. Therefore, ECC is preferred and recommended for constrained environments.

2.6 Related Works

Conventional security primitives cannot be applied due to the heterogeneous nature of sensors, low resources, and the system architecture in IoT applications. To prevent unauthorized use of user's data, protect their privacy, and to mitigate security and privacy threats, strong network security infrastructures are required. Peer authentication and End-to-End data protection are crucial requirements to prevent eavesdropping on sensitive data or malicious triggering of harmful actuating tasks [3].

Symmetric key cryptography such as AES provides fast and lightweight encryption and decryption on smart devices, and their integrated hardware supports it as well. However, when the number of devices connected becomes high, exchanging symmetric keys becomes a challenging task, and an efficient scalable key establishment protocol is required.

Asymmetric key cryptography is another method for key establishment at two ends, but it involves high computational overheads, which are the main concerns for resource-constrained devices. Sensors with low resources (energy, computation) are not meant to perform complex asymmetric cryptographic operations.

Security in IoT context involves secure End-to-End communications; an IoT node can be expected to act alternatively as a client and as a server contrary to the wireless sensor network. Only two participants involved at the ends in the pair-wise key exchange protocol should have access to the agreed secret key [19].

Key establishment protocols are used to provide shared secrets between two or more parties, typically for subsequent use as private keys for a variety of cryptographic objectives [18]. These objectives are in turn used as security primitives for enabling various security protocols such as source authentication, integrity protection, or confidentiality [19].

To afford interoperable network security between endpoints from independent network domains, variants of traditional End-to-End IP security protocols have recently been proposed for resource-constrained devices and the networks formed by them [17].

- Protocol variants such as Datagram Transport Layer Security (DTLS) [1], HIP-DEX [21], and minimal IKEv2 [20] consider public-key cryptography in their protocol design. As public-key cryptography acquires significant computational processing and transmission overheads in resource-constrained network environments, research and standardization currently focuses to reduce the public-key related overheads during the protocol handshake.
- Other approaches proposed include session resumption mechanisms [26] and caching of static handshake information such as certificates [27]. However, the considerable RAM and ROM requirements make the use of public-key cryptography unsuitable for a wide range of constrained devices [29]. One such implementation of two-way authentication scheme for IoT based on DTLS protocol is described in [27]. This approach even generates considerable overheads to the network traffic due to the utilization of X.509 certificates and RSA public keys with DTLS handshake. Both these X.509 certificate and RSA public key with DTLS handshake involve heavy computations for the low performing and high resource-constrained sensor nodes.
- Another interesting approach has been suggested in [23] and [19]. In these papers, a proxy-based solution is proposed to delegate the heavy cryptographic operations from a resource-constrained device to less constrained nodes. A similar approach might be found in [30] for ambient-assisted living and also in [31] where communication is made from one resource-constrained node to another resource-constrained sensor

node.

    - These approaches have assumed the nodes to be trustworthy, and there's no mechanism if nodes are compromised, misbehave, authentication fails, or nodes fail to deliver the assigned share.
    - There is risk involved for the secret shared key to be revealed by the attacker from the compromised nodes.
    - There is no selection criteria described for these assisting nodes to evaluate their abilities before they are assigned computational tasks to work as proxies.

- There are several compression schemes proposed such as the compression of IPV6 header, extension headers, and UDP (User Datagram Protocol) header now standard in 6LoWPAN. The authors in [31] have presented 6LoWPAN compressions for IPsec payload headers for authentication header and encapsulating security payload.

As we know, Wireless sensor networks (WSNs) comprise small battery-powered and low-cost devices, each with sensing, data processing, and communication capabilities. The ZigBee specification is an emerging wireless technology designed to address the specific needs of low-cost, low-power wireless sensor networks, but the constraints of WSNs make them vulnerable to attacks like including denial of service, traffic analysis, and node replication. Localized Encryption and Authentication Protocol (LEAP+) is a distributed symmetric-based key management protocol designed to support multiple types of keys based on the message type that is being exchanged. In CHAPTER 3, we first conduct a detailed qualitative security analysis of LEAP+ and the current ZigBee key management schemes and then have investigated the feasibility of implementing LEAP+ onto the ZigBee stack to get improved security and better scalability for Wireless Sensor Networks (WSNs).

2.7 Summary

This chapter aims to provide the reader with a basic overview about the major security and privacy challenges in the Internet of Things, how wireless sensors are integrated into IoT, an overview of cryptography, and what kind of security primitives and solution approaches are being taken to make communication secure and to protect the user's data. Moreover, related works are also given to compare our proposed protocols for security and energy efficiency. Conventional security primitives cannot be applied due to the heterogeneous nature of sensors, low resources, and the system architecture in IoT applications. To prevent unauthorized use of a user's data, protect their privacy, and to mitigate security and privacy threats, strong network security infrastructures are required. Peer authentication and End-to-End data protection are crucial requirements to prevent eavesdropping on sensitive data or malicious triggering of harmful actuating tasks. Any unauthorized use of data may restrict users to utilize IoT based applications. This review chapter provides the security solution approaches that have been proposed recently in identifying both the challenges related to security and privacy and the attack techniques used to compromise/fail the sensor nodes in the Internet of Things as well.

Current approaches are focused on pre-deployed, pre-shared keys on both ends, whereas certificate-based authentication is generally considered infeasible for constrained resource sensors. New security paradigms are needed for End-to-End secure key establishment protocols that are lightweight for resource-constrained sensors and secure through strong encryption and authentication. Further detail is provided on key establishment protocols and the network scenarios in CHAPTERS 4 and 5.

CHAPTER 3: LEAP+ in ZigBee Specification

The ZigBee specification is an emerging wireless technology designed to address the specific needs of low-cost, low-power wireless sensor networks and is built upon the physical and medium access control layers defined in the IEEE 802.15.4 standard for wireless personal area networks (WPANs). A key component for the widespread success and applicability of ZigBee-based networking solutions will be its ability to provide enhanced security mechanisms that can scale to hundreds of nodes. Currently, however, an area of concern is the ZigBee key management scheme uses a centralized approach that introduces well-known issues of limited scalability and a single point of vulnerability.

Moreover, ZigBee key management uses a public key infrastructure. Due to these limitations, we suggest replacing ZigBee key management with a better candidate scheme that is decentralized, symmetric, and scalable while addressing security requirements. In this work, we investigate the feasibility of implementing Localized Encryption and Authentication Protocol (LEAP+), a distributed symmetric-based key management scheme. LEAP+ is designed to support multiple types of keys based on the message type that is being exchanged. In this paper, we first conduct a qualitative security analysis of LEAP+ and the current ZigBee key management scheme. Using the QualNet 5.0.2 simulator, we implement LEAP+ on the ZigBee platform for the very first time. Experimental results show that a distributed key management scheme such as LEAP+ provides improved security and offers good scalability. Wireless sensor networks (WSNs) comprise small battery-powered and low-cost devices each with sensing, data processing, and communication capabilities. ZigBee is an emerging standard that aims to address applications in a wide range of markets, including commercial building automation, residential appliance networks, healthcare, fitness, telecommunication, and even military, police, safety, and rescue applications [32]. In ZigBee networks, a large number of sensor nodes are deployed to monitor a wide area where the working situations are commonly tough. Since these nodes are typically positioned in distant locations, and they might have mission critical tasks, they should be armed with security appliances to provide information assurance against any unwanted information leakage [33]. Unfortunately, the constraints of WSNs make them more vulnerable to attacks, including denial of service (DoS), traffic analysis, and node replication. Even jamming mitigation techniques are not generally feasible in WSNs to use against DoS due to their design complexity and high energy consumption [34, 35].

The ZigBee specification includes a number of security provisions and options [36] while improving the basic security framework defined in IEEE 802.15.4 [37]. ZigBee security service facilitates carrying out secure communications, establishing of cryptographic keys, and controlling devices. Indeed, key management is a core mechanism for any other security services in ZigBee protocol stack. The objectives of this key management are to generate and securely distribute required cryptographic keys between the communicating nodes that need to transfer data [38]. Unfortunately, ZigBee asymmetric-based key establishment is not efficient (e.g., Diffie-Hellman key establishment protocol [39]) due to energy consumption and hardware requirements [40, 35]. Sufficient security cannot be provided by ZigBee when large sensor networks are employed [41]. ZigBee key management scheme is not too flexible for node addition and revocation while working in any desired environments. Moreover, ZigBee key management is not meant for distributed application due to its centralized design. This results in a need to find an efficient and reliable candidate scheme to replace ZigBee key management to overcome such limitations and security issues.

Hence, in this work, we select LEAP+ [42] to be implemented as an alternative to ZigBee key management scheme on the ZigBee stack. Using decentralized key management such as LEAP+, however, has its own costs to be paid to satisfy anticipated security requirements. Our contributions include: first, a detailed security analysis and comparison of the LEAP+ and ZigBee key management schemes; second, the implementation and integration of LEAP+ in the ZigBee protocol stack as an alternative to the standard ZigBee key management.

3.1 ZigBee Specification and Security

ZigBee specification is considered a reliable, low-power, wirelessly networked monitoring and control product by the ZigBee Alliance [36]. ZigBee comprises IEEE 802.15.4 for the physical and MAC layers along with its support for network and application layers and places itself on top of the IEEE 802.15.4 layers as shown in Figure 3-1.

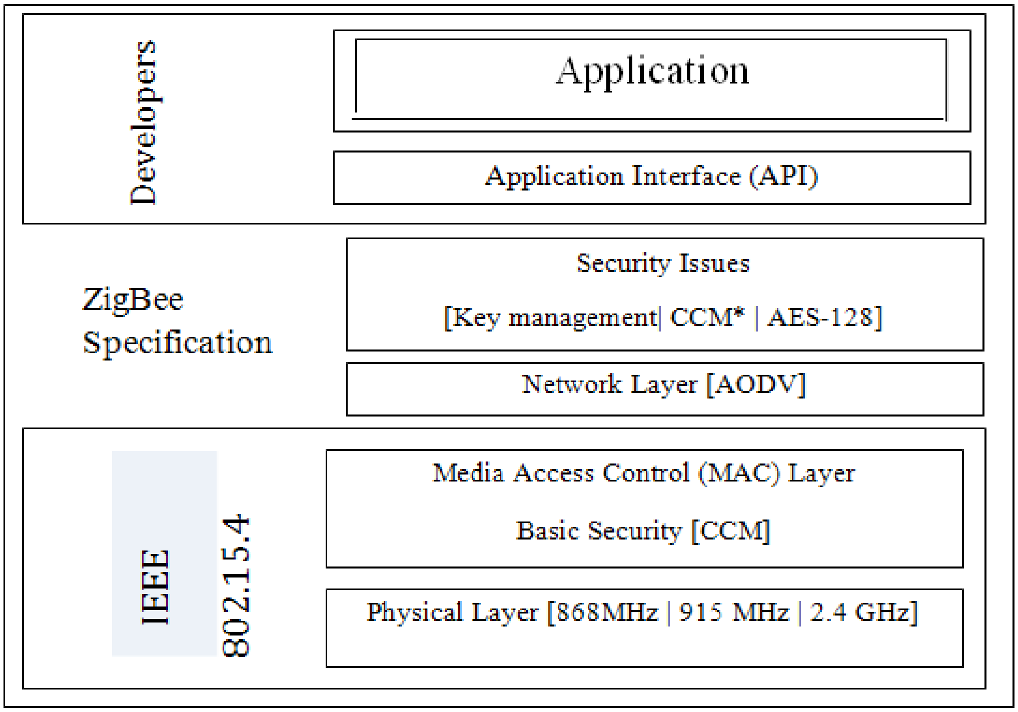

Developers
Application
Application Interface (API)
ZigBee Specification
Security Issues
[Key management| CCM* | AES-128]
Network Layer [AODV]
IEEE 802.15.4
Media Access Control (MAC) Layer
Basic Security [CCM]
Physical Layer [868MHz | 915 MHz | 2.4 GHz]

Figure 3-1 The ZigBee protocol stack

The IEEE 802.15.4 standard applies to WPAN that operate at low data rate wireless connectivity and are confined to operate in the range of 10m [37]. ZigBee supports various levels of security that can be configured depending on the needs of the application. It includes methods for key establishment, key transport, frame protection, and device management [36, 43]. ZigBee provides three types of security modes: residential, standard, and high security. Residential security is first supported in the ZigBee 2006 standard [36]. This level of security requires a network key to be shared among devices and is designed for lower security residential applications. Standard security adds a number of optional security enhancements over residential security, including an application support sub-layer (APS layer) link key. High security (commercial) adds entity authentication and a number of other features not widely supported. This mode is intended to be implemented for high security commercial applications. ZigBee high security utilizes three types of keys: master key, link key, and network key. The master key is used for secure communication between nodes and the base station. The link key is shared by two devices for secure uni-cast communications, whereas the network key is used for broadcast communications and is shared among all devices in the network; both of these types of keys can be updated periodically. The base station (Trust Center device) authenticates devices that are going to join the network. This Trust Center takes care of the link key distribution in the network and also selects a proper network key. All the devices, therefore, must be pre-configured with the proper link key to enhance the network security. Fundamentally, all the keys are delivered via pre-installation, key-Transport, or key-Establishment methods as defined by ZigBee [36]. In the pre-installation method, keys are loaded before placement in the network.

Key-Transport technique: the Trust Center transmits the key in a secure fashion to the device whenever possible. Any node may obtain its network key via key-transport or pre-installation. In the Key-Establishment approach, the link key establishment is processed by a Symmetric-Key Key Establishment (SKKE) protocol [44], and it is based on the master key. The master key itself is acquired via key-transport or pre-installation [44]. The network key is securely transported between the Trust Center and the device by using a link key based on the 128-bit Advanced Encryption Standard (AES) encryption algorithm [45]. First, the Trust Center encrypts the network key by using a link key and sends the encrypted data; then the device decrypts the received data by using the link key. The master key is a secret key between two nodes and provides a starting point for establishing a link key. This task can be done via other mechanisms such as Certificate-based Key Establishment (CBKE) and Alpha-secure Key Establishment (ASKE) [46]. Besides all the security specifications of ZigBee, WSNs have similar type of security requirements to those of ad-hoc networks [47, 48]. There are general and specific security requirements for any key management of WSNs as discussed in [48, 49]. One issue is the centralized nature of ZigBee key management. If the Trust Center is compromised, then the whole network becomes compromised since it has a single point of exposure. Another issue is the scalability of the network. ZigBee cannot supply sufficient security when large sensor networks appear. Such shortcomings give a motivation to use any other key management protocol such as LEAP+.

3.2 Analysis

In this section, we draw a comparison of different features of original ZigBee’s key management scheme and LEAP+ that is given in Table 3-1. ZigBee’s key management scheme is a centralized scheme that relies on a Trust Center having both public and private keys, whereas LEAP+ is a distributed key management scheme having a symmetric key as its main cryptosystem. Obviously, asymmetric cryptosystem does not scale well in a large network that consists of hundreds of devices. However, this scalability issue is addressed in LEAP+ by eliminating the distribution center for key management [50].

