End-to-End Security Scheme for Mobility Enabled Healthcare Internet of Things

Sanaz Rahimi Moosavi[1], Tuan Nguyen Gia[1], Ethiopia Nigussie[1], Amir-Mohammad Rahmani[1], Seppo Virtanen[1], Hannu Tenhunen[1,2], and Jouni Isoaho[1]

[1] *Department of Information Technology, University of Turku, Turku, Finland*
[2] *Department of Industrial and Medical Electronics, KTH Royal Institute of Technology, Stockholm, Sweden*

Abstract

We propose an end-to-end security scheme for mobility enabled healthcare Internet of Things (IoT). The proposed scheme consists of i) a secure and efficient end-user authentication and authorization architecture based on the certificate based DTLS handshake, ii) secure end-to-end communication based on session resumption, and iii) robust mobility based on interconnected smart gateways. The smart gateways act as an intermediate processing layer (called fog layer) between IoT devices and sensors (device layer) and cloud services (cloud layer). In our scheme, the fog layer facilitates ubiquitous mobility without requiring any reconfiguration at the device layer. The scheme is demonstrated by simulation and a full hardware/software prototype. Based on our analysis, our scheme has the most extensive set of security features in comparison to related approaches found in literature. Energy-performance evaluation results show that compared to existing approaches, our scheme reduces the communication overhead by 26% and the communication latency between smart gateways and end users by 16%. In addition, our scheme is approximately 97% faster than certificate based and 10% faster than symmetric key based DTLS. Compared to our scheme, certificate based DTLS consumes about 2.2 times more RAM and 2.9 times more ROM resources. On the other hand, the RAM and ROM requirements of our scheme are almost as low as in symmetric key-based DTLS. Analysis of our implementation revealed that the handover latency caused by mobility is low and the handover process does not incur any processing or communication overhead on the sensors.

Preprint submitted to Journal of Future Generation Computer SystemsFebruary 15, 2016

Keywords: End-to-End Security, Internet of Things, Smart Healthcare, Mobility, Session Resumption

1. Introduction

Recent advances in information and communication technologies have given rise to a new technology: Internet of Things (IoT) [1, 2, 3]. IoT enables people and objects in the physical world as well as data and virtual environments to interact with each other, hence realizing smart environments such as smart transport systems, smart cities, smart healthcare, and smart energy. The rising cost of healthcare, and the prevalence of chronic diseases around the world urgently demand the transformation of healthcare from a hospital-centered system to a person-centered environment, with a focus on citizens' disease management as well as their wellbeing [4]. It has been predicted that in the following decades, the way healthcare is currently provided will be transformed from hospital-centered, first to hospital-home-balanced in the 2020's, and then ultimately to home-centered in 2030's [5]. This essential transformation necessitates the fact that the convergence and overlap of the IoT architectures and technologies for smart spaces and healthcare domains should be more actively considered [4, 6, 7, 8].

Security is a major concern wherever networks are deployed at large scales. IoT-based healthcare systems deal with human-related data. Although collected from innocuous wearable sensors, such data is vulnerable to top privacy concerns [9, 10, 11, 12]. In IoT-based healthcare applications, security and privacy are among major areas of concern as most devices and their communications are wireless in nature [13]. An IP-enabled sensor in a Medical Sensor Network (MSN), for instance, can transmit medical data of patients to a remote healthcare service. However, in such scenarios, the conveyed medical data may be routed through an untrusted network infrastructure, e.g. the Internet. Hence, in healthcare IoT, security and privacy of patients are among major areas of concern. In this regard, the authentication and authorization of remote healthcare centers/caregivers and end-to-end data protection are critical requirements as eavesdropping on sensitive medical data or malicious triggering of specific tasks can be prevented [14]. Due to direct involvement of humans in IoT-based healthcare applications, providing robust and secure data communication among healthcare sensors, actuators, patients, and caregivers are crucial. Misuse or privacy concerns may restrict people to utilize IoT-based healthcare applications.

Conventional security and protection mechanisms including existing cryptographic solutions, secure protocols, and privacy assurance cannot be reused due to resource constrains, security level requirements, and system architecture of IoT-based healthcare systems [15]. To mitigate the aforementioned risks, strong network security infrastructures for a short and long-range communication are needed. There are significant security solutions to current wireless networks which are not directly applicable to IoT-based healthcare applications due to the following challenges [16]: i) security solutions must be resource-efficient as medical sensors have limited processing power, memory, and communication bandwidth. ii) Medical sensors can be easily lost or abducted as they are tiny in terms of size.

To deal with the mentioned challenges, Constrained Application Protocol (CoAP) [17] proposes Datagram Transport Layer Security (DTLS) [18] to be used for resource-constrained services/applications. DTLS is a complete security protocol as it offers authentication, key exchange, and protection of application data. An IoT-enabled application may be in one of the following four security modes: i) *NoSec*, meaning that the DTLS is disabled and there is no protocol level security. However, the use of *IPsec* as network layer security is recommended. ii) *Symmetric Key-based DTLS*, meaning that DTLS is enabled and symmetric key-based authentication is utilized. iii) *Public Key-based DTLS*, meaning that DTLS is enabled and the resource constrained device has an asymmetric key pair. The public key is not embedded in an X.509 certificate. iv) *Certificate-based DTLS*, meaning that DTLS is enabled and the constrained device has an asymmetric key pair. The X.509 certificate is signed by a Certificate Authority (CA). Medical sensors used in healthcare IoT have limited ROM, RAM, CPU and energy resources. Thus, new challenges arise when using certificates on such resource-constrained devices.