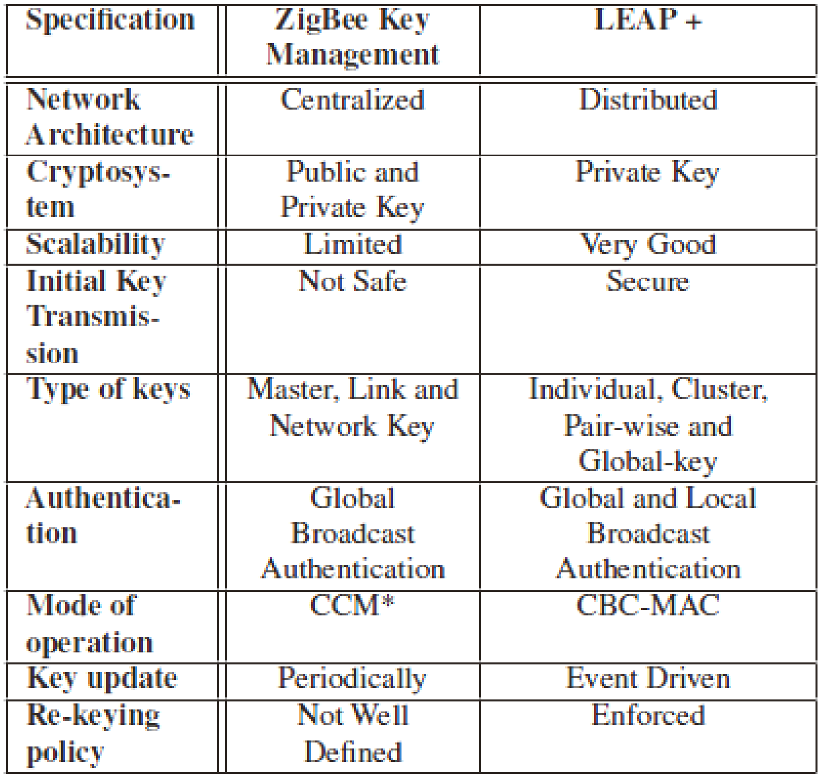
3.2.1 ZigBee Specification

ZigBee provides three types of keys that are preloaded before node deployment [36]. Both link and network keys can be updated either manually or online. ZigBee does not have a proper mechanism to transmit the master key to each node in a secure fashion, resulting in utilizing public key system technology that incorporates performance overheads. LEAP+ provides four types of keys where individual and global keys are preloaded before deployment [42]. Both the pair-wise and cluster keys are generated and established after node deployment and discovery of its own immediate neighbors. ZigBee does not support a practical node revocation mechanism. Therefore, if a node is captured, the keys might become available to the adversary. The ZigBee re-keying policy is not well-defined even for the Smart Energy Profile [36, 43] and it does not have proper re-keying mechanisms, increasing security and efficiency concerns. In LEAP+, all four type of keys can be revoked and updated via LEAP+'s revocation and re-keying mechanism where the re-keying and revocation policies are comprehensively defined and enforced.

3.2.2 LEAP+ key management scheme

Unlike ZigBee that offers global broadcast authentication, LEAP+ supports both local and global broadcast authentications without preventing passive participation initially inherited from µTESLA. Employing local broadcast authentication has its own performance benefits, especially where the event- or time-driven messages can be locally authenticated, such as routing control messages or aggregated sensor readings. Having local broadcast authentication mechanism eliminates potential associated costs that can be introduced by global broadcast authentication [42]. Routing control messages or aggregated sensor readings are examples of event- or time-driven local broadcasts that do not impose delay and energy overheads as global broadcasts usually have. ZigBee uses the CCM* [51] mode of operation, which is a general combined encryption and authentication block cipher mode with a block size of 128-bit such as AES-128. To extend it to other block sizes requires further definitions, again increasing the performance overhead, whereas CBC-MAC [52] is used in LEAP+, where block size is fixed to offer authentication. Thus, a key management scheme that scales a large flexible network with decentralized controller is found in LEAP+, making it an obvious potential choice to replace ZigBee's key management scheme.

Table 3-1 Comparison between ZigBee and LEAP+ Key Management Schemes

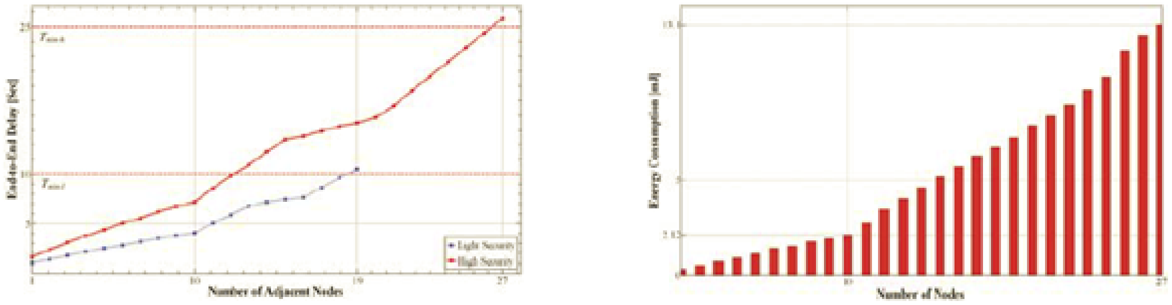


Specification	ZigBee Key Management	LEAP +
Network Architecture	Centralized	Distributed
Cryptosystem	Public and Private Key	Private Key
Scalability	Limited	Very Good
Initial Key Transmission	Not Safe	Secure
Type of keys	Master, Link and Network Key	Individual, Cluster, Pair-wise and Global-key
Authentication	Global Broadcast Authentication	Global and Local Broadcast Authentication
Mode of operation	CCM*	CBC-MAC
Key update	Periodically	Event Driven
Re-keying policy	Not Well Defined	Enforced

3.3 Implementation and Performance Evaluation

This section describes how the cryptographic functions presented in LEAP+ are implemented on top of the ZigBee stack as part of application layer. The LEAP+ Key management scheme routines are coded in C++ and introduced as part of simulation functions of QualNet. LEAP+ requires some specific cryptographic functions such as key generation, message authentication, and encryption. To meet these requirements, RC5 is used as the random key generation and encryption function. Respectively, Cipher-based Message Authentication Code (CMAC) is utilized to check integrity and confidentiality of data and to authenticate the communicating entities [53, 54, 52]. RC5 is appropriate for WSNs applications due to its low memory requirement. Moreover, the RC5 block cipher has embedded parameter variability to get flexibility at all levels of security and efficiency. RC5 is also employed as part of CMAC implementation. We have configured two levels of security (high and light security) by modifying data dependent rotations (via increasing RC5 rounds from 16 to 64) to be used in our designed scenarios. We performed experiments for three different scenarios. These experiments are replicated ten times for each scenario.

The nodes are Fully Functional Device (FFD), im-mobile and randomly placed within the areas of 20×20 25×25, 30×30 and 50×50 square meters. In the first scenario, a node establishes a unique pair-wise key with each node within its communication range (immediate neighbors). In each step, we deploy a new node within this range. We continue adding new nodes up to the point that the time delay reaches the T_{min} threshold. We adopt two different thresholds. When high security is required, T_{min-h} is the maximum tolerable time delay allowed by the protocol for the nodes to establish their pair-wise keys, and T_{min-l} is used when higher performance is required while still maintaining a minimum level of security. We also adopt the value of 10 seconds for T_{min-l} based on the performed experiment in [42], and we assume the value 25 sec for T_{min-h}. Note that these values can be decided differently (based on application requirements) since the time to establish pair-wise keys is far less than the time for an adversary to obtain copies of all the memory and data on the captured sensor node.



Time delay for a node to establish pair-wise keys with all its neighbors for the high and light security configurations

Energy consumption for a node to establish pair-wise keys with all its neighbors

Figure 3-2 Time delay for a node to establish pair-wise keys with its all neighbors

Our main objectives of this experiment are to indicate the maximum number of nodes that can be placed within the radio range of the aiming node, and the impacts of network density on both time delay and energy consumption. The total number of nodes n within the area of communication range r of the aiming node defines the node density $d = (n/\pi r^2)$. Figure 3-2a depicts the time that is taken for added nodes (27 nodes in high security and 19 nodes for light security) to generate and deliver pair-wise keys to a target node within one cluster. Correspondingly, energy consumption due to these key transmissions is shown in Fig. 3.2b. It is observed that when the number of nodes in a network is increased, more time is taken for a node to make a pair-wise key with the other neighboring nodes and also more energy is consumed for transmissions and communications accordingly. The simulation statistic file shows that for the first ten nodes, the total number of Bytes sent by the sending node is identical to received Bytes by receiving nodes; this confirms that there is no congestion, and as a result, the graph inclines linearly. Immediately after adding more nodes to this cluster, we observe that the time delay increases exponentially. This is due to retransmission of sending packets in some cases up to five times. This introduced packet loss ratio is mainly caused by network congestion. At the same point, energy consumption grows at an altered rate, as shown in Figure 3-2b. Moreover, monitoring the physical status of the targeting node shows that it spends over 40% of time in Transmit and Receive mode, rather than Idle or Sleep mode.

To demonstrate the scalability of LEAP+, we expanded the first scenario. We ran the simulation for different numbers of nodes and areas of network. In contrast from the previous observation, we consider time delay for the entire network due to pair-wise key establishment. First, we begin establishing pair-wise keys for a network size of 20 × 20. Once we reach the time threshold, we stop adding nodes to the network. We then ran the same simulation for a larger network size of 25 × 25. As soon as we hit the thresholds, we proceed with network sizes of 30 × 30 and 50 × 50 square meters. Figure 3-3 exhibits the time taken for the entire network to obtain the corresponding pair-wise Keys. When operating in an area of 20 × 20 square meters with the light security configuration, the frequency spectrum gets congested when the number of nodes reaches nineteen. Delay due to this congestion can put the key establishment protocol in a compromised state. The only way to add more nodes to the network is to provide more room by increasing the network area, thereby decreasing average network density. However, we have to keep the average density to a reasonable amount where the network is still connected (we enforce this in the simulation by not having any isolated nodes). Also, the average density of the network must be less than or equal to the node density calculated from the first scenario in order to maintain the pair-wise key delay always less than the time threshold. Let n be the total number of nodes in a cluster, A be the network area, and r be the communication range of a single node. The expected maximum admitted nodes N to the network can be calculated as:

$$N \leq n\left(\frac{A}{\pi r^2}\right) \tag{3-1}$$

For instance, given a network region of 30 × 30 square meters and radio range of 10 meters, the cluster size is 19 nodes where nodes are positioned uniformly. The total number of admitted nodes to the network should not be more than 54 nodes for the light security configuration. For the same setup, our simulation result confirms, that the network size of 53 nodes does not exceed the threshold of 10 sec (see Figure 3-3).

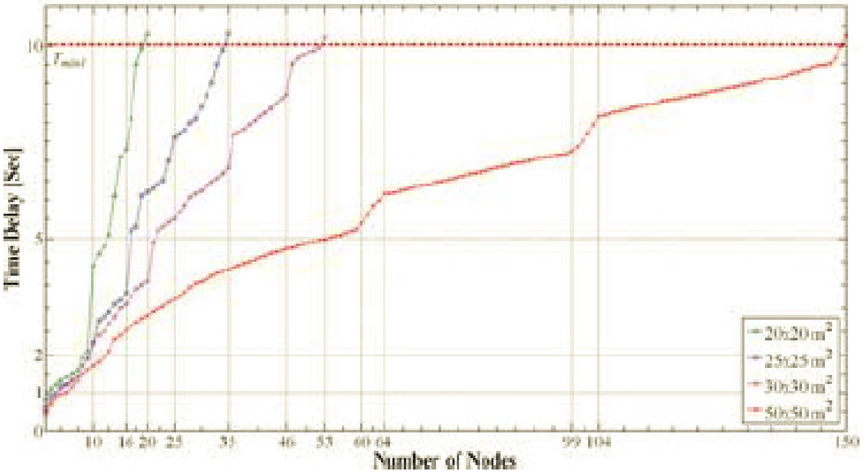


Figure 3-3 Time delay to establish pair-wise keys for the light security configuration

The high security configuration anticipates more delays while providing more security over the light security setup. Figure 3-3 shows simulation results for this setup; these delays might not be interesting for some applications where the performance aspects of key management require more attention. Furthermore, having a higher time threshold in the high security setup allows more node admission to the network that increases the overall scalability and connectivity of network.

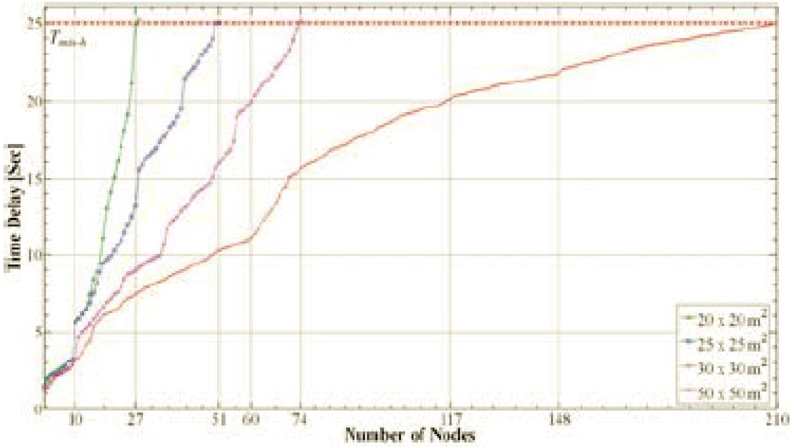


Figure 3-4 Time delay for the entire network to establish pair-wise keys for the high security configuration

As mentioned in Section III, cluster-key is used for secure broadcast within a cluster. Cluster formation happens right after pair-wise key establishment, and these cluster keys are securely transported by pair-wise key encryption. In the second scenario, we validate the operation of cluster key establishment procedure. In this setup, for all nodes added to the network in first scenario, a unique cluster-key is established. The extra cost of this secure mechanism (cluster-key establishment) is calculated in term of time delays and energy consumption. Fortunately, LEAP+ did not fix the time threshold for the key delivery and left this option open for application developers to decide. It is evident from Figure 3-5 that the total time delays to establish cluster keys increases with regard to the number of nodes being increased in a network, and it is the same behavior for the energy consumptions as described in Figure 3-6.

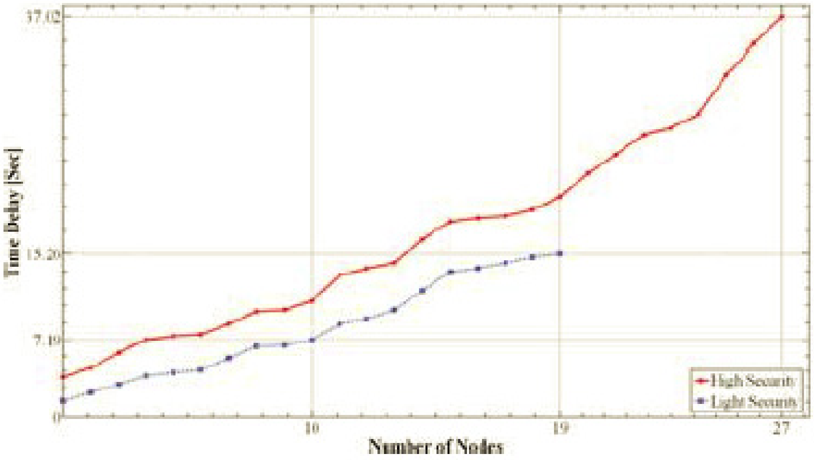


Figure 3-5 Time delay for all the nodes within a cluster to generate and exchange their cluster-keys for the high and light security configurations

Increasing the number of nodes impacts the end-to-end delivery and energy consumption for the same reasons described for the first scenario. The only difference is that there is additional time and energy taken by the encryption and decryption for the cluster-keys transmissions. The cluster-keys are transported securely (encrypted with pairwise key); therefore, there will be several uni-cast secure sessions between each set of immediate neighbors for secure exchanging of the cluster-keys.

Finally, in the third scenario, compromised nodes are consecutively revoked from the aforementioned cluster. Rekeying procedures subsequently are performed to stabilize the connectivity of the network by recovering disconnected paths. When a node is revoked, all nodes that are neighbors of the revoked node need to encrypt their new cluster-keys using the pair-wise key shared with each neighbor. Therefore, the numbers of such secure key transportations are determined by the number of neighbors and the density of the sensor network.

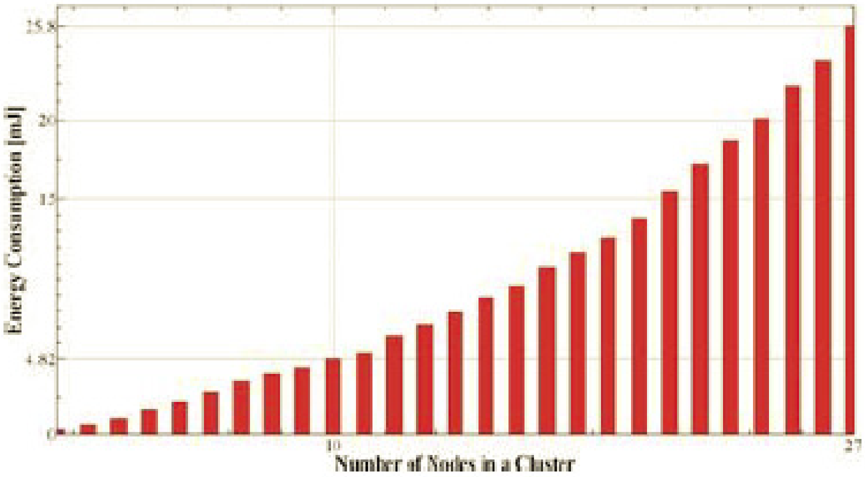


Figure 3-6 Energy consumption for all the nodes within a cluster to generate and exchange their cluster-keys

Figure 3-7 shows the total time taken to revoke a node from the network. If a node is revoked from a cluster that comprises 27 nodes, there is a delay of 4.37 seconds at most for the high security profile. This time is taken to inform all of the 26 remaining nodes and let them authenticate the revocation message coming from the base station. For a network with reasonable density, it seems that transmission time delays do not cause many performance problems in LEAP+. For example, for a network of 26 nodes, the total re-keying time increased from 59.01 seconds as depicted in Figure 3-8. All of this happens within a single cluster. Therefore, if there is a need to deploy additional nodes, LEAP+ provides the possibility to increase the number of clusters instead of overcrowding a single cluster as was argued in previous scenarios.