In [19], as shown in Figure 1, we presented a secure and efficient authentication and authorization architecture for IoT-based healthcare systems using smart e-health gateways in a distributed fashion. More precisely, we proposed to exploit the smart gateways' advantageous property of being non-resource constrained for outsourcing the processing burden of end-user authentication and authorization from tiny medical sensors. The system architecture of our proposed IoT-enabled healthcare system includes the following main components: i) *Device Layer*: enabled with ubiquitous identification, sensing, and communication capacity, in which bio-medical and context signals are captured from home/hospital room(s) or patients' body to be used for treatment and diagnosis of medical states. ii) *Fog Layer*: consists of a network of

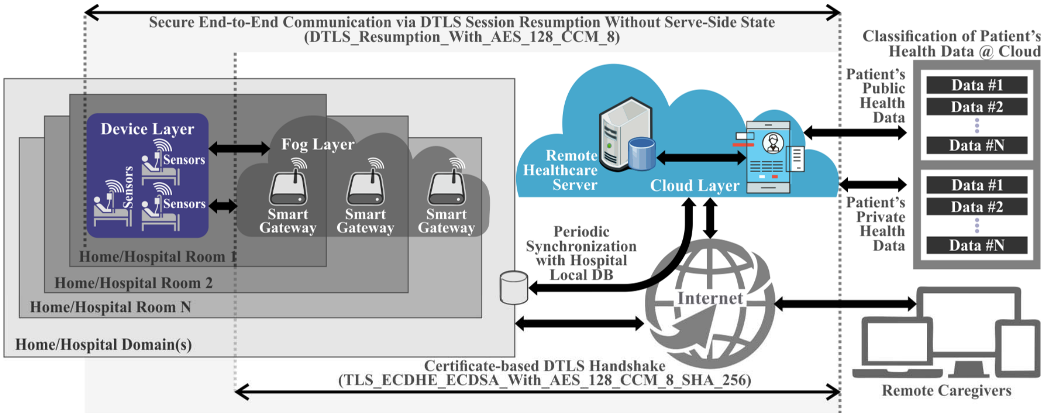
Secure End-to-End Communication via DTLS Session Resumption Without Serve-Side State
(DTLS_Resumption_With_AES_128_CCM_8)
Classification of Patient's Health Data @ Cloud
Device Layer
Sensors
Sensors
Sensors
Fog Layer
Smart Gateway
Smart Gateway
Smart Gateway
Home/Hospital Room 1
Home/Hospital Room 2
Home/Hospital Room N
Home/Hospital Domain(s)
Remote Healthcare Server
Cloud Layer
Patient's Public Health Data
Data #1
Data #2
Data #N
Patient's Private Health Data
Data #1
Data #2
Data #N
Periodic Synchronization with Hospital Local DB
Internet
Remote Caregivers
Certificate-based DTLS Handshake
(TLS_ECDHE_ECDSA_With_AES_128_CCM_8_SHA_256)

Figure 1: The architecture of a healthcare IoT system with secure end-to-end communication

distributed smart e-health gateways where those gateways support various communication protocols and acts as a touching point between the device layer and cloud layer. iii) *Cloud Layer*: this layer is composed of the remote healthcare server and patients' classified health data. iv) *Web Interface*: as a graphical user interface to be used by remote caregivers for final visualization and apprehension.

Recently, there have been efforts in designing *Smart e-Health Gateways* for Healthcare Internet of Things (Health-IoT) systems [4]. In a smart home-/hospital, where the mobility and location of patients are confined to hospital facilities or buildings, gateways can play a key role. The stationary nature of such gateways enables them with the exclusivity of being non-resource constrained in terms of power consumption, memory, and communication bandwidth. By providing the necessary security context to the medical sensors, smart gateways remove the need to authenticate and authorize remote healthcare centers/caregivers from the sensors. Therefore, any malicious activity can be blocked before entering to a medical constrained domain. For this purpose, we employed the certificate-based DTLS handshake as it is the main transport layer security solution for IoT.

In healthcare IoT systems, improving patients' quality of life is important to mitigate the negative effects of being hospitalized. Providing patients with the possibility to walk around the medical environments knowing that the monitoring of their health condition is not interrupted is an important feature. Enabling mobility support for patient monitoring systems offers a high quality of medical service as it allows patients to move around freely within the premises. Patients do not need to be worried about moving around as the system can enable mobility while monitoring their vital signs continuously.

In our previous work [19], the main focus was on the analysis and development of authentication and authorization between peers rather than end-to-end security. In [20], we proposed a session resumption-based end-to-end security scheme for healthcare IoT systems to securely and efficiently manage the communication between medical sensors and remote healthcare centers/caregivers. The proposed scheme relied on the certificate-based DTLS handshake between non-resource-constrained distributed smart gateways and end-users at the start of the communication (initialization phase). To provide end-to-end security, the session resumption technique without server-side state is utilized. The session resumption technique has an abbreviated form of the DTLS handshake and it neither requires heavy-weight certificate-related nor public-key operations as it relies on the previously established DTLS connection.

In this article, an end-to-end security scheme for mobility enabled healthcare IoT is proposed. The main contributions of this article, which is a major extension of our recent works published in [19, 20], are twofold. First, we propose an end-to-end security scheme for healthcare IoT with the explicit consideration of mobility for medical sensors. We exploit the concept of fog layer in IoT for realizing efficient and seamless mobility since fog extends the cloud paradigm to the edge of the network. Second, we analyze the characteristics of the proposed scheme in terms of security and energy-performance on a prototype of a healthcare IoT system through simulation and hardware/software prototype.

The remainder of the article is organized as follows: in section 2, the related work and motivation are discussed. Section 3 presents our proposed system architecture for healthcare IoT. In section 4, the requirements of secure and efficient communication for healthcare IoT system are presented and discussed. Section 5 presents the proposed end-to-end security scheme for healthcare IoT systems. Fog layer-based mobility for our proposed end-to-end security scheme is presented in section 6. Experimental results including energy-performance and security evaluations are provided and discussed in section 7. Finally, section 8 concludes the article.

2. Related Work and Motivation

For the discussion of related work, we recognize three main research directions: (i) IoT-based Healthcare Security, (ii) Smart Gateways, and (iii) Mobility solutions for IoT systems.

2.1. IoT-based Healthcare Security

CodeBlue is one of the most popular healthcare research projects that has been developed at the Harvard sensor network Lab [21]. In this approach, several medical sensors are placed on a patients' body. CodeBlue has been expected to be deployed in in-hospital emergency care, stroke patient rehabilitation and disaster response. The authors of CodeBlue admit the necessity of security for IoT-based medical applications. However, the security aspects of CodeBlue are still left as future work. Lorincz *et al.* [22] suggest that Elliptic Curve Cryptography (ECC)[23] and TinySec [24] are efficient solutions to be used for key generation and symmetric encryption in the CodeBlue project, respectively. Kambourakis *et al.* discuss some attack models and security threats concerning the CodeBlue project: denial-of-service attack, snooping attack, grey-hole attack, sybil attack, and masquerading attacks [25]. An in-hospital patient monitoring system called MEDiSN has been developed at Johns Hopkins University [26]. It consists of multiple physiological motes which are battery powered and equipped with medical sensors in order to collect patients' medical and physiological health information. The MEDiSN architecture focuses on reliable communication, routing, data rate, and QoS [26]. In their proposed architecture, the authors of MEDiSN acknowledged the necessity of having encryption for the physiological monitors. However, they did not mention which cryptosystems have been used for the data confidentiality and integrity. Although the authors claim that security is provided by the MEDiSN architecture, their study did not reveal much information regarding the security implementation. An architecture called Sensor Network for Assessment of Patients (SNAP) [13] has been proposed to address the security challenges concerning the wireless health monitoring systems. However, the main problem of the aforementioned architecture is that it does not authenticate users when providing medical data. Furthermore, the data collected from medical sensors are conveyed to a controller in plaintext format. Hence, the medical data of the patients can be modified or intercepted by a malicious user. In [27], a lightweight identity-based cryptography solution called IBE-Lite has been proposed. The basic idea of IBE-Lite is to balance security and privacy with availability. Nevertheless, several security and privacy issues as well as efficiency problems are recognized in IBE-Lite. First, in their work, Tan *et al.* do not consider sensor to base station/end-user data authentication. Therefore, falsified medical information can be introduced or treated as authentic due to the lack of authentication schemes. Second, IBE-Lite cannot resist against replication attacks. Consequently, an adversary can insert malicious medical sensors into the network.

To establish interoperable network security between end-peers from independent network domains, variants of conventional end-to-end security protocols have been recently proposed among which Datagram Transport Layer Security (DTLS) is one of the most relevant protocols [18]. In this regard, Hummen *et al.* [14] present an implementation of a delegation architecture based on an off-path delegation server. Their proposed delegation-based architecture relies on a centralized delegation server. Due to this, their proposed architecture lacks scalability and reliability. More precisely, their architecture cannot be extended to be employed for multi-domain infrastructures, e.g. large in-home/hospital domains. Also, their proposed architecture suffers from a considerable network transmission overhead resulting to a long transmission latency. Moreover, if an adversary performs a Denial of Service (DoS) attack or compromises the delegation server, a large quantity of stored security context of a constrained domain can be retrieved.

2.2. Smart e-Health Gateway

There have been many efforts in designing gateways for one or several specific applications and architectural layers. Muller *et al.* [6] present a gateway called SwissGate which handles and optimizes the operation of sensor networks. They transparently employ their proposed gateway on home automation applications. Shen *et al.* [7] propose a prototype of a smart 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) border router that makes local decisions of users' health states based on a Hidden Markov Model. Finally, Rahmani *et al.* [4] present a smart e-health gateway called *UT-GATE* in order to bring intelligence into IoT-based ubiquitous healthcare systems. These gateways are intelligent in the sense that they have been empowered to autonomously perform local data storage and processing, to learn, and to make decisions at the edge of the network (i.e., in a distributed fashion), thanks to the provided embedded processing power and storage capabilities of the gateways. A smart gateway can rapidly provide preliminary results and reduce the redundant remote communication to cloud servers by using data aggregation, embedded machine learning, and inferences, thus offering the basic services at the edge of the network. In this way, remote cloud computers will just provide premium services which are often computationally intensive and require access to the central database.