We evaluate the computation and communication costs of the LEAP+ key establishment schemes for each of the aforementioned scenarios. This paper does not provide a quantitative comparison among ZigBee key management and LEAP+ schemes based on mentioned costs. The reason is that both these protocols have different fundamental characteristics. First, in a distributed design such as LEAP+, there is no single building block for key establishment, whereas in a centralized architecture the main controller organizes key establishment.

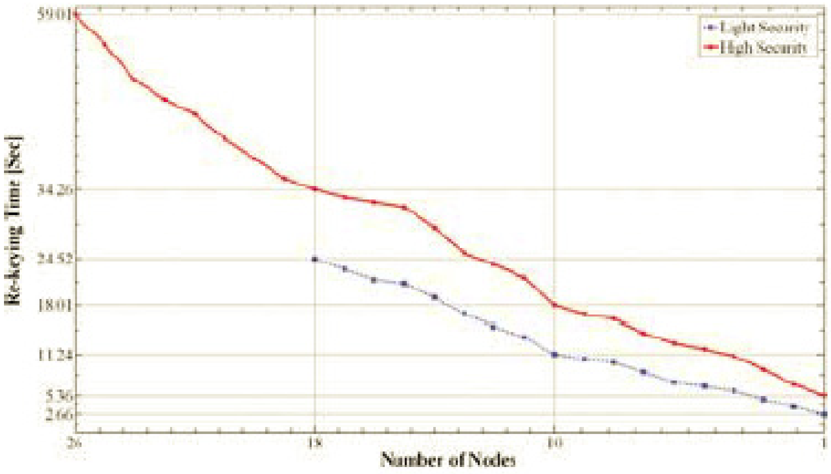


Figure 3-7 Time delay for nodes within a cluster to update and re-establish keys (re-keying) with the other nodes for the high and light security configurations

Therefore, the method of key distribution will be different and cannot be compared within the same scenarios. Second, both pair-wise keys and cluster-keys in LEAP+ are established after node deployment, whereas link and network keys in ZigBee are preloaded. Also, LEAP+ keys can be regenerated and reestablished securely after authentic revocation by the base station as in our third scenario. The achieved results help network application developers to have a clear picture about both security features and additional communication overheads introduced by LEAP+. Inspecting Table 3-1, it is possible to see the benefits of LEAP+, but these have associated costs, which are examined in this section.

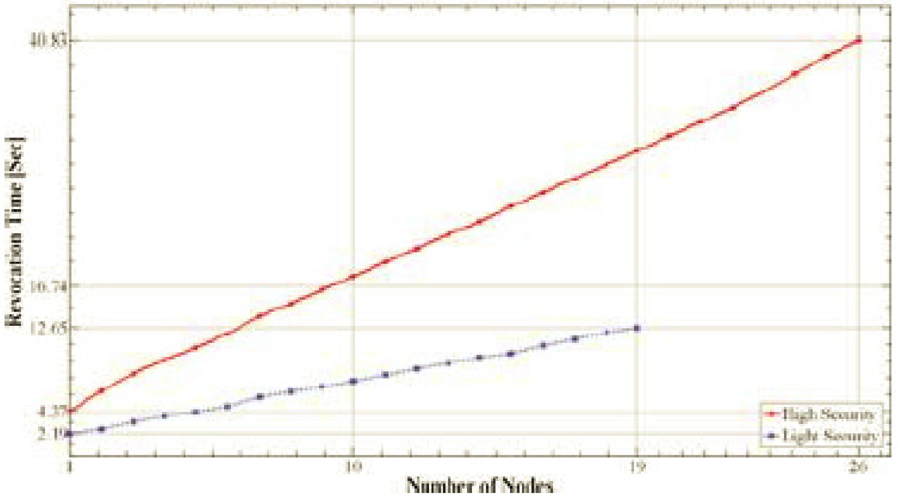


Figure 3-8 Time delay for all the nodes within a cluster to revoke a node

3.4 Summary

In this chapter, we have described the work substituting the key management scheme of ZigBee by implementing LEAP+ to enhance the security capabilities. LEAP+ is a symmetric distributed key management protocol for sensor networks that is designed to support multi-type keys depending on the type of message that is being exchanged. In fact, LEAP+ forms the network into overlapping small clusters, providing the possibility to have better security by reducing the risks of information leakage that are caused by broad information exchanged. LEAP+ is surprisingly well-suited to different types of network topologies, device types, and addressing modes offered by ZigBee stack. Our experimental results and performance evaluation parameters are not only valuable to assess the feasibility of LEAP+ scheme on the ZigBee protocol stack, but they also provide the basis for having an effective mechanism to get reasonable scalability within WSNs. There is, however, a significant point to be considered. That is, LEAP+ is essentially meant for stationary nodes. Mobility of the nodes within the network is highly significant for mobile wireless sensor networks and needs to be considered for future work. This need inspires the idea to upgrade LEAP+ with mobile capability, keeping in mind that there are a lot of design challenges and potential issues that must be addressed and resolved in order to enable mobility in sensor networks to get enhanced security and reduced performance overheads.

CHAPTER 4: Secure End-to-End Key Establishment Protocols

4.1 Introduction to key establishment Protocols in context of IoT

Internet of Things (IoT) is an ubiquitous concept where physical objects are connected over the internet and are provided with unique identifiers to enable their self-identification to other devices and the ability to transmit data over the network. Sensor nodes along with their heterogeneous nature are the main part of IoT. Communication security and end-users privacy protection is a major concern in the development of IoT, especially if these IP-enabled sensor nodes have limited resources. Security in IoT context involves End-to-End communications; an IoT node can be expected to act alternatively as a client and as a server, contrary to the wireless sensor network. This implies that providing security means only the two participants involved at the ends in the pair-wise key exchange protocol should have access to the agreed secret key. By having mutual authentication, these two peers should also authenticate each other and link the generated keys to their respective identities [19]. We have considered the healthcare application for our system model where sensors installed or implanted on the body are supposed to have low (energy, computation) resources and are not meant to perform complex asymmetric cryptographic operations. Secret key distribution for heterogeneous sensors becomes challenging due to the inconsistencies in their cryptographic primitives and computational resources as in healthcare applications. This chapter introduces new End-to-End key establishment protocols that are lightweight for resource-constrained sensors and secure through strong encryption and authentication. By using these protocols, resource-constrained nodes can also benefit from the same security functionalities that are typical of unconstrained domains without having to execute computationally intensive operations. The main concept is based on cooperation by offloading the heavy cryptographic operations of constrained nodes to the neighboring trusted nodes or devices. Security analysis and performance evaluation results (described in CHAPTER 5) show that the proposed protocol is secure and is sufficiently energy efficient.

Key establishment protocols are used to provide shared secrets between two or more parties, typically for subsequent use as private keys for a variety of cryptographic objectives [28]. These objectives are in turn used as security primitives for enabling various security protocols such as source authentication, integrity protection, or confidentiality [19]. Security in IoT must ensure secrecy and integrity of communication as well as the authenticity of messages being exchanged. There are various challenges to design security solutions in IoT because of network characteristics e.g., device heterogeneity, resource constraints, unreliable communication links, and the distributed nature. From the end-user's perspective, it is not possible to easily modify these smart devices; security primitives must be pre-embedded into the system. The integration of sensors in the internet must ensure the interoperability, transparency, and flexibility. However, sensor nodes inherently have constrained resources with regards to the processing power, memory, communication bandwidth, and energy, especially in healthcare and well-being applications in the context of IoT. Small batteries are typically the main energy sources for these sensor nodes with the requirement to operate for longer durations. Hence, energy efficiency becomes an important factor besides security and privacy issues.

Conventional security primitives cannot be applied due to the heterogeneous nature of sensors (either implanted, on-body, or wearable), low resources, and the system architecture of IoT based healthcare systems. Physiological data measurements are collected and transmitted to remote servers to analyze the medical data and to intervene in case of an emergency. Any unauthorized use of a patient’s data or privacy concerns may restrict people to utilize IoT-based healthcare applications. To mitigate these security and privacy threats, strong network security infrastructures are required. Peer authentication and End-to-End data protection are crucial requirements to prevent eavesdropping on sensitive data or malicious triggering of harmful actuating tasks [3].

Symmetric key cryptography such as AES provides fast and lightweight encryption and decryption on smart devices, and their integrated hardware supports it as well. However, when the number of devices connected becomes high, exchanging symmetric keys becomes a challenging task, and therefore, an efficient scalable key establishment protocol is required. Asymmetric key cryptography is another method for key establishment at two ends, but it involves high computational overheads, which are the main concerns for resource constrained devices [55].

In this paper, we propose to offload the cryptographic computational load to less resource-constrained nodes/devices in a cooperative way through exploiting the resources heterogeneity of devices in IoT. The proposed protocol aims to establish shared secret keys in a secure and efficient way to provide confidentiality and authentication while exchanging data. Session keys are required to update after a certain period of time or data-counts to avoid any security breach. The proposed key establishment protocol turns out to be lightweight for resource-constrained sensor nodes through getting assistance from neighboring powerful nodes in order to perform high computational cryptographic operations. The selection of these assisting nodes is made based on the trust level on them, their abilities, past cooperating performance, and the availability of resources. Moreover, the proposed protocol is able to identify the neighboring nodes that don’t cooperate and fail to deliver the assigned shared part during the key establishment process. This work is important due to its adaptability with other existing End-to-End secure communication solutions in IoT.

4.2 System Architecture

IoT provides appropriate solutions for a wide range of applications such as smart cities, traffic congestion, waste management, structural health, security, emergency services, logistics, retails, industrial control, and healthcare. Several instances of these applications require establishing secure End-to-End connections between resource-constrained devices. For example, in healthcare applications, a remote server (doctor/hospital server) may be required to get health-related sensitive data from highly resource-constrained sensors. This data can be supplemented with context information (e.g., date, time, location, and temperature). The architecture of an IoT-based health monitoring system is illustrated in Figure 4-1. Both the remote server and resource-constrained sensor require having a secure End-to-End communication link between them. Both these two ends first need to authenticate each other and securely establish a secret key for encryption of transmitted data.

4.2.1 Key Establishment Requirements

Most of the current internet key establishment protocols are primarily based on asymmetric cryptographic techniques; either they are for key exchange/agreement or for the authentication method employed within the protocol itself. Symmetric key transport protocols also exist, but they basically comprise in key refresh or key derivation schemes so are not good candidates to be considered as key establishment protocols. Also, symmetric key agreement protocols call for complex setup (pre-distribution) so are not very common [19].

As mentioned before, IoT based network require the End-to-End secure communications and the pervasiveness where any two given nodes may have to interoperate with each other without considering their respective nature. Special care should therefore be taken when designing an IoT key establishment protocol to make sure two nodes with significant differences in their resource capabilities are still able to communicate with each other. Efficiency of the key establishment protocol becomes important as well along with adoptability when it involves highly resource-constrained nodes with their low computational power and low battery capacity.

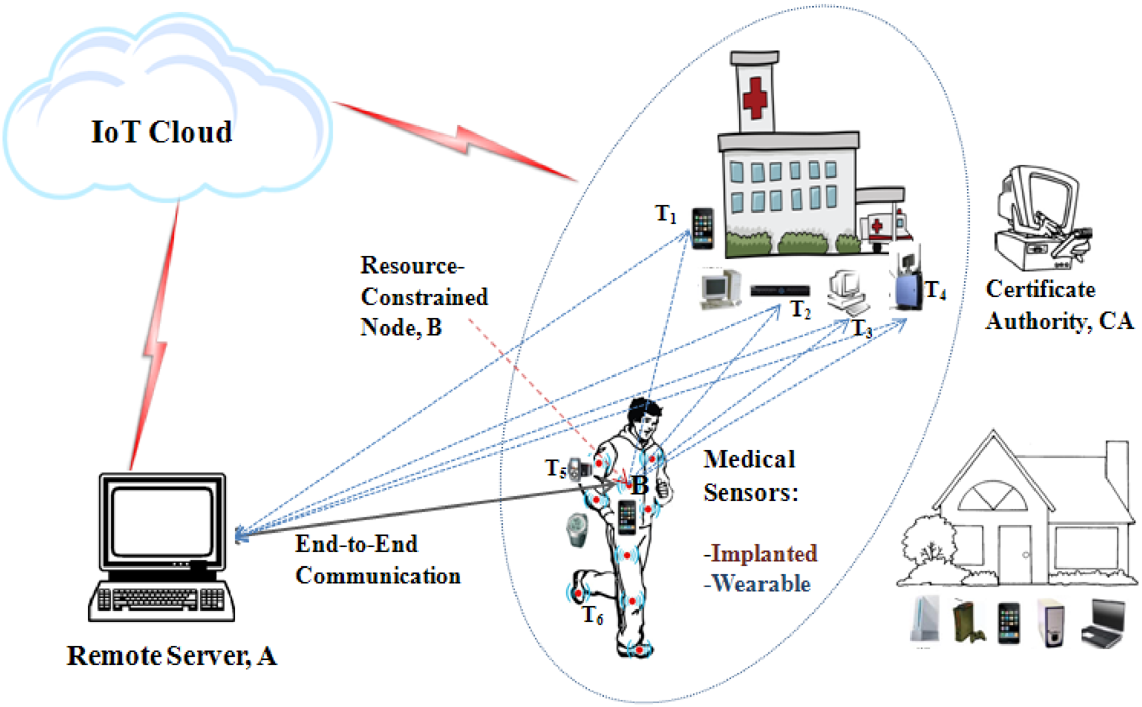


Figure 4-1 Network Model Scenario

4.2.2 Network Model

The network model consists of IoT infrastructure with heterogeneous sensor nodes of different capabilities in terms of computing power and energy resources. In the healthcare monitoring scenario, the network consists of medical sensors implanted and worn on the body of the patient. There might be other sensors present in that vicinity such as surveillance sensors or for climate monitoring (temperature and humidity). Among all this heterogeneous equipment (computers, smart-phones, iPads and sensor devices, surveillance sensors) operating in the hospital or home environment, the implanted sensors on human body are highly resource-constrained nodes and might be located in inaccessible places inside the body (i.e., replacing batteries is impossible, needs surgery). Therefore, preserving their energy resources becomes critically important with the requirement to have an End-to-End secure communication to protect their data. These sensors and devices are classified into three categories as follows:

- Highly resource constrained nodes that are unable to perform asymmetric operations required by the key exchange such as implanted sensors on the human body.
- Less resource constrained nodes that can perform asymmetric operations required by the key exchange but want to preserve their energy for long lasting such as wearable or on-body sensors.
- Devices/sensor nodes with no constrains on available resources (energy, computing power, or storage capabilities) such as remote servers, workstations, laptops, etc.

The network scenario is illustrated in Figure 4-1. The remote server A wants to communicate with highly resource-constrained node B, having no prior shared key or secured connection with it. This communication for data transmission may happen as a push method as well by the implanted sensor. As the sensor, node B doesn’t have any pre-shared secret key with the remote server, so an End-to-End key establishment protocol is required to make a secure channel for further communications. Cryptographic functions required, such as public key infrastructure for the authentication and End-to-End secure communication, are too heavy for the scarce resources of the implanted sensor node. Hence, resource-constrained sensor B assigns these expensive cryptographic functions to its neighboring nodes or devices for assistance (wearable sensors, devices on one-hop) for the authentication of remote server A and establishing as secure channel for further communications.