In a smart home/hospital, gateway is in a unique position between Body/-Patient/Local Area Network (BAN/PAN/LAN) and Wide Area Network (WAN). This promising opportunity can be exploited by different means such as collecting health and context information from those networks and providing different services accordingly. As mentioned above, compared to the conventional gateways which often just perform basic functions such as translating between the protocols used in the Internet and sensor networks, smart e-health gateways are empowered with the property of being non-resource constrained in terms of processing power, memory, power consumption, and communication bandwidth. In [19] and [20], we demonstrated the use of a smart gateway to handle medical sensors' main computation and communication overhead that results from end-user authentication and authorization.

2.3. Mobility Solutions for IoT Systems

In [28], Valenzuela *et al.* propose a solution to support mobility for inhome health monitoring systems using wearable sensors. This approach utilizes a coordinator sensor attached to patients' body that is responsible for all the communications between wearable sensors and network Access Points (APs). Jara *et al.* in [29, 30, 31], propose a solution to support the mobility of sensors employed to monitor patients in hospital environments. This approach supports intra-mobility exploiting elements such as sink nodes and gateways in their proposed architecture. This proposal supposes that each mobile node has a base network and can move into other networks. Fotouhi *et al.* [32] present a handover approach for mobility support in Wireless Sensor Networks (WSNs) which can be easily employed for Body Sensor Networks (BSNs) [33, 34]. In their work, different parameters are utilized to specify the time for handover, but the most important ones are the Received Signal Strength (RSS) and the sensor velocity. To verify the quality of the link as well as to decide handover mechanism, this solution requires a continuous exchange of probe or acknowledge messages between the sensor and the corresponding access point. However, this continuous messages exchange weaken the network in terms of transmission overhead, memory, and energy consumption.

In [19], our main focus was on the development and analysis of a secure and efficient authentication and authorization architecture, while in [20] we proposed a secure end-to-end communication scheme via session resumption for healthcare IoT system. In these works patients' mobility support was not considered. This article essentially extends our previous works by incorporating enhanced mobility while providing secure end-to-end communication. Our proposal is motivated by the fact that to enable mobility for healthcare IoT systems, an intermediate computing layer, that is the fog layer [35], can be exploited between the device layer and the cloud layer. More precisely, the mobility support can be provided to the medical sensors ubiquitously from the fog layer so that no more reconfiguration is needed in the resource-constrained device layer.

3. Healthcare IoT System Architecture

Healthcare IoT systems are distinct in that they are built to serve human beings, which inherently raises the requirements of safety, security, and reliability. Moreover, they have to provide real-time notifications and responses regarding the status of patients. In a typical healthcare IoT system, to monitor patients' activities and vital signs, the system has to ensure the safety of patients. In addition, physicians, patients, and other caregivers demand a dependable system in which the results are accurate and timely, and the service is reliable and secure. To guarantee these requirements, the smart components in the system require a predictable latency and reliable communication with the upper computing layer. The conventional cloud-based approaches cannot assure the requirements of healthcare IoT systems, as the connection to the cloud is less reliable and may incur additional latency. In this article, we utilize a novel system architecture as a suitable paradigm to address the aforementioned requirements.

Fog computing is a paradigm extending cloud computing and its services to the edge of the network. Fog distinguishes from cloud in its proximity to end-users/devices, dense geographical distribution, real-time interaction, support for mobility, heterogeneity, interoperability and pre-processing along with interplay with the cloud. Fog devices are heterogeneous in nature, ranging from end-user devices and access points to edge routers and switches allowing their use in wide variety of environments. Fog services can be implemented in a variety of devices ranging from smart phones to edge routers and access points with a reasonable support of local storage and processing.

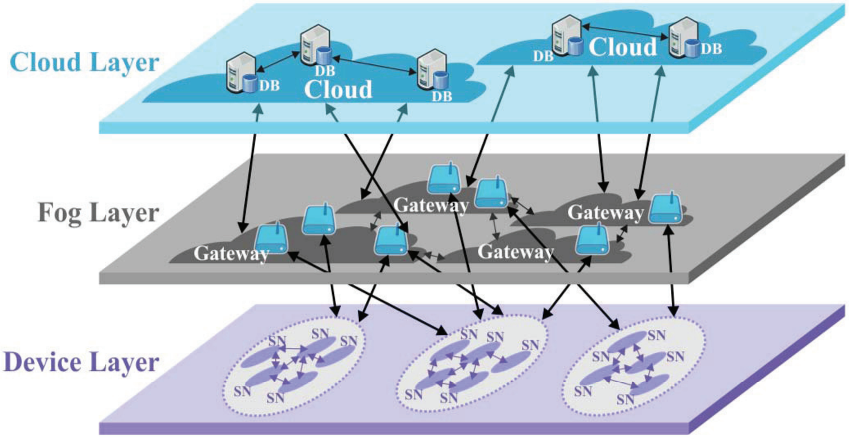

Figure 2: The three-tier system architecture of the healthcare IoT system (SN and DB stand for Sensor Node and Database, respectively)

The three-tier system architecture of the healthcare IoT system on which we apply our end-to-end security scheme is shown in Figure 2. In such a system, patients' health-related information is recorded by implanted or wearable medical sensors with which the patient is equipped for personal monitoring of multiple parameters. This health-related data may also be supplemented with context information, i.e. time, date, location, and relevant environment information which enables the recognition of abnormal patterns and the making of more precise inferences. The functionality of each layer in this architecture is as follows:

(i) **Device Layer**: the lowest layer consisting of several physical devices including implantable or wearable medical sensors that are integrated into a tiny wireless module to collect contextual and medical data. Enabled by the ubiquitous identification, sensing, and communication capacity, bio-medical and context signals are captured from the body and/or the room. The signals are used for managing the treatment and diagnosis of medical conditions. The signal is then transmitted to the upper layer (i.e., smart gateways in the Fog layer) via wireless or wired communication protocols such as IEEE 802.15.4, Bluetooth LE, Wi-Fi, etc.

(ii) **Fog Layer**: the middle layer consists of a network of interconnected smart gateways. Cloud computing paradigm is an efficient alternative to establishing and maintaining private servers and data centers. Particularly, due to its "pay-as-you-go" business model, it gives more efficiency and freedom to web applications. However, these features demand high computation and storage as well as batch processing. This model enables developers and end-users to exploit cloud services with a minimum knowledge of the underlying hardware and infrastructure. However, this becomes an issue in applications which require low latency (emergency care). Such challenges are addressed in the Fog computing paradigm by extending the cloud services to the edge of the network. As mentioned before, we exploit Smart e-Health gateways which support different communication protocols, act as a the touching point between a sensor network and the local switch/Internet. A smart gateway receives data from different sub-networks, performs protocol conversion, and provides other higher level services. It acts as repository (local database) to temporarily store sensors' and users' information, and provides intelligence at the edge of the network. In addition, by taking responsibility for handling some computational and processing burdens of the sensors and the cloud, a smart gateway at the fog layer can cope with many challenges such as energy efficiency, scalability, and reliability issues [35].

(iii) **Cloud Layer**: The cloud layer includes broadcasting, data warehousing and big data analysis servers, and a hospital local database that periodically performs data synchronization with the remote healthcare database server in the cloud. In the cloud layer, accessability to patients-related health data is classified as public data (e.g., patients' ID or blood type) and private data (e.g., DNA).

4. Requirements of Secure and Efficient Communication For Healthcare IoT System

In this section, various criteria that represent desirable characteristics of secure communication for a healthcare IoT system are presented.

Data Confidentiality: All relevant data being transmitted between communicating peers remains unknown for others. To prevent patients' health data from the leakage attack, such data needs to be kept confidential. This can be achieved using strong encryption schemes meaning that even if an adversary eavesdrops on transmitted packets, he/she cannot easily get access to them. Data confidentiality should also be resistant to any device compromise attack, for example, medical sensor or smart gateway compromise attack.

Data Integrity: Ensures that patients' health data is received in the exact way as it was sent and it has not been manipulated in transit. Since in healthcare IoT systems most devices and their communications are wireless in nature, maintaining data integrity is a necessary task. To provide data integrity, a Cyclic Redundancy Checksum (CRC), that is used to detect random errors during packet transmission, or a Message Authentication Code (MAC) are usually employed.

Mutual Authentication and Authorization: Allows the communication peers to ensure and validate the identity of each other. Mutual authentication needs to be done in the whole system so that private medical information cannot be accessed by any unauthorized user. This way, an adversary cannot claim to be a valid user to obtain patients' health data or inject invalid information. Authentication can be achieved by sending a MAC along with the message. On the other hand, authorization indicates that only authorized users/sensors can access resources and services in an IoT-enabled healthcare system.

Data Freshness and Forward Security: Data freshness indicates that patients' health data is fresh and an adversary has not replayed the previously transmitted data. The property of forward security ensures that the revelation of current encrypted medical sensors' data does not threaten the security of the previously transmitted health data.

Availability: Ensures that medical sensors and all services utilized in an IoT-enabled healthcare system can constantly provide services to authorized users whenever required (despite of possible Denial of Service (DoS) attacks). Fulfilling availability, however, is a difficult task as DoS attacks can exhaust the power supplies of the medical sensors or heavily reduce the network performance by jamming the radio channel.

Scalability and Lightweight Solutions: Scalability refers to the capability of an IoT-enabled healthcare system to continue functioning well even if such a system may be modified in terms of size (e.g. sensors, hardware or services may be added/removed). In emergency situations, an IoT-enabled healthcare system should have the capability of fast reaction without compromising the patients' security and privacy. It is necessary to minimize communication, computation, and memory overhead of medical sensors due to the low capabilities of these sensors. Hence, cryptographic solutions being proposed should be lightweight to fulfill the aforementioned requirements.

Data Access Control: In healthcare IoT systems, caregivers (i.e. doctors, pharmacists, nurses, etc.) are directly involved with patients' physiological and medical data. Thus, a real-time role-based access control needs to be available to restrict caregivers' access based on their privileges.

Patient Consent: Patients' consents are always essential when caregivers decide to circulate their medical records to another healthcare sector/hospital in order to provide higher quality of healthcare. Informed consent refers to the process of getting patients' permission before conducting medical procedures/interventions (e.g. medical treatment's nature, consequences, harms, risks, and benefits). Informed consent is a fundamental principle of healthcare and it is collected according to the guidelines of medical and research ethics.