Selection of these assisting nodes in sensor node B’s proximity is made by their trustworthiness, performance, and the resource capabilities. These assisting nodes are required to maintain their trust factor by delivering the assigned task honestly. As these assisting nodes only perform one portion of cryptographic functions, they cannot learn the secret shared key unless they all cooperate among themselves. If any such node is not able to perform its assigned task or to deliver cryptographic share after computations (either to the remote server or back to sensor node B), this means either that particular node is compromised or doesn’t have enough resources available (battery died). In this situation, if the node is compromised, its shared pair-wise key with the sensor node will be revoked and asked to prove its legitimacy for re-keying. The proposed key establishment protocol is considered not to have the disclosure threat, and the system still continues to work in case of any number of neighboring nodes unavailability.

Table 4-1 Table of used notations

Symbol	Explanation
B	Resource-constrained sensor node
ID_R	Unconstrained node (remote server)
T_i	Trusted Neighboring nodes/devices
CA	Certification authority
N_i	Nonce generated by any node X
K_{ir}	Pre-installed Shared pairwise keys between B and neighboring nodes/devices
K_{ni}	Shared keys between remote servers and neighboring nodes/devices through IPsec
C_i	Coefficients of Lagrange Polynomial function
MAC	Message Authentication Code
K_{DHi}	Shares of Diffie-Hellman secret key
K_{DH}	Established session DH key between B and ID_R

4.2.3 Assumptions

- Each sensor node has a pair-wise secret key with its neighboring nodes/trusted devices as K_{1r}, K_{2r}, K_{3r},... K_{nr} after the initialization phase through a process of bootstrapping using a trusted key management server.
- The resource-constrained sensor nodes are able to discover a set of trusted high resource nodes or devices in their neighborhood.
- The remote server also has shared secret keys with these trusted devices/neighboring nodes as K_{1d}, K_{2d}, K_{3d},..., K_{nd}. The security protocol IPSec can be used as both ends are not having any resource limitations.
- The network also contains a local trusted entity that has a shared secret with all the nodes in the network and a public/private keys pair.

4.2.4 Selection of assisting Nodes/Devices

The resource-constrained sensor node selects the neighboring assisting nodes on the basis of trust on them and their abilities to perform an assigned task (their resource capabilities). Our approach requires that the remote server reports back the IDs of assisting nodes that are not able to deliver the message share or perform their assigned cryptographic functions. The reason for any such assisting node not to perform the assigned task is believed to be either it has been compromised, or it hasn't enough available resources. In the case of node compromise, authorization and authentication questions arise at this particular node, and it must be asked to prove its legitimacy by getting a certificate associated with its public key from a certificate authority (CA). The certificate should include dynamic parameters added by CA in order to become part of the network again. After verification of its certificate by a trusted local entity, such as gateway in our network scenario, this same proposed key establishment process might be performed to establish a new pair-wise key.

Performance of the assisting nodes can be calculated by this simple function:

$$\%\mathrm{R} = \frac{\text{Completed Tasks}}{(\text{Completed Tasks}+\text{Uncompleted Tasks})} \times 100 \qquad (1)$$

4.2.5 Diffie–Hellman (DH) protocol

The Diffie–Hellman (DH) protocol [56] is used in our key establishment protocol to securely establish the secret key at two ends. It's a public distribution key scheme used to establish a common key that is known only to the two participants. Security of this key scheme lies on the difficulty of computing discrete algorithms (similar to factorizing) that is very hard, and it also fulfills the perfect forward secrecy property. Diffie-Hellman key establishment protocol is illustrated below in Figure 4-2. It requires both ends to agree first on appropriate prime (*p*) and generator (g) numbers. Then after both the ends choose their respective secret values, XA and XB (random numbers), they calculate their public values to exchange with each other. After that, the Diffie–Hellman shared secret key can be obtained by both ends, for example, by computing (*g XA XB mod p*).

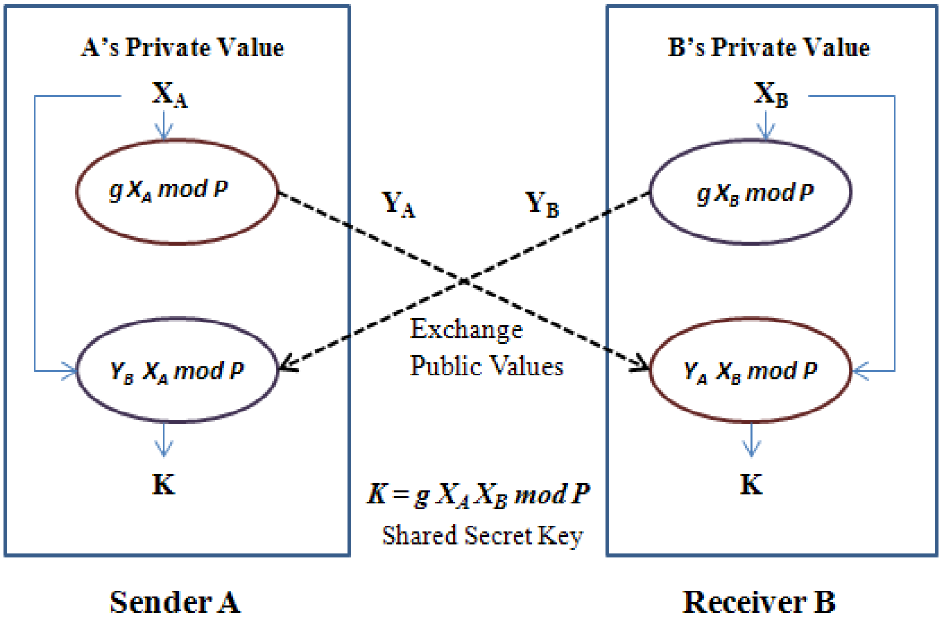


Figure 4-2 Diffie-Hellman Key Establishment Scheme [?]

4.3 Proposed Authentication and Key Establishment Protocol

Offloading the heavy cryptographic functions to neighboring trusted devices is based on guaranteed deliveries of all secret shares to reconstruct the sender's secret shared key at the receiving end.

As illustrated in Figure 4-3, the remote server sends a communication request along with its identity ID_R and a random number (*Nonce*). The implanted resource-constrained sensor node *B* on the body checks the remote server identity and the message freshness by verifying the nonce. *B* computes its private key values shares as *b1, b2, b3 … bn* to assign them to the trusted assisting nodes/devices.

Each trusted assisting node Ti computes its part of initiator's DH public key (g^{bi} *mod p*) upon receiving *bi* and forwards it to the remote server. Resource-constrained sensor node's

Diffie-Hellman public key is computed at the remote server from the product of the shares received from the participating assisting nodes as follows:

$$\prod_{i=1}^{n} g^{bi} \; mod\, p = g^{\sum_{i=1}^{n} bi} \, mod\, p = g^{b} \, mod\, p \qquad (4\text{-}2)$$

Similarly, the remote server shares Ai of its DH public key to each participating neighboring node T_i, and ultimately, the computation of the Diffie-Hellman session key is made by the source, which obtains K_{DH} as:

$$K_{DH} = \prod_{i=1}^{n} K_i = \prod_{i=1}^{n} g^{a.bi} \, mod\, p = g^{a.b} \, mod \qquad (4\text{-}3)$$

From this expression, we can find out that, in this case, if even a single trusted assisting node fails to deliver its assigned share, the key establishment protocol fails. Although all the participating assisting nodes are assumed to be honest and reliable, delivery cannot be guaranteed because of these assisting nodes/devices compromise, misbehave, or other security issues. A re-transmission operation, optionally preceded by a new assisting node assignment, may have to take place with an additional latency. To overcome this issue, (n, k) threshold method as illustrated in [57, 19] is used as if any of the assisting nodes/devices are not able to deliver its assigned share, and the receiver will still be able to compute the shared secret key based on the shares received from the remaining assisting nodes using Lagrange Polynomial interpolation.

The Lagrange interpolating polynomial [56] is the polynomial *P(x)* of degree $\leq (n-1)$ that passes through the *n* points $\big(x_1,\; y_1 = f(x_1)\big), \big(x_2,\; y_2 = f(x_2)\big)$, … $\big(x_n,\; y_n = f(x_n)\big)$ and is given by

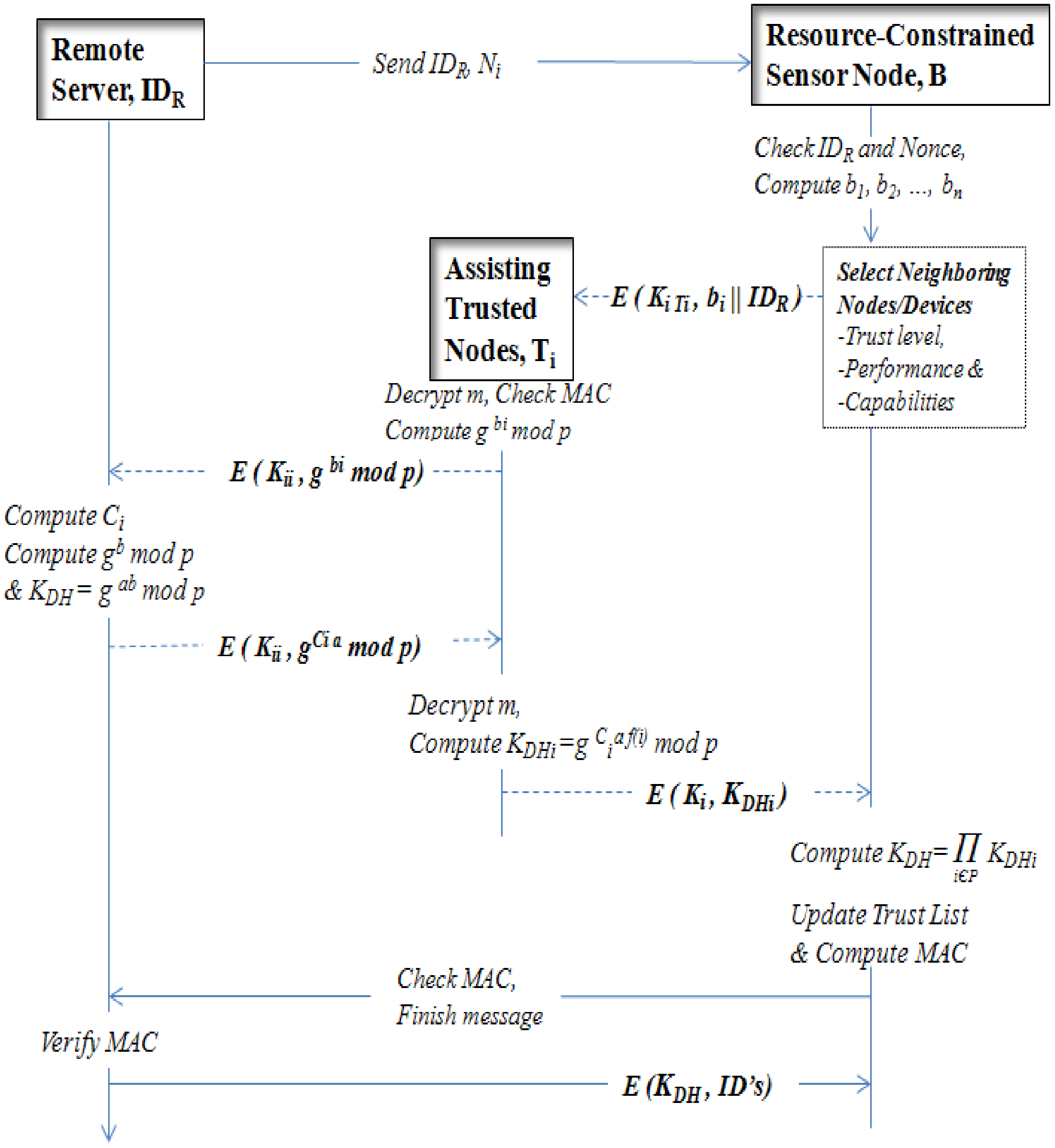

Remote Server, ID_R
Send ID_R, N_i
Resource-Constrained Sensor Node, B
Check ID_R and Nonce,
Compute b_1, b_2, ..., b_n
Assisting Trusted Nodes, T_i
$E\,(K_{i\,Ti},\, b_i \,||\, ID_R\,)$
Select Neighboring Nodes/Devices
-Trust level,
-Performance &
-Capabilities
Decrypt m, Check MAC
Compute $g^{\,bi}$ mod p
$E\,(K_{ii},\, g^{\,bi}$ mod p)
Compute C_i
Compute g^b mod p
& $K_{DH} = g^{\,ab}$ mod p
$E\,(K_{ii},\, g^{Ci\,a}$ mod p)
Decrypt m,
Compute $K_{DHi} = g^{\,C_i a f(i)}$ mod p
$E\,(K_i,\, K_{DHi})$
Compute $K_{DH} = \prod_{i\in P} K_{DHi}$
Update Trust List
& Compute MAC
Check MAC,
Finish message
Verify MAC
$E\,(K_{DH},\, ID\text{'s})$

Figure 4-3 Message Flow of Key Establishment Protocol

$$P(x\,) = \sum_{j=1}^{n} p_j(x)$$

Where $P_j(x) = \; y_j \; \prod_{k=1,k\neq j}^{n} \frac{x-x_k}{x_j-x_k}$

Another issue arises by using this *(n, k)* scheme; as *k* number of shares are sufficient to reconstitute the secret key, what if more than k number of neighboring nodes or devices are compromised in *n*? There is a potential risk of revealing the secret key during the key establishment protocol in the case of compromised $(n - k) \geq k$. This kind of security threat is avoided by inquiring the receiving end to report back the identities of cooperating nodes that are successful to deliver their respective secret shares. This feedback not only helps to identify the trusted assisting nodes and their performances but also to figure out the compromised or greedy nodes. These compromised nodes can therefore be forced for re-keying their pair-wise secret key with the resource-constrained sensor node after their secret keys are revoked. Hence, the proposed key establishment protocol in a way also helps to identify network intrusion. This *(n, k)* threshold scheme for *k* polynomial shares to be sufficient to reconstitute the sender's DH public key through the Lagrange Polynomial interpolation was first used by Shamir [56] and has been implemented by many, such as in [19].

Given a polynomial function *f* of degree $k – 1$ is represented as $f(x) = q_0 + q_1 x + . . . + q_{k-1} x^{k-1}$ where $q_1, q_2, . . . , q_{k-1}$ are random, uniform, and independent coefficients and $b = q_0$.

According to Lagrange formula, the polynomial function f can be derived for this set-up as follows:

$$f(x) = \sum_{j=1}^{k} f_j(x) \tag{4-4}$$

Where $f_j(x) = y_j \prod_{l=1, l \neq j}^{k} \frac{x-l}{j-l}$ The shares of the private exponent *b* are named as b_is and $b_i = f(i)$.

After splitting the private key into its components, the resource-constrained sensor node selects *n* number of neighboring nodes or devices for assistance based on their trust and resource capabilities. This selection is important as *B* (resource-constrained sensor node) assigns these assisting nodes *n* values *f(1), f(2), . . . , f(n)* of polynomial function *f* where *n* > *k* and $b = f(0)$, and transfers each *f(j)* value to the corresponding assisting node after encryption where *j = 1 to n.*

This message consists the encrypted *f(j)* value and received ID_R using a pre-shared key. After receiving this message, each device decrypts it to obtain the identity of the remote server to establish a secure connection and session *keys* $(K_{i1}, . . . , K_{in})$ with the remote server using their secure link protocol IPSec. The legitimacy of the remote server is also verified during this process as well.

After this phase, each assisting node calculates its part of *B*'s Diffie-Hellman public key g^{bj} *mod p* = $g^{f(j)}$ *mod p*, encrypts the values again with K_{j1} keys, and forwards them to the initiator. When the initiator receives *k* number of values from these assisting nodes, it starts computing the C_j coefficients according to the following relation:

$$C_j = \prod_{j\epsilon P, l\neq j} \frac{-l}{j-l} \tag{4-5}$$

Then the initiator reconstructs the responder's Diffie-Hellman public key using the Lagrange formula and the C_j values calculated above. Accordingly, the recipient (i.e., initiator) has to use only *k* successful deliveries out of *n* total messages for the consistent recovery of the responder's DH public key.

$$\prod_{j\epsilon P}(g^{f(j)})^{Cj}\ mod\ p = g^{\sum_{j\epsilon P} f(j)*C_j} = g^{f(0)} mod\ p = g^b\ mod\ p \tag{4-6}$$

Once the responder's Diffie-Hellman public key is calculated, the initiator derives Diffie-Hellman key $K_{DH} = (g^b mod\ p)^a$.

Now the initiator encrypts the messages $g^{C_j a} mod\ p$ by using K_{ij} and transfers them to the assisting nodes $T_{ij}s$ where$j\epsilon \{1, 2, \dots, n\}$. These assisting nodes decrypt their corresponding messages, calculate the Diffie-Hellman key's share, and send them to the resource-constrained sensor node. After the resource-constrained sensor node receives k number of messages from these assisting nodes, it decrypts these receiving messages to reconstitute the Diffie-Hellman key as given below. The resource-constrained sensor node also keeps track of the assisting nodes that are successful in delivering their message share to update the trustworthy neighboring nodes and to delete the pair-wise common keys with the compromised assisting nodes so that such neighboring nodes can be forced to initiate the re-keying process after their key revocation.

$$K_{DH} = \prod_{j \epsilon P} K_{DHj} = \prod_{j \epsilon P} \left(g^{aC_j} \, mod \, p\right)^{f(j)}$$

$$= g^{\sum_{j \epsilon P} f(j) * C_j} \, mod \, p \;=\; g^{ba} mod \, p \qquad (4\text{-}7)$$

After this Diffie-Hellman's key is computed at the resource-constrained sensor node, it computes the message authentication code (MAC) using this key and transfers it to the remote server to complete the handshake process. Once the remote server sends back the ID's of the assisting nodes, who were successful to deliver their messages shares for the computation of sensor node's DH public key and the confirmation that it has successfully derived the key $K_{DH,}$ the communication channel has been established between the remote server and the resource-constrained sensor node. By keeping the ID's of the neighboring nodes and their corresponding pair-wise keys, the protocol becomes more reliable and helps to identify the compromised nodes. The process of event–driven re-keying mechanism is initiated for such neighboring nodes with having characteristics of messages delivery failure, misbehaving, or unreliability.

4.4 Security and Performance Analysis

4.4.1 Security Analysis: Key Establishment Properties

Security and privacy of the key establishment protocol by offloading heavy cryptographic primitives in such a cooperative way mainly depends on the trust of these neighboring nodes/devices. Pre-shared keys are installed during the bootstrapping phase among these neighboring nodes, and they are considered to be trustworthy initially. As *k* number of devices are at least required to deliver their data to re-constitute the shared secret key in *(n, k)* threshold scheme, unless there are not more than k number of neighboring nodes malfunction due to a lack of energy or being compromised, the key establishment protocol is safe. Moreover, our protocol is able to identify the compromised participating nodes so a mechanism can be implemented to revoke their pair-wise symmetric keys and to initiate the process of re-keying for them. The requirement for security and privacy primitive fulfillment is verified by some major attacks and vulnerabilities in the key establishment protocol as follows:

4.4.1.1 Denial of Service (DoS) and Confidentiality

Implementation of this key establishment protocol avoids the possibility of Denial of Service (DoS) attacks from any compromised or fake nodes. In DoS attack, a malicious node tries to interrupt the key establishment protocol by sending redundant messages to the sensor, but communication here in this protocol is through trusted devices; moreover, the remote server is also authenticated by the trusted nodes in the network to establish a secure channel. The protocol also provides confidentiality for the exchanged data between different entities involved. The shared keys between the participating trusted nodes and the remote server are established through well-known IPsec as devices at both ends have enough resources. IPsec is a protocol suite that uses cryptographic security services such as authentication and encryption for each IP packet in a communication session over network.

4.4.1.2 Authentication and Integrity

Authentication and integrity in this key establishment protocol are achieved by the use of MACs to make sure that the communication channel is secure. Moreover, random choice on parameters *(a, b)* and the use of nonces (such as time stamps, random numbers) to ensure the freshness of messages make the re-play attacks impossible.

4.4.1.3 Overheads and Resilience

Resource-constrained senor nodes are not involved in heavy asymmetric encryption primitives, so overheads of our key establishment are very low, making it efficient and secure. With the advancements in flash memory technology, resource-constrained sensor nodes have enough capacity to store the trusted nodes along with their shared keys to avoid the threat of storage overflow. Similarly, the resilience of our protocol is high as secret share is divided into *n* number of parts and for an attacker to reveal the shared secret key, it has to get access to at least *k* number of nodes that is almost impossible due to the trust mechanism implied in the protocol.

4.4.1.4 Scalability and Interoperability

The proposed protocol is highly scalable and is developed in the context of the Internet of Things (IoT), where scalability and interoperability are the main features. The network model completely allows the integration of new sensors into the existing system by going through the initialization phase to have a shared pair-wise key with the neighboring nodes. After a successful initialization, the newly integrated sensor is able to establish an End-to-End secure channel with any remote entity/server.

4.4.2 Security Analysis: Resistivity against MITM and Sybil Attacks

Other than splitting the secret key into *n* parts before passing to trusted devices, since all communication from resource-constrained sensor to trusted devices and from trusted devices to remote server is encrypted and authenticated to ensure the confidentiality, man-in-the-middle (MITM) and eavesdropping attacks cannot determine any information on the content of the messages.

Sybil attacks are where a node claims multiple fake identities that could be used by an intruder to send false information or a participating neighboring node could fake as being multiple neighboring nodes to increase its chances to retrieve more shares of the secret key. This protocol handles the Sybil attacks by keeping a list of identities of trusted neighboring nodes and through the message authentication. Therefore, the participating nodes are not able to use multiple identities with the same secret key.

Moreover, ephemeral Diffie-Hellman primitives are implied in this protocol that has forward secrecy feature, i.e. Short-term session keys cannot be derived from the long-term asymmetric keys.

4.4.3 Energy Consumptions

For the energy consumption evaluation, we have focused only on the resource-constrained sensor nodes in this key establishment protocol. Although our protocol introduces little extra communication overheads due to the reception of trusted devices identities by the resource-constrained sensor node, but these little extra communication costs provide us a secure and reliable key establishment protocol in an efficient way. The threat of revealing the secret key by the compromised neighboring devices is removed, and QoS is improved as well.

According to the expression given in equation 4, the resource-constrained node only spends *(n-1)* modular multiplication operations instead of two modular exponentiation operations, with exponents of considerable length that take too many resources and too much energy consumption.

Table 4-2 Computational energy costs for cryptographic operations of key establishment for resource-constrained sensor node

CH	**Cryptographic Operations**	**Energy Costs**
Cooperative Threshold Approach	*n * (k − 1) * compute_mult f(i) + compute add f(i) + n *(encrypt f(i) + compute MAC) +* *k * (verify MAC + decrypt* K_i *) + compute* K_i *+ K * (compute MAC + verify MAC)*	5 * 2 * (0.09µJ + 0.05µJ) + 5 * (2.47µJ + 10.46µJ) + 3 * (23.02µJ + 19.48µJ) + 290µJ + 3 * (2.1µJ + 2.1µJ) = **496.15µJ**
HIP-DEX	ECC Point Multiplication	**17 mJ**
HIP-BEX	Compute K_{DH}	**104.73 mJ**

Cryptographic energy costs are evaluated using Crypto++ library and the number of participating devices is set as five, whereas at least three participating devices shares are required to be delivered at remote server to reconstruct the shared secret in our performance evaluation. AES-128 algorithm is selected for encryption and decryption and applied on *TelosB* platform. The total energy costs of a specific operation for a sensor (E_{TelosB}, in Joules) can be calculated by multiplying the energy consumption per CPU cycle with the estimated number of CPU cycles (C_{TelosB}):

$$E_{TelosB} = \frac{U\,.\,I}{N}\,.\,C \qquad (4\text{-}8)$$

where *U, I* and *N* are the voltage, intensity, and frequency of the TelosB sensor node.

Here, the amount of energy consumed by the resource-constrained sensor by using this cooperative offloading mechanism comes as *496.15 µJ,* calculated through the similar method as illustrated in [19]. Communication costs are not yet added and are obtained by calculation of the costs by transmission, reception, and listening processes. The proposed key establishment protocol includes a transmission of *80 bytes* and a reception of *48 bytes*, corresponding with energy consumptions of *236 µJ* and *648 µJ* respectively. Therefore, the total energy consumptions turn out to be *1.380mJ* in our proposed protocol. Comparing this energy consumption result to energy consumptions required by other famous key establishment protocols such as HIP-DEX and HIP-BEX, requiring *13.694mJ* and *237.948mJ*, respectively, it seems a significant savings in terms of energy (battery) of the resource-constrained resources of the sensors implanted in the body. The main reason for this huge difference is the use of digital signatures in both these HIP-DEX and HIP-BEX protocols that in our case has been transferred to the participating nodes to perform it on behalf of the resource-constrained node by exploiting the heterogeneity of the sensors/devices in the context of the Internet of Things.

5.1 Summary

In this CHAPTER, we propose a cooperative key establishment protocol to create a secure End-to-End connection for the resource-constrained sensor nodes with any remote server or entity. The protocol is based on offloading heavy cryptographic primitives to the neighboring assisting nodes based on their capabilities and the trust on them by exploiting the heterogeneity of healthcare sensors in the context of the Internet of Things (IoT). Security analysis and performance evaluations prove a considerable security improvement and also the resilience of protocol against well-known attacks and security vulnerabilities. However, there are still more attacks that may threaten this proposed key establishment protocol. Most of these attacks might be the improved version of the mentioned attacks or a combination of them. Nevertheless, for the sake of having secure communications, it is required to investigate the security of the protocol against any new attacks in the future if there are any. Moreover, the proposed protocol is lightweight in terms of energy consumption for resource-constrained sensor nodes and can be employed to other IoT applications or can be integrated with many widely adopted security protocols in IoT environments involving devices with constrained resources. The performance and QoS of the protocol can be further enhanced by finding the optimized number of neighboring nodes to assist the resource-constrained sensor node by running more simulations. We also intend to implement the proposed protocol on actual hardware/sensors and ultimately extend this protocol to body area network where sensors are accessed through a centralized gateway in the context of the Internet of Things.

CHAPTER 6: Improved End-to-End Key Establishment Protocol

In CHAPTER 4, we have proposed a cooperative key establishment protocol to create a secure End-to-End connection for the resource-constrained sensor nodes with any remote server or entity. The protocol is based on offloading heavy cryptographic primitives to the neighboring assisting nodes based on their capabilities and the trust on them by exploiting the heterogeneity of healthcare sensors in the context of the Internet of Things (IoT). Security analysis and performance evaluations prove a considerable security improvement and also the resilience of protocol against well-known attacks and security vulnerabilities. However, there are still more attacks that may threaten this proposed key establishment protocol, and the criteria defined to trust the assisting nodes is not well-defined. Therefore, in this chapter, we will focus on the selection criteria for the assisting nodes based on their past performance and available resources and have added another factor to trust them to know whether they are installed on the same body as the resource-constrained wireless sensors are. All these assisting sensors/devices are accompanied with accelerometers in order to get their data for finding correlation. This data correlation is used to know whether these sensors/devices are installed on the same body to select only trustworthy sensors/devices. In other words, this trust verification is ensured by finding the data correlations from the data of embedded accelerometers in the smartphone acting as gateway and the sensors installed on the body. Security analysis and performance evaluation results show that the proposed protocol is secure and is energy efficient.

6.1 Why Accelerometers for Wireless Sensors?

Smart wearable sensors and devices are becoming popular and are an important part of IoT. These wearable devices are being employed for numerous healthcare applications, ambient assisted living, and sports and fitness applications. Usually a person has more than one device worn on his body. For instance, a person can have a Fit Bit, smart watch to collect physiological data, sensors to take the blood pressure, pulse oximeter, pedometer, and other implanted sensors. Typically, these devices or sensors are highly specialized in their working nature and have varying computational and energy resources. Several issues related to security and privacy of a user's data come up as to how the gateway (smart phone) authenticates these sensors. How are these sensors paired with gateway? How does one balance the user's privacy and usability? Encryption employed cannot be the same for all these sensors with varying available resources [57]. From the end-user's perspective, it is not possible to easily modify these smart devices; security measures must be pre-embedded into the system. The integration of sensors in the internet must ensure the interoperability, transparency, and flexibility. However, sensor nodes inherently have constrained resources with regards to the processing power, memory, communication bandwidth, and energy, especially in healthcare and well-being applications in IoT. Small batteries are typically the main energy sources for such sensors with the limitation to operate for long durations making efficient energy consumption a critical factor besides security issues.

In many practical applications, the gateway needs to send periodic control messages, notifications, and sensitive confidential data to all the wearable devices where a common secret key is required to encrypt the broadcast messages. Symmetric key cryptography such as AES provides fast and lightweight encryption on such smart devices, and their integrated hardware supports it as well. However, as the number of connected devices becomes high, exchanging symmetric keys is infeasible, and therefore, to have an efficient scalable key establishment protocol becomes critical. Another approach is using asymmetric cryptography, but it requires high computational costs, the main concern for resource-constrained sensors. Therefore, conventional security primitives cannot be applied due to the heterogeneous nature, low resources, and the system architecture of IoT-based healthcare systems. Any unauthorized use of a patient's data or privacy concerns may restrict people to utilize IoT-based healthcare applications. Peer authentication and End-to-End data protection are crucial requirements to prevent eavesdropping on sensitive data and malicious triggering of harmful actuating tasks [3]. IoT sensor nodes can be expected to act as a client and as a server contrary to wireless sensor networks, so secure E2E communication is required. It means only the two participants involved at both ends should have access to the agreed secret key shared between them. Sensors worn or implanted on the body are supposed to have low (energy, computation) resources, and they are not meant to perform complex asymmetric cryptographic operations. Ideally, the data coming from wearable or implanted sensors physically installed on the body needs to be authenticated. To ensure this, we proposed to include an accelerometer sensor accompanying the wearable devices. Modern accelerometers are cheap and small enough in size, making them an economic addition; also, they are not supposed to consume much power.

In this chapter, we describe our improved authentication and key establishment protocol to offload the heavy cryptographic computational load to less resource-constrained sensor nodes in a cooperative way and only to the trusted sensors instead of using any random neighboring sensors. We proposed to involve only the trusted neighboring sensors in order to perform these asymmetric cryptographic tasks. Accelerometers data accompanying sensors is correlated with smart phone's accelerometer data to answer the simple question: are these nodes attached to the same body? Their relevant on-body movements and activities are detected by using accelerometer sensors to find the data correlations. This work can further be enhanced to recognize different activities of users to help them in ambient assisted living and is important due to its adaptability with other existing E2E secure communication solutions and IoT applications. The proposed protocol is lightweight in terms of consuming resources, and the selection of trusted neighboring sensors makes it more secure.

6.2 Accelerometers and Acceleration Measurement

As described earlier, the heavy computational tasks of asymmetric key agreement and distribution are computed by offloading them to the neighboring nodes with enough resources. Selection of these neighboring nodes is made by verifying whether they are installed or worn on the body. These sensor nodes are assumed to have an extra tri-axial accelerometer accompanying, the same type as smartphone (gateway) has integrated to compare data directly for finding correlations. Since accelerometers are tiny, cheap, and require little energy to operate, this is a reasonable assumption and feasible to implement. As an example, the freescale MMA845xQ line of accelerometers costs about a couple of dollars and consumes "1.8 micro amps in standby mode and as low as 6 micro amps in active mode" [60, 57], whereas accelerometers employed in our work are LIS3DSH (developed by Mouser Electronics), used in Broadcom WICED Sense Bluetooth smart sensor development kit, Figure 5-1. They have a similar price as the freescale MMA845xQ line of accelerometers. LIS3DSH accelerometer is an ultra-low-power high performance three-axis linear accelerometer belonging to the "nano" family with an embedded state machine that can be programmed to implement autonomous applications. It has dynamically selectable full scales of ±2g/±4g/±6g/±8g/±16g and is capable of measuring accelerations with output data rates from 3.125 Hz to 1.6 kHz [61].