Mobility support: Mobility is one of the most important challenges in healthcare IoT systems which increases the applicability of these technologies. The mobility support enables patients to go for a walk around the medical domain(s) while he/she is continuously monitored. Furthermore, mobility allows the patient to move from his/her base MSN to other rooms for medical tests without loosing the continuous monitoring.

End-to-End Security: End-to-end security is one of the major requirements in healthcare IoT systems. This feature enables the end-points of a healthcare IoT system, that is caregivers and medical sensors, to securely communicate with each other beyond the independent network.

5. End-to-End Security Scheme For Healthcare IoT System

In [19], we presented a secure and efficient authentication and authorization architecture for healthcare IoT system using smart e-health gateways called *SEA* (lower black arrow shown in Figure 1). In [20], we presented a comprehensive end-to-end security scheme for healthcare IoT systems using the session resumption technique (upper black arrow shown in Figure 1). Before presenting the fog layer-based mobility for our proposed end-to-end security scheme, we briefly explain our previous work in this section.

5.1. Secure and Efficient Authentication and Authorization Architecture

In the paradigms of healthcare IoT, not only data can be collected by smart devices (medical sensors) and transmitted to end-users (caregivers), but end-users can also access, control, and manage medical sensors through the Internet. Since patients' health data is the basis for enabling applications and services in healthcare IoT, it becomes imperative to provide secure end-to-end communication between end-users and medical sensors to protect the exchange of health data. In addition, privacy of patients and key negotiation materials should be protected to prevent anyone other than the negotiation peers from learning the contents of the negotiations. It is also important that malicious activities be blocked at the entrance to MSNs. Hence, mutual authentication and authorization of end-users and devices used in healthcare IoT systems is a crucial task.

Our proposed architecture called *SEA* exploits the role of smart e-health gateways in the fog layer to perform the authentication and authorization of remote end-users securely and efficiently on behalf of the medical sensors [19]. By providing the established connection context to the medical sensor nodes, these devices no longer need to authenticate and authorize a remote healthcare center or a caregiver. Thus, any malicious activity can be blocked before entering to a constrained medical domain. The architecture of our proposed healthcare IoT monitoring system in home/hospital domain(s) is shown in Figure 1. In such an architecture, patient health-related information is recorded by body-worn or implanted sensors, with which the patient is equipped for personal monitoring of multiple parameters. This health data can be also supplemented with context information (e.g., date, time, location, and temperature) which enables to identify unusual patterns and make more precise inferences about the situation. Our proposed SEA focuses on a fact that the smart e-health gateway and the remote end-user have sufficient resources to perform various heavy-weight security protocols as well as certificate validation. To provide end-to-end communication between a remote end-user and a constrained medical device, distributed smart e-health gateways are introduced to build a transport layer security protocol that is Datagram Transport Layer Security (DTLS) [18].

DTLS handshake protocol is the main transport layer security solution for IoT. As Figure 3 presents, a full handshake begins with a *ClientHello* message, that includes the security parameters for the connection which is used later during the handshake to compute the pre-master secret key. Flight 3 contains additional cookie from *ClientHelloVerify.* Flight 4 includes several messages and starts with *ServerHello* message which contains the negotiated cipher suite for the current handshake and the smart gateway's random value which is utilized later during the handshake to compute the

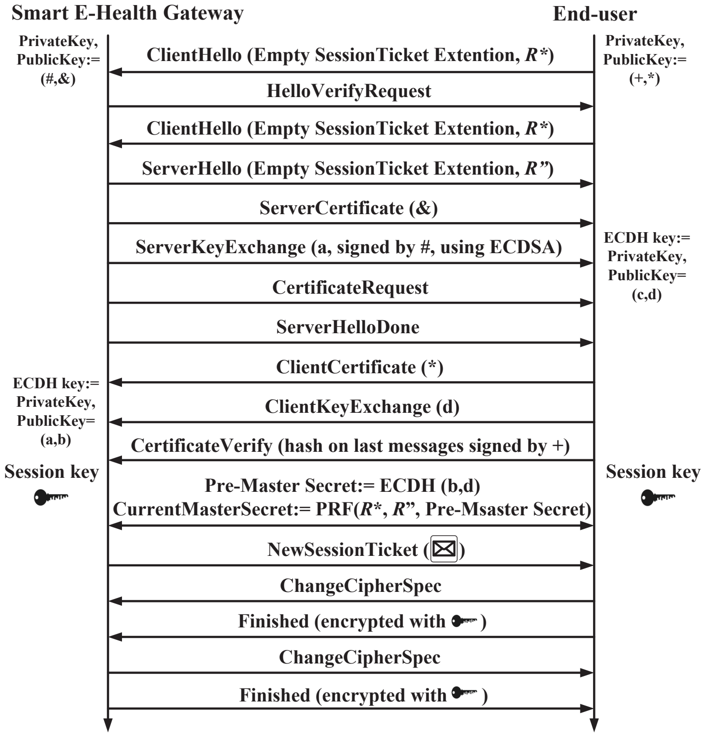
Smart E-Health Gateway
End-user
PrivateKey,
PublicKey:=
(#,&)
PrivateKey,
PublicKey:=
(+,*)
ClientHello (Empty SessionTicket Extention, R*)
HelloVerifyRequest
ClientHello (Empty SessionTicket Extention, R*)
ServerHello (Empty SessionTicket Extention, R”)
ServerCertificate (&)
ServerKeyExchange (a, signed by #, using ECDSA)
ECDH key:=
PrivateKey,
PublicKey=
(c,d)
CertificateRequest
ServerHelloDone
ClientCertificate (*)
ECDH key:=
PrivateKey,
PublicKey=
(a,b)
ClientKeyExchange (d)
CertificateVerify (hash on last messages signed by +)
Session key
Session key
Pre-Master Secret:= ECDH (b,d)
CurrentMasterSecret:= PRF(R*, R”, Pre-Msaster Secret)
NewSessionTicket (☒)
ChangeCipherSpec
Finished (encrypted with 🔑)
ChangeCipherSpec
Finished (encrypted with 🔑)