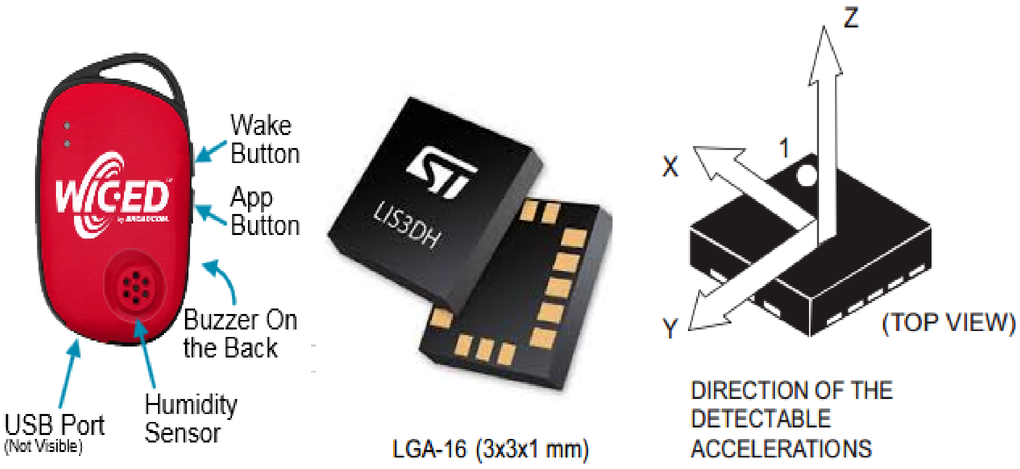


Figure 5-1 Broadcom WICED Sensor and LIS3DSH Accelerometer

Accelerometers detect force acting in the opposite direction to the displacement vector, and measured acceleration needs to be corrected for the gravitational influence. The raw data from smartphone and sensors' accelerometers is converted to readable data through calibration. Accelerometer's raw data is given in local device coordinates; axes orientations change if the device orientation is misplaced. Like if the device is turned on its side, the Z-axis no longer points upwards; instead, it is also rotated, Figure 5-1(c). While the acceleration vectors can be used to determine the roll and pitch angles, these values may not be appropriate for real-time calculations. Therefore, the assumption is made that orientation of the smartphone is not changed with respect to position of the sensor installed on the body. Hence, the transformations are not required to compare data for correlations. Also, the acceleration measurements are assumed to be normalized; the device measured gravity with a unit value.

6.3 Authentication and Correlation of data

As our approach depends on recognizable acceleration events, the algorithm performs authentication if the user is walking. Also, data for a very short duration may lead to incorrect results (false positives and false negatives). The smartphone working as gateway records data from sensors' accelerometers to initiate the authentication process of the neighboring sensor. In the case that any sensor is removed from the body, it loses the status of authenticated trusted sensor. This trust authentication procedure is described below from the smartphone perceptive.

The magnitude of all three axes for accelerometers samples is used for data correlation as accelerometers can be mounted in different orientations.

$$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \tag{5-1}$$

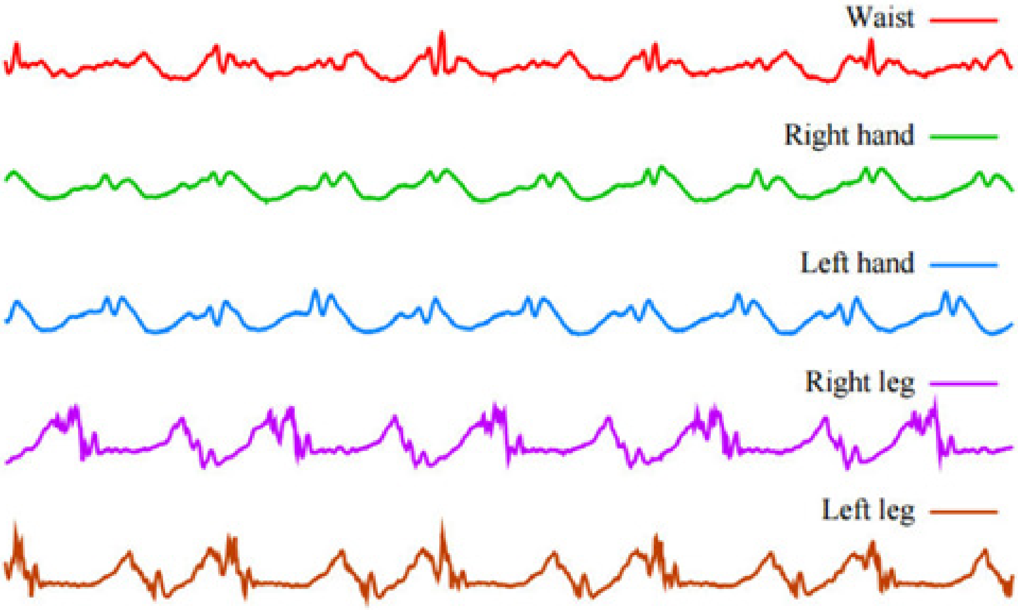


Figure 5-2 Accelerometers magnitude data for each position on the body

We also get the rate of change of speed over time for that specific sensor. Coherence is a measure of how well two signals correlate in the frequency domain. More precisely, it is the cross-spectral density of two signals divided by the auto spectral density of each individual signal [62]. Hence, for two sets of feature matrices *A = (F1, F2, . . .)* and *B = (F1, F2, . . .)* with entries Fj, we can find out how well A and B are correlated. A and B represent the feature matrices extracted from the accelerometer data of the smartphone and sensors, respectively.

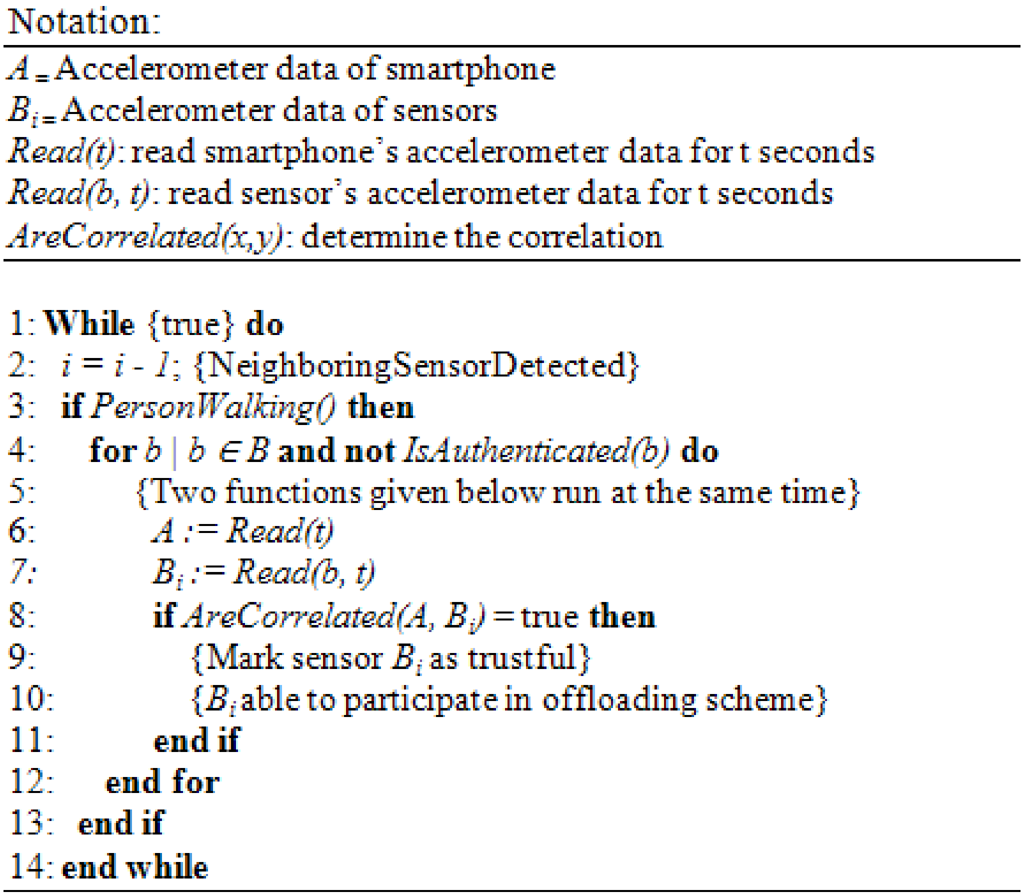
6.4 System Architecture

IoT has solutions for a wide range of applications, and it requires having a secure E2E connection between devices. For example, there are healthcare applications where a remote server may require health-related sensitive data from a highly resource-constrained sensor. The architecture of an IoT-based health monitoring system is illustrated in Figure 5-2. Both the remote server and resource-constrained sensor require having a secure E2E communication link between them, meaning both ends first need to authenticate each other and securely establish a secret key to encrypt the transmitted data.

6.4.1 Key Agreement and Distribution Requirements

For key agreement protocol and authentication method employed within the protocol itself, typically asymmetric keys are used. As symmetric key based protocols require key refresh after a certain time so are not good candidates for key agreement and distribution protocols. Moreover, symmetric key based security protocols inquire for pre-distribution process, a complex setup [19]. When sensors have constrained resources, extraordinary attention is needed to have a secure E2E channel considering the capabilities of the both ends.

Table 5-1 Notations and data correlation pseudo code



Notation:

A = Accelerometer data of smartphone

B_i = Accelerometer data of sensors

Read(t): read smartphone's accelerometer data for t seconds

Read(b, t): read sensor's accelerometer data for t seconds

AreCorrelated(x,y): determine the correlation

```
While {true} do
  $i = i - 1$; {NeighboringSensorDetected}
  if PersonWalking() then
    for $b \mid b \in B$ and not IsAuthenticated(b) do
      {Two functions given below run at the same time}
        $A := Read(t)$
        $B_i := Read(b, t)$
        if AreCorrelated($A$, $B_i$) = true then
          {Mark sensor $B_i$ as trustful}
          {$B_i$ able to participate in offloading scheme}
        end if
    end for
  end if
end while
```

6.4.2 *Network Model*

In the body area network health monitoring scenario illustrated in Figure 5-2, it consists of heterogeneous resources medical sensors accompanying tri-axial accelerometers implanted and worn on the body of person along with a smartphone working as gateway (or base station). Among all this, sensors installed on body and other equipment (computers, smart-phones, iPads and sensor devices, surveillance sensors working in the vicinity hospital or home environment), implanted sensors on the human body are highly resource-constrained nodes and might be located in inaccessible places inside body as well (i.e., replacing batteries is impossible, needs surgery). Therefore, our focus is to preserve their energy resources and is critically important with the condition to have an E2E secure communication to secure their data. These sensors and devices are classified into different categories as follows:

- Highly resource-constrained nodes, unable to perform asymmetric operations such as implanted sensors.
- Less resource constrained sensors, able to perform asymmetric operations required by the key agreement protocol having accelerometers integrated with them, including the smartphone working as a gateway.
- Devices or sensors with no constrains on resources (energy, computing power, or storage capabilities) such as remote servers, workstations, laptops, etc.

If the remote server *A* wants to communicate with highly resource-constrained sensor *B* having no prior shared key or secured connection to get some data, an E2E key distribution protocol is needed to establish a secure channel between them for further communications. *B* delegates its expensive cryptographic tasks required for asymmetric key agreement to its neighboring trusted sensors installed on the body for assistance.

6.4.3 Selection of Assisting Nodes

Selection of assisting trusted sensors in *B*'s proximity is made by finding out whether they are installed on the body. If any such sensor node's accelerometer data is not correlated with the smart phone's accelerometer data, this implies that the specific sensor is not trustful and is involved in assigning any task to contribute in key agreement mechanism. As sensors cannot be installed without the permission and knowledge of person, they are considered trustworthy initially. Moreover, if any such neighboring node is compromised, its shared pair-wise key with the sensor node and gateway is revoked, forcing it to initialize re-keying process. The proposed protocol therefore rules out the disclosure threat and continues working even if a neighboring sensor becomes unavailable.

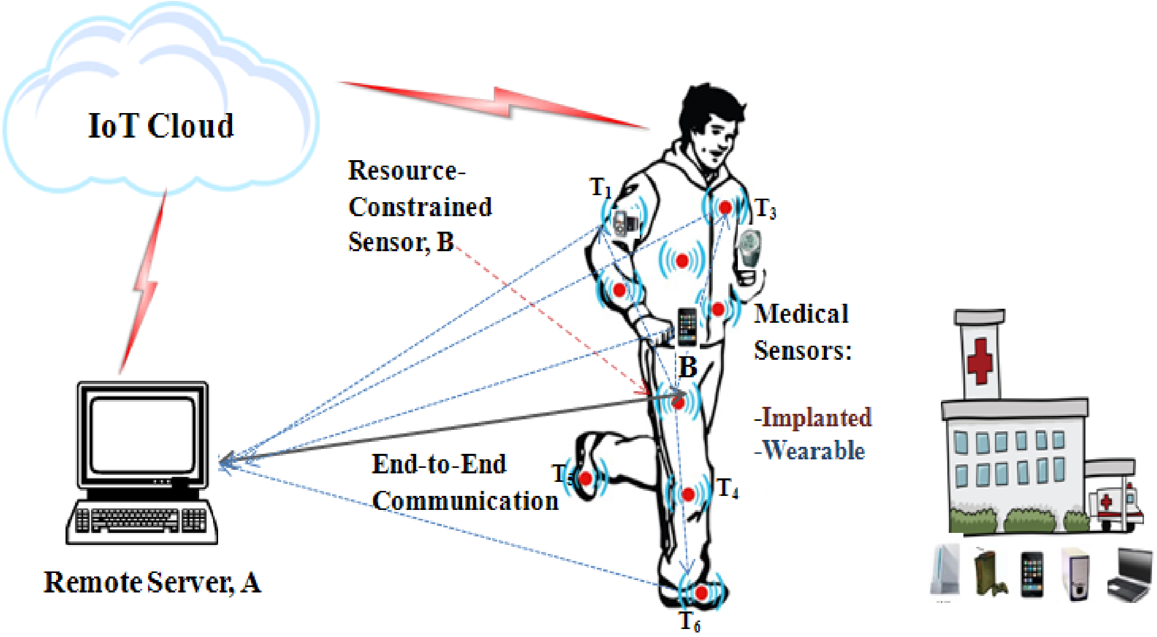


Figure 5-3 Network Model Scenario

6.4.4 Assumptions

- Each sensor node has a pair-wise secret key with its neighboring nodes/trusted devices as K_{1r}, K_{2r}, K_{3r},… K_{nr} after the initialization phase through a process of bootstrapping using a trusted key management server.
- The resource-constrained sensor nodes are available to discover a set of trusted high-resources sensor nodes in their neighborhood by finding the data correlations through accelerometers data (provided by gateway).
- Remote server also has shared secret keys with these trusted devices/neighboring nodes as K_{1d}, K_{2d}, K_{3d}, …, K_{nd}. The security protocol IPSec can be used as both ends are not having any resource limitations.

6.5 Proposed Authentication and Key Agreement/Distribution Protocol

Our approach requires that the gateway reports the *IDs* of neighboring assisting nodes installed on the body by the data correlation process. If a neighboring sensor accelerometer's data fails to correlate with gateway accelerometer's data, it is considered as a compromised sensor. Authentication issues arise at this particular sensor, and it is not involved further in the key agreement and distribution process.

Diffie–Hellman (DH) [56] security protocol is used in our proposed key agreement and distribution protocol. Security of this key scheme lies on the difficulty of computing discrete algorithms (similar to factorizing) that are very hard and also fulfill the perfect forward secrecy property. DH key establishment protocol requires both ends to agree first on appropriate prime (*p*) and generator (*g*) numbers. Then after both the ends choose their respective secret values, X_A and X_B (random numbers), they calculate their public values to exchange with each other. After that DH shared secret key is computed by both ends by

Table 5-2 Table of selected notation

Symbol	Explanation
ID_R	Unconstrained node (remote server)
T_i	Trusted Neighboring sensors/devices
N_i	Nonce generated by any node *X*
K_{ir}	Shared pairwise keys between *B* and neighboring sensors
K_{ni}	Shared keys between remote servers and neighboring sensors/devices through IPsec
MAC	Message Authentication Code
K_{DHi}	Shares of Diffie-Hellman secret key
K_{DH}	Established session DH key between *B* and ID_R

computing (*g* X_A $\mathrm{X_B}$ *mod p)*. Getting assistance from trusted neighboring sensors relies on guaranteed deliveries of all secret shares so as to reconstruct the sender's secret shared key at the receiving end. As described in Figure 5-3, remote server generates a data request along with its identity ID_R and a random number (Nonce). The implanted resource-constrained sensor *B* verifies ID_R and the message freshness by checking nonce. *B* divides its secret key into b_1, b_2, b_3 ... b_n shares to assign them to the trusted assisting sensors. Each trusted assisting sensor T_i computes its part of initiator's DH public key (g^{bi} *mod p)* upon receiving b_i and forwards it to the remote server. *B'*s DH public key is computed at the remote server from the product of the shares received from participating assisting nodes as:

$$\prod_{i=1}^{n} g^{bi} \ mod\, p = g^{\sum_{i=1}^{n} bi} \ mod\, p = g^{b} \ mod\, p \qquad (5\text{-}2)$$

Similarly, the remote server shares A_i of its DH public key to each participating neighboring node T_i, and ultimately, the computation of DH session key is computed by the source, which obtains K_{DH} as:

$$K_{DH} = \prod_{i=1}^{n} K_i = \prod_{i=1}^{n} g^{a.bi} \, mod\, p$$

$$= g^{a.b} \, mod\, p \qquad (5\text{-}3)$$

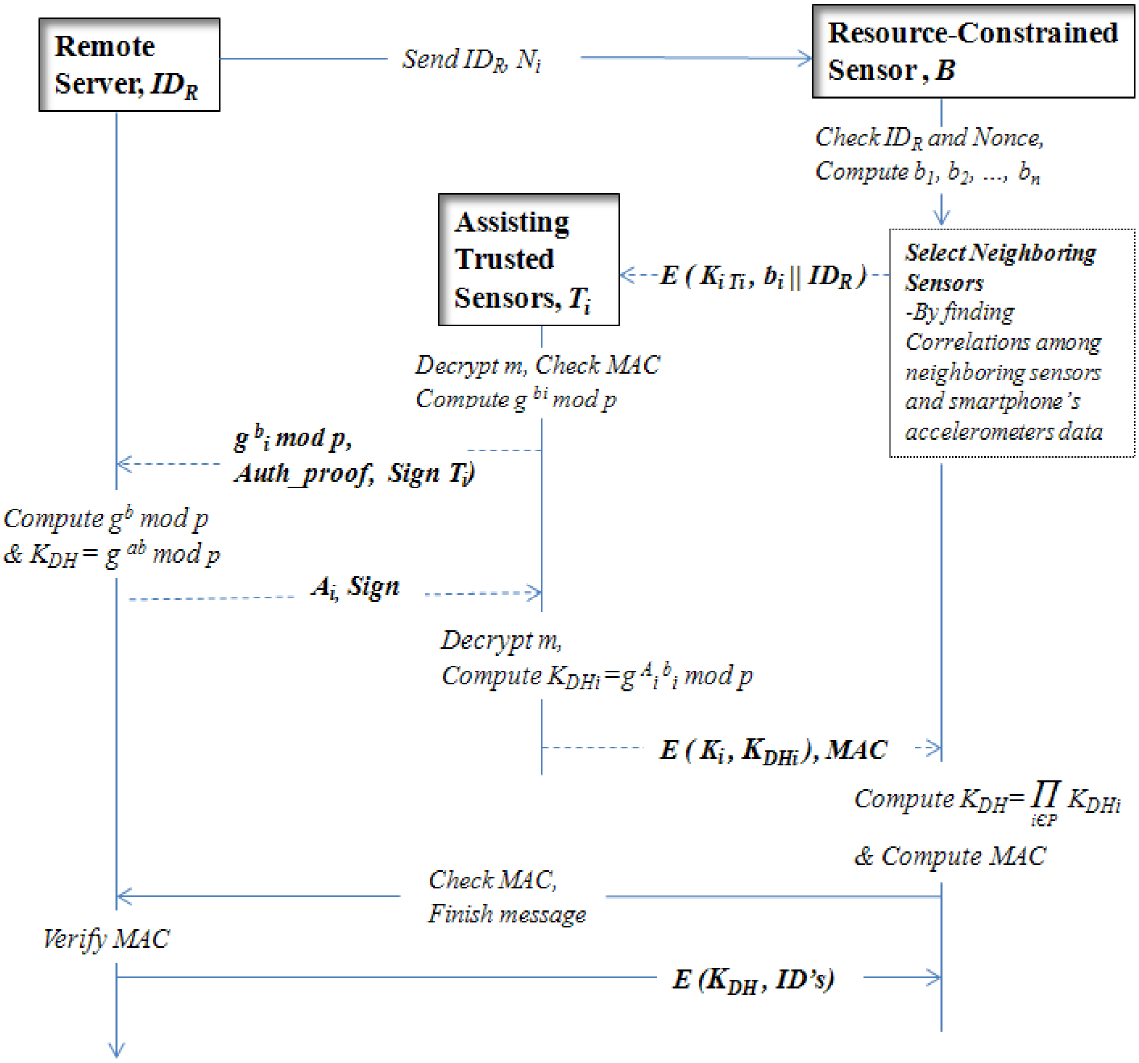

Remote Server, ID_R
Send ID_R, N_i
Resource-Constrained Sensor , B
Check ID_R and Nonce,
Compute b_1, b_2, …, b_n
Assisting Trusted Sensors, T_i
$E (K_{i\,Ti} , b_i \,||\, ID_R)$
Select Neighboring Sensors
-By finding Correlations among neighboring sensors and smartphone's accelerometers data
Decrypt m, Check MAC
Compute g^{bi} mod p
$g^{b}{}_{i}$ mod p,
Auth_proof, Sign T_i)
Compute g^b mod p
& $K_{DH} = g^{ab}$ mod p
A_i, Sign
Decrypt m,
Compute $K_{DHi} = g^{A_i b_i}$ mod p
$E (K_i , K_{DHi})$, MAC
Compute $K_{DH} = \prod_{i \in P} K_{DHi}$
& Compute MAC
Check MAC,
Finish message
Verify MAC
$E (K_{DH}$, ID's)

Figure 5-4 Message flow of key agreement and distribution protocol

The secret key is not computed at the receiving end if even a single trusted assisting sensor fails to deliver its assigned share. Delivery for shared part is assumed to be guaranteed, meaning all the participating sensors are trustful, capable, and reliable. A re-transmission operation, optionally preceded by a new assisting sensors assignment, may have to take place with an additional latency. After *B* receives messages back from these assisting neighboring sensors, it decrypts these receiving messages to reconstitute the DH public key as:

$$K_i = A_i^{bi} = \ (g^a \ bmod \ p)^{bi}$$

$$= g^{a.bi} \ mod \ p \qquad (5\text{-}4)$$

Hence, the computation of DH secret key is made by the responder as K_{DH}. It also computes the MAC using this key and transfers to the remote server to complete the handshake process. Once the remote server confirms its successful derivation of the key $K_{DH,}$ the communication channel is established between the remote server and the resource-constrained sensor.

6.6 Security and Performance Analysis

6.6.1 Security Analysis: Key Establishment Protocols

In this proposed protocol, security and privacy depends mainly on the trust of these neighboring sensors. Pre-shared keys are exchanged to the sensors during the bootstrapping phase shared with each other and the gateway and are considered trustworthy initially. Later on, data correlations among their accelerometers are determined to exchange symmetric secret keys only to trusted neighboring sensors. Moreover, the protocol is able to identify the compromised participating sensors. Some major attacks and vulnerabilities considered are as follows:

6.6.1.1 Denial of Service (DoS) and Confidentiality

In DoS attacks, a malicious node tries to interrupt by sending redundant messages to the responder, as communication in this protocol is through trusted sensors, and installation of sensors is not possible without the approval or notice of the user (patient); protocol implementation leaves out the possibility of DoS attacks from any compromised or fake sensors. Moreover, the remote server is also authenticated by the trusted sensors in the network to create a secure channel to provide confidentiality for the exchanged data between different entities involved.

6.6.1.2 Authentication and Integrity

MACs are used to ensure a secure communication channel and to provide authentication and integrity. Random choice of parameters *(a, b)* and also the use of nonces (such as time stamps, random numbers) ensure the freshness of messages making the re-play attacks impossible.

6.6.1.3 Overheads and Resilience

The resilience of this proposed protocol is high as secret key is divided into *n* parts, making it almost impossible to compromise all *n* neighboring sensors by the attacker due to trust mechanism employed, whereas resource-constrained sensors themselves are not involved in heavy asymmetric encryption tasks to enjoy asymmetric security primitives. Hence, the proposed protocol is secure enough and lightweight for resource-constrained sensor power consumptions.

6.6.1.4 Scalability and Interoperability

Network model scenario considered completely allows adding new sensors through the initialization phase to have a shared pair-wise key with other sensors. Once the initialization process is successful, the newly integrated sensor is able to create an E2E secure channel with any remote server or device, making the protocol highly scalable and interoperable.

6.6.2 Security Analysis: Resistivity against MITM and Sybil Attacks

Other than splitting the secret key into *n* parts before passing to trustful neighboring sensors, all communication from resource-constrained sensor to trusted devices and from trusted devices to remote server is encrypted and authenticated to ensure confidentiality; man-in-the middle (MITM) and eavesdropping attacks cannot determine any information on the content of the messages. Sybil attacks, where a sensor node claims multiple fake identities, could be used by an intruder to send false information, or a participating neighboring sensor could fake as being multiple neighboring sensors to increase its chances to retrieve more shares of the secret key. Such attacks cannot be used against this proposed protocol as the basic requirements for the neighboring sensors to get selected for assistance is through the correlations of the accelerometers data. Participating neighboring sensors are not able to use multiple identities with the same secret key. Moreover, ephemeral DH primitives are utilized in this protocol that includes the forward secrecy feature, making the proposed protocol more secure.

6.6.3 Energy Consumptions

Our main concern in the given network scenario is to conserve the already scarce energy resources of implanted sensors. Resource-constrained sensor '*B'* spends (*n-1)* modular multiplication operations (as shown in equation 3) instead of two modular exponentiation operations, with exponents of considerable length that take too much resources and energy consumption. Accelerometers accompany and consume power of the wearable sensors where energy conservation is not a major concern. The little extra communication overheads introduced due to exchanging *IDs* of trusted sensors are not considerable in getting secure and reliable communications.

Cryptographic energy costs are evaluated using Crypto++ library, keeping the number of participating neighboring sensors 5 in our performance evaluation as in [19]. AES-128 algorithm is selected for encryption and decryption and applied on *TelosB* platform. The total energy costs of a specific operation for a sensor (E_{TelosB}, in Joules) can be calculated by multiplying the energy consumption per CPU cycle with the estimated number of CPU cycles (C_{TelosB}):

$$E_{TelosB} = \frac{U\,.\,I}{N}\,.\,C \tag{5-5}$$

U, I and *N* are the voltage, intensity, and frequency of the *TelosB* sensor. Test programs for individual computational operations were executed on an Intel i5 processor, and the corresponding number of processor cycles for each was retrieved. Some advanced features for test processor such as hyper threading, multi-core, and variable clock speed were disabled.

The number of cycles can be derived from the number of CPU cycles measured on the *i5* (C_{i5}) as:

$$C_{TelosB} = \frac{Register\_Size_{i5}}{Register\_Size_{TesoB}}\,.\,\alpha\,.\,\mathrm{C}_{i5} \tag{5-6}$$

α here is a coefficient showing the richer instruction of the i5 and is set to 2 for our analysis. The total amount of energy consumed by the resource-constrained sensor by using this cooperative offloading mechanism comes as 702.*64µJ*. Communication costs are yet to be added, consisting of the costs by transmission, reception, and listening process. The proposed key agreement and distribution protocol generates *80 bytes* of transmission and *48 bytes* of reception, corresponding with energy consumptions of *236 µJ* and *648 µJ*, respectively, making

Table 5-3 Computational energy costs for cryptographic operations of key establishment protocol for resource-constrained Sensor

	Cryptographic Operations	**Energy Costs**
Basic Approach	*compute_DH*$_I$*+compute_*K$_{\mathrm{DH}}$ + *compute_*K$_{\mathrm{i}}$ + *compute_sign* K$_I$*_encrypt_msg3* + *compute_MAC_*K$_{\mathrm{i}}$ + *verify_MAC_*K$_{\mathrm{i}}$ + Ki*_decrypt_msg4* + *verify_CERT* + *verify_sign*	*58.97 mJ + 104.73 mJ + 16.74 µJ + 24.39 mJ + 205.25 µJ + 142.31 µJ + 138.12 µJ + 200.31 µJ + 2.1 mJ + 1.22 mJ* *=* ***192.11 mJ***
HIP-DEX	ECC Point Multiplication	***17 mJ***
HIP-BEX	Compute K_{DH}	***104.73 mJ***
Cooperative Approach	n*(encrypt_*b$^{\mathrm{i}}$ + *compute_MAC* + *verify_MAC* + *decrypt_*g$^{\mathrm{a.bi}}$ *mod* p*)* + *compute_mult_*g$^{\mathrm{bi.a}}$ + *compute_*K$_{\mathrm{i}}$ + K$_{\mathrm{i}}$*_encrypt_msg3'* + *compute_MAC_*K$_{\mathrm{i}}$ + *verify_MAC_*K$_{\mathrm{i}}$ + K$_{\mathrm{i}}$*_decrypt_msg4'* + n*(compute_MAC* + *verify_MAC) decrypt Ki)* + *compute Ki* + *K * (compute MAC* + *verify MAC)*	*5*(2.47 µJ + 10.46 µJ + 23.02 µJ + 19.78 µJ) + 290 µJ + 16.74 µJ 29.67 µJ 23.02 µJ 18.83 µJ 24.73 µJ 5* (2.1 µJ + 2.1 µJ)* *=* ***702.64 µJ***

the total energy consumed as *1.586mJ*. After comparing it with energy consumptions of other key distribution protocols such as HIP-DEX and HIP-BEX, requiring *13.694mJ* and *237.948mJ*, respectively, there is a significant energy savings for the resource-constrained resources. The difference is due to the use of digital signatures in both these key distribution protocols. We have established secure channel by offloading the heavy cryptographic functions to trustful neighboring sensors by exploiting the heterogeneity of resources in context of IoT.

6.7 Summary

In this work, we proposed a key agreement and authentication protocol to establish a secure E2E channel based on offloading heavy cryptographic functions to the trusted neighboring sensors in the context of IoT. These sensors also accompany tri-axial accelerometers for correlating their data to find out whether sensors are installed on the body. Security analysis and performance evaluations are made to prove that the proposed protocol is secure enough and is lightweight for implanted sensor that has scarce computational and energy resources. Accelerometers were used independently on the body to find the correlations between their data and the accuracy obtained in identifying whether accelerometer is installed on the body is more than 85%. Proposed protocol can be integrated with many widely adopted security protocols in the IoT environment involving devices with constrained resources. Use of accelerometers with security protocol can have many applications such as fall detection for older people in nursing homes or in patient-tracking inside the hospitals where GPS is not available. In the future, we intend to perform experiments with the actual medical sensors accompanying accelerometers installed on the body and ultimately develop a full body area network system for healthcare application in IoT.

CHAPTER 7: Conclusion and Future Works

7.1 Conclusion

A new distributed security paradigm for resource-constrained wireless sensors in the context of the Internet of Things (IoT) is described, including key agreement and authentication protocols to establish a secure E2E channel based on offloading heavy cryptographic functions to the trusted neighboring sensors and also having investigated the feasibility study to implement LEAP+ key management protocol over ZigBee stack. While developing these key establishment protocols, the assisting sensors or devices also accompany tri-axial accelerometers for correlating their data to find out whether sensors are installed on the body. This data correlation is utilized in finding out whether these assisting sensors or devices are installed on the body or not, as we have taken healthcare IoT application as an example where body has either implanted, installed, or wearable heterogeneous wireless sensors in implementing our key establishment protocols. Security analysis and performance evaluations are made to prove that the proposed protocol is secure enough and is lightweight for the implanted sensor that has scarce computational and energy resources. Accelerometers were used independently on the body to find the correlations between their data and the accuracy obtained in identifying whether the accelerometer installed on the body is more than 85%. Proposed protocol can be integrated with many widely adopted security protocols in IoT environment involving devices with constrained resources.

In CHAPTER 2, we have described how these wireless sensors are integrated into the Internet of Things (IoT) and how the security and privacy challenges are different than the legacy internet and the security solution approaches being taken to the data of these resource-constrained sensors in the context of IoT as related works.

In CHAPTER 3, we have described our work for investigating the feasibility of substituting the key management scheme of ZigBee by implementing LEAP+ to enhance the security capabilities when wireless sensors are integrated into IoT as Wireless Sensor Networks (WSNs). LEAP+ is a symmetric distributed key management protocol for sensor networks that is designed to support multi-type keys depending on the type of message being exchanged. LEAP+ is surprisingly well-suited to different types of network topologies, device types, and addressing modes offered by ZigBee stack, resolving the issue of scalability due to ZigBee's key management centralized approach. Our experimental results and performance evaluation parameters are not only valuable to assess the feasibility of LEAP+ scheme on the ZigBee protocol stack, but they also provide the basis for having an effective mechanism to get reasonable scalability within WSNs.