Figure 3: Message flights for the full certificate-based DTLS handshake while issuing a session ticket [19]

master secret key. The agreed cipher suite relies on supported cipher suites by the end-user. If the smart gateway and the end-user cannot agree on a common cipher suite, the handshake is canceled with a *HandshakeFailure* alert message. The next message of flight 4 is smart gateway's *Certificate* message which holds gateway's certicate-chain. The first certificate in the chain includes the smart gateway's public key which is created using *OpenSSL* in version of 1.0.1.j. OpenSSL is an open source project for implementing SSL, TLS and various cryptography libraries such as symmetric key, public key, and hash algorithms. It is commonly utilized for creating and managing keys and certificates. Once the certificate is validated, the end-user can extract the smart gateway's public key. The *CertificateRequest* is only sent in a mutual handshake and includes the lists of the smart gateway's valid certificates. The *ServerKeyExchange* message is only sent with specific cipher suites that need more parameters in order to compute a master secret key. The cipher suite employed in this work is $TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8\_SHA\_256$. The name indicates the use of elliptic cryptography, particularly- *Elliptic Curve Diffie-Hellman* (*ECDH*) and *Elliptic Curve Digital Signature Algorithm* (*ECDSA*). Furthermore, for encryption AES-based CCM with an IV of 8 bytes is used. With this cipher suite, ServerKeyExchange message contains the ECDH public key of the smart gateway and the detail of the associated elliptic curve. The *ServerHelloDone* message announces the end of flight 4 messages. The first message of flight 5 is the end-user's certificate in case mutual authentication is run. *ClientKeyExchange* includes additional parameters utilized to compute the master secret key. In this case, the ECDH public key of the smart gateway is conveyed. *CertificateVerify* is a message which enables the end-user to prove to the smart gateway that it carries the private key which corresponds to the public key contained in the certicate. Thus, it is only transmitted in the mutual authentication. With the *ChangeCipherSpec* message, the end-user informs the smart gateway that next messages will be encrypted using the agreed cipher suites and secret keys. The *Finished* message includes the encrypted hash over all flight messages which ensure that both peers have been performing handshake based on unmodified flight messages and the handshake is performed successfully. In flight 6, the smart gateway responds with its own ChangeCipherSpec and Finished messages. With the Finished messages both peers agree to send and receive securely protected application information over this connection. Upon this connection setup, as shown in Figure 4, the remote end-point and the smart e-health gateway mutually authenticate each other.

It is supposed that within the certificate-based DTLS handshake, from one hand, the smart gateway authenticates (*Auth-req.1*) the remote end-user through certificates. In this regard, similar to current web browsers, smart gateways hold a pool of trusted certificates. On the other hand, the smart gateway either authenticates (*Auth-req.2*) to the remote end-user through certificates within the DTLS handshake or based on an application-level password once the handshake is terminated. Once the mutual authentication between the end-user and the smart gateway is done successfully, the end-user authorizes (*Authz.*) as a trusted entity so that a data query from the end-users' side is transmitted to the medical sensor through the smart gateway. To facilitate the security and authorization of communication, it is required that both entities, the constrained medical sensor and the smart gateway, also mutually authenticate (*Mut-auth.*) one another once during the initialization

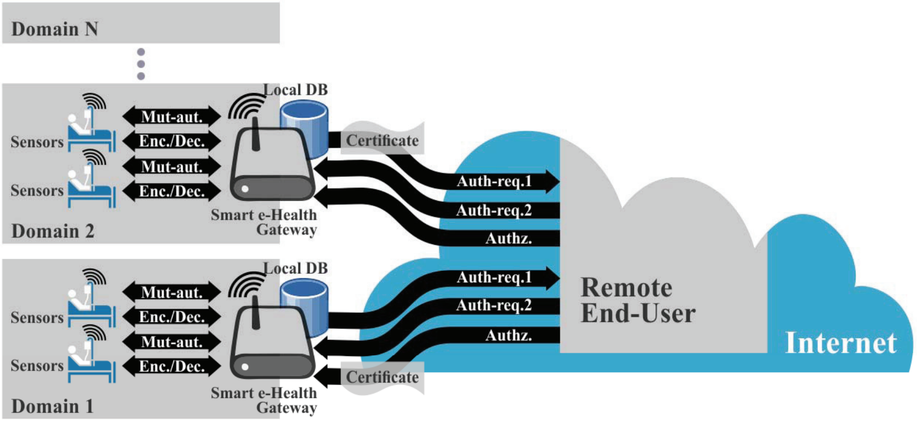

Figure 4: The Proposed SEA Architecture Overview Using Distributed Smart e-Health Gateways

phase. In SEA, this is done by performing a public key-based DTLS handshake between both entities. Although symmetric key-based DTLS handshake provides an efficient alternative to public key-based DTLS handshake, the symmetric key-based handshake needs secret keys to be pre-shared and readily available at both communication end-points. Moreover, compared to the symmetric key-based DTLS handshake, obtaining secret points in a public key-based handshake implies the computation of elliptic curve discrete logarithm problem. Since solving the discrete logarithm problem is as hard as integer factorization, this problem cannot be solved effortlessly [23].