7.2 Future Work

Throughout this research, some ideas have occurred that may expand the scope of our original goals and mitigate restrictions of these proposed key establishment algorithms and protocols. This section provides an overview of possible ideas that could be followed in further work.

- Our experimental results and performance evaluation parameters are not only valuable to assess the feasibility of LEAP+ scheme on the ZigBee protocol stack, but they also provide the basis for having an effective mechanism to get reasonable scalability within WSNs. There is, however, a significant point to be considered; that is, LEAP+ is essentially meant for stationary nodes. Mobility of the nodes within the network is highly significant for mobile wireless sensor networks and needs to be considered for future work. This need inspires the idea to upgrade LEAP+ with mobile capability, keeping in mind that there are a lot of design challenges and potential issues that must be addressed and resolved in order to enable mobility in sensor networks to get enhanced security and reduced performance overheads.

- Proposed protocol can be integrated with many widely adopted security protocols in IoT environment involving devices with constrained resources. Use of accelerometers with security protocol can have many applications, such as fall detection for older people in nursing homes or patients-tracking inside hospitals where GPS is not available. In the future, we intend to perform experiments with the actual medical sensors accompanying accelerometers installed on the body and ultimately develop a full body area network system for healthcare application in IoT.

BIBLIOGRAPHY

[1] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security Version 1.2,” RFC 6347, IETF, 2012.

[2] Christian Dancke Tuen “Security in Internet of Things Systems” Master Thesis Norwegian University of Science and Technology.

[3] D. E Vans, The Internet of Things: How the Next Evolution of the Internet is Changing Everything, Cisco Internet Business Solutions Group (IBSG), 2011.

[4] A. J. Menezes, S. A. Vanstone , P. C. Van Oorschot, Handbook of Applied Cryptography, CRC Press, Inc., Boca Raton, FL, 1996.

[5] Benjamin Kleine, Bethany Lobo, Amanada Levendowski (2015) “Internet of Things: The new frontier for data security and privacy” (Part 1).

[6] “Internet of Things: An overview by Internet Society” https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151014_0.pdf

[7] Muhammad A. Iqbal, Magdy Bayoumi “Wireless Sensors Integration Into Internet of Things and the Security Primitives” The Seventh International Conference on Ubiquitous Computing (Ubic-2016) September 24 ~25, 2016 Dubai,UAE.

[8] Cristina Alcaraz, Pablo Najera, Javier Lopez, Rodrigo Roman, “Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?” University of Malaga, Spain.

[9] Bruce Ndibanje, Hoon-Jae Lee, and Sang-Gon Lee, “Security Analysis and Improvements of Authentication and Access Control in the Internet of Things”.

[10] Emmanouil Vasilomanolakis, Jorg Daubert, Manisha Luthra, Vangelis Gazis, Alex Wiesmaie and Panayotis Kikiras "On the Security and Privacy of Internet of Things Architectures and Systems".

[11] Y. K. Lee, K. Sakiyama, L. Batina, I. Verbauwhede, "Elliptic-Curve-Based Security Processor for RFID," Computers, IEEE Transactions on Volume: 57, Issue: 11, 2008, Page(s): 1514 – 1527.

[12] S. S. Kumar and C. Paar, "Are standards compliant Elliptic Curve Cryptosystems feasible on RFID?", In Proceedings of Workshop on RFID Security, page 19, Graz, Austria, July 2006.

[13] Hossein Shafagh (2013) "Leveraging Public-key-based Authentication for the Internet of Things" Master Thesis, RWTH Aachen University, Germany.

[14] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, M. Rossi. "Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples". In IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'12), San Francisco, CA (June 2012), pp. 1–7.

[15] Sepideh Fouladgar, Bastien Mainaud, Khaled Masmoudi, Hossam Afifi. "Tiny 3-TLS: a trust delegation protocol for wireless sensor networks". In Proceedings of the Third European conference on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS'06), Hamburg, Germany (Nov 2006), pp. 32–42.

[16] Vipul Gupta, Michael Wurm, Yu Zhu, Matthew Millard, Stephen Fung, Nils Gura, Hans Eberle, Sheueling Chang Shantz. (2005) "Sizzle: A standards-based end-to-end security architecture for the embedded Internet. In Pervasive and Mobile Computing".

[17] R. Hummen, H. Shafagh, S. Raza, T. Voigt, and K. Wehrle, (2014) "Delegation based Authentication and Authorization for the IP-based Internet of Things," in IEEE SECON.

[18] A. J. Menezes, S. A. Vanstone, P. C. Van Oorschot, “Handbook of Applied Cryptography”, CRC Press, Inc., Boca Raton, FL, 1996.

[19] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, “Lightweight collaborative key establishment scheme for the Internet of Things” Computer Networks, 2014 vol. 64, pp. 273 – 295.

[20] T. Kivinen, “Minimal IKEv2,” draft-kivinen-ipsecme-ikev2-minimal-01 (WiP), IETF, 2012

[21] R. Moskowitz and R. Hummen, “HIP Diet EXchange (DEX),” draftmoskowitz-hip-dex- 01 (WiP), IETF (2012).

[22] T. Kivinen, “Minimal IKEv2,” draft-kivinen-ipsecme-ikev2-minimal-01 (WiP), IETF (2012).

[23] Y. Saied and A. Olivereau, “D-HIP: A distributed key exchange scheme for HIP-based Internet of Things,” in Proceeding of IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM) 2012, pp.1–7.

[24] Muhammad A Iqbal, Magdy Bayoumi “Secure End-to-End Key Establishment Protocol for Resource-Constrained Healthcare Sensors in the Context of IoT” The 14th Annual IEEE International Conference on High Performance Computing and Simulations (HPCS) 2016, Innsbruck Austria.

[25] P. Porambage, A Braeken, A Gurtov, M Ylianttila and Susanna Spinsante “Secure end-to-end communication for constrained devices in IoT-enabled Ambient Assisted Living systems” in proceedings of 2nd World Forum on Internet of Things (WF-IoT), 2015.

[26] S. Santesson and H. Tschofenig, “Transport Layer Security (TLS) Cached Information Extension,” draft-ietf-tls-cached-info-16 (WiP), IETF 2014.

[27] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, “DTLS based security andtwo-way authentication for the Internet of Things,” Ad Hoc Networks, (2013) vol. 11, no. 8, pp. 2710–2723.

[28] A. J. Menezes, S. A. Vanstone, P. C. Van Oorschot, Handbook of Applied Cryptography, CRC Press, Inc., Boca Raton, FL, 1996.

[29] R. Roman, J. Zhou, and J. Lopez, “On the Features and Challenges of Security and Privacy in Distributed Internet of Things,” Computer Networks, ELSEVIER, vol. 57, no. 10, pp. 2266 – 2279, 2013.

[30] P. Porambage, A Braeken, A Gurtov, M Ylianttila and Susanna Spinsante “Secure end-to-end communication for constrained devices in IoT-enabled Ambient Assisted Living systems” in proceedings of 2nd World Forum on Internet of Things (WF-IoT), 2015.

[31] S. Raza, S Duquennoy, T. Chung, D. Yazar, T. Voigt, U. Roedig. “Securing communication in 6LoWPAN with compressed IPsec”. In: International conference on distributed computing in sensor systems and workshops (DCOSS). IEEE; 2011. p. 1–8.

[32] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” Computer Networks, vol. 51, pp. 921–960, 2007.

[33] J. Lee, V. Leung, K. Wong, J. Cao, and H. Chan, “Key management issues in wireless sensor networks: current proposals and future developments,” Wireless Communications, IEEE, vol. 14, no. 5, pp. 76 –84, October 2007.

[34] N. Gura, A. Patel, A. W, H. Eberle, and S. C. Shantz, “Comparing elliptic curve cryptography and rsa on 8- bit cpus,” 2004, pp. 119–132.

[35] A. S. W, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” 2005.

[36] Z. Alliance, “Zigbee specification,” ZigBee Alliance, Tech. Rep., June 2007.

[37] “IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans),” IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), 2006.

[38] M. Rezaeirad, M. Orooji, S. Mazloom, D. Perkins, and M. Bayoumi, “A novel clustering paradigm for key pre-distribution: Toward a better security in homogenous wsns,” in Consumer Communications and Networking Conference (CCNC), 2013 IEEE. IEEE, 2013, p. 308–316.

[39] W. Diffie and M. E. Hellman, “New directions in cryptography,” 1976.

[40] A. Hegland, E. Winjum, S. Mjolsnes, C. Rong, O. Kure, and P. Spilling, “A survey of key management in ad hoc networks,” Communications Surveys Tutorials, IEEE, vol. 8, no. 3, pp. 48 –66, qtr. 2006.

[41] B. Zhang and L. Chen, “An improved key management of zigbee protocol,” in Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, ser. IITSI ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 416–418.

[42] S. Zhu, S. Setia, and S. Jajodia, “Leap+: Efficient security mechanisms for large-scale distributed sensor networks,” ACM Trans. Sen. Netw., vol. 2, pp. 500– 528, November 2006.

[43] G. Dini and M. Tiloca, “Considerations on security in zigbee networks,” in Proceedings of the 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, ser. SUTC ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 58–65.

[44] H. Kim, J.-M. Chung, and C. H. Kim, “Secured communication protocol for internetworking zigbee cluster networks,” Comput. Commun., vol. 32, pp. 1531–1540, August 2009.

[45] Advanced Encryption Standard (AES), Federal information processing standards (FIPS 197) Std., November 2001.

[46] L. Zhou and Z. Haas, “Securing ad hoc networks,” Network, IEEE, vol. 13, no. 6, pp. 24 –30, 1999.

[47] F. Stajano and R. Anderson, “The resurrecting duckling: Security issues for ad-hoc wireless networks,” in Security Protocols, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2000, vol. 1796, pp. 172–182.

[48] “Key distribution mechanisms for wireless sensor networks: a survey,” Rensselaer Polytechnic Institute, Tech. Rep., 2005.

[49] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, “A survey of key management schemes in wireless sensor networks,” Comput. Commun., vol. 30, pp. 2314–2341, September 2007.

[50] D. Christin, P. S. Mogre, and M. Hollick, “Survey on wireless sensor network technologies for industrial automation: The security and quality of service perspectives,” Future Internet, vol. 2, no. 2, pp. 96–125, 2010.

[51] D. Whiting, R. Housley, and N. Ferguson, “Counter with cbc-mac (ccm),” United States, 2003.

[52] B. Preneel, “Cbc-mac and variants,” in Encyclopedia of Cryptography and Security, H. Tilborg, Ed. Springer US, 2005, pp. 63–66.

[53] R. R. Mit and R. L. Rivest, “The rc5 encryption algorithm.” Springer-Verlag, 1995, pp. 86–96.

[54] Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, NIST, U.S. National Institute of Standards and Technology Std., May 2005

[55] H. Shafagh and A. Hithnawi, “Poster Abstract: Security Comes First, A Public-key Cryptography Framework for the Internet of Things”, 2014 IEEE International Conference on Distributed Computing in Sensor Systems, (2014), pp. 135-136

[56] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory 22 (1976) 644–654.

[57] A. Shamir, “How to Share a Secret,” Communication ACM, vol. 22, no. 11, pp. 612–613, 1979.

[58] J L. Atzori, A. Lera, G. Morabioto, The internet of things: a survey, Comput. Netw. 54(15) (2010) 2787–2805.

[59] O. Garcia-Morchon, S. Keoh, S. Kumar, R. Hummen, R. Struik, Security considerations in the IP-based internet of things, Draft-Garcia-Core-Security-04 (2012) March 26

[60] Freescale Semiconductor. Freescale Xtrinsic accelerometers optimize resolution and battery life in consumer devices, September 2010

[61] Mouser Electronics STMicroelectronics LIS3DSH accelerometers specification.

[62] C. T. Cornelius, D. F. Kotz, Recognizing whether sensors are on the same body, Pervasive and Mobile Computing, Volume 8, Issue 6, December 2012, Pages 822–836

[63] Dominique Guinard, A Web of Things Application Architecture Integrating the Real-World into the Web, ETH Zurich Diss. ETH No. 19891, 2011

[64] Cisco The Internet of Things Reference Model draft document white papers, 2014

[65] Intel corporations, USA. Intel® Gateway Solutions for the Internet of Things

[66] Sherin C Abraham Internet of Things with Cloud Computing and M2M Communication, Engineering publication Slide share, Sep 2016

Muhammad Aamir, Iqbal. Bachelor of Science, Quaid-i-Azam University, Islamabad, Spring 2000; Master of Science, University of Engineering & technology, Lahore, Summer 2005; M.S Computer Science, University of Louisiana at Lafayette, Fall 2011; Doctor of Philosophy, University of Louisiana at Lafayette, Spring 2017
Major: Computer Science
Title of Dissertation: Distributed Security Paradigm for Resource-constrained Wireless Sensors in the Context of Internet of Things (IoT)
Dissertation Director: Dr. Magdy A. Bayoumi
Pages in Dissertation: 130; Words in Abstract: 444

ABSTRACT

This dissertation addresses new challenges in the Internet of Things (IoT) related to security and privacy. The current transition from legacy internet to Internet of Things leads to multiple changes in its communication paradigms. Today's Machine to Machine (M2M) and Internet of Things architectures further accentuated this trend, not only by involving wider architectures but also by adding heterogeneity, resource capabilities inconstancy, and autonomy to once uniform and deterministic systems and the issue of scalability within a WSN. Unlike internet servers, most of IoT components are characterized by low capabilities in terms of both energy and computing resources and thus, are unable to support complex security schemes. A direct use of existing key establishment protocols to initiate connections between two IoT entities may be impractical unless both endpoints are able to run the required (expensive) cryptographic primitives, thus leaving aside a whole class of resource-constrained devices. In this dissertation, we propose novel security solution approaches for key establishments designed to reduce the requirements of existing security protocols in order to be supported by resource-constrained devices and for the scalability of sensors with a WSN in contest of IoT. We have investigated the feasibility of substituting the key management scheme of ZigBee stack by implementing LEAP+ to enhance its security and scalability capabilities in a WSN. LEAP+ is surprisingly well-suited to different types of

network topologies, device types, and addressing modes offered by ZigBee stack, resolving the issue of scalability due to ZigBee’s key management centralized approach, and our experimental results and performance evaluation parameters illustrated these facts. We designed new key establishment protocols for the constrained wireless sensors to delegate their heavy cryptographic load to less constrained nodes in their neighborhood, exploiting the spatial heterogeneity of IoT nodes. Allowing cooperation between sensor nodes may open the way to a new class of threats, known as internal attacks, that conventional cryptographic mechanisms fail to deal with. This introduces the concept of trustworthiness within a cooperative group. Proposed protocols aim to track nodes behaviors and past performances to detect their trustworthiness and select reliable ones for cooperative assistance. Sensor nodes’ trustworthiness is verified by accompanying them with an accelerometer to detect whether these cooperative sensors are installed on the same body. Based on an extensive analysis and their accelerometers’ data correlations with the base station (mobile phone in this case) accelerometer data, we identify a set of neighboring devices able to provide assistance in performing heavy asymmetric computations effectively without compromising the security of the whole system. Formal security and privacy verifications and performance analyses with respect to the resource-constrained sensor’s energy are also conducted to ensure the security effectiveness and energy efficiency of our proposed protocols.

BIOGRAPHICAL SKETCH

Muhammad Aamir Iqbal received his Bachelor of Science in Physics from Quaid-i-Azam University in Islamabad and his Master of Software Engineering from the University of Engineering & Technology in Lahore in 2005. He joined The Center for Advanced Computer Studies (CACS) at the University of Louisiana at Lafayette and completed his M.S. in Computer Science in the fall of 2011. He started working on his Ph.D. studies in Computer Science in the spring of 2012 at CACS under the supervision of Dr. Magdy A. Bayoumi. He has contributed to several research projects at CACS in the Integrated Wireless Information Network (iWIN) and Very Large Scale Integration (VLSI) labs in research areas such as Wireless Sensor Networks, Internet of Things (IoT), Body area network, RFID security and privacy algorithms, and protocols designs. He completed the requirements for this Doctor of Philosophy in Computer Science in the Spring of 2017. His research interests include security and privacy of data, Internet of Things, networks and communications, and software development.