Once mutual authentication and key exchange protocol is done, it is required that both peers agree upon a common key. This shared common key can be generated using an already agreed elliptic curve between the both peers. Using the shared common key, one peer (i.e., constrained medical sensor) encrypts the gathered patients' medical data applying the efficient *Advanced Encryption Standard* (*AES-CCM*) [36] algorithm and transmits the encrypted medical information (*Enc./Dec.*) to the smart e-health gateway and vice versa. AES-CCM offers confidentiality, integrity, and authentication of payload compared to other commonly known symmetric encryption/decryption algorithms (e.g., RC5, and Triple-DES), it is known as one of the most efficient ones. Moreover, AES is supported by many constrained devices used for IoT platforms. This make AES-CCM a desirable encryption/decryption algorithm choice for constrained devices.

Our SEA architecture achieved the following benefits: (i) network transmission overhead and latency were reduced compared to the most recently proposed architectures. This is because a great part of the work, that is authentication and authorization of a remote end-user/ healthcare center, is shifted to be performed by distributed smart e-health gateways. (ii) the privacy of patients, vital certificates, and key negotiation materials were effectively protected, and (iii) the scalability and reliability of the system were enhanced as the architecture was changed from centralized to distributed.

5.2. The Proposed End-to-End Security Scheme

In SEA [19], our main focus was on the development and analysis of an authentication and authorization architecture for IoT-enabled healthcare systems rather than end-to-end secure communication. In [20], we enabled end-to-end secure communication between end-points of a healthcare IoT system (i.e., medical sensors and end-users) by developing a session resumption-based scheme which offloads the encrypted session states of DTLS towards a non-resource-constrained end-user. The main motivation to employ the DTLS session resumption is to mitigate the overhead on resource-constrained sensors. Because, transmitting and processing of messages in the certificate-based DTLS handshake are resource intensive tasks. The session resumption technique is an extended form of the DTLS handshake which enables a client/server to continue the communication with a previously established session state without compromising the security properties. The session resumption approach improves the performance of the DTLS handshake in terms of required bandwidth, computational overhead, and number of transmitted messages. The main idea to employ session resumption is to perform heavy-weight operations only once, during an initial DTLS handshake connection (initialization) phase. Thus, the peers need to keep minimal session state, even after the session is terminated. The session resumption enables the peers to resume the secure connection without the need for running expensive operations and transmitting long certificates.

Two types of DTLS session resumption techniques have been proposed by IETF for constrained network enviroments [17]. (i) *Abbreviated DTLS handshake* where both peers have similar resources and both peers maintain session state through connections. (ii) *DTLS session resumption without server-side state* which is an extension of DTLS handshake that allows a server to offload the encrypted session state towards a non-resource-constrained client [37]. In [20], we employed the second type of session resumption (i.e. with-

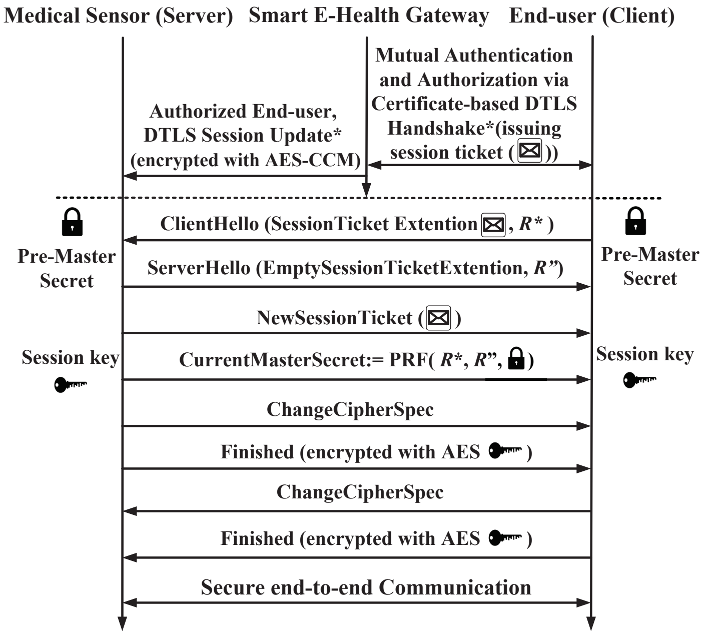
Medical Sensor (Server)
Smart E-Health Gateway
End-user (Client)
Mutual Authentication and Authorization via Certificate-based DTLS Handshake*(issuing session ticket (⊠))
Authorized End-user, DTLS Session Update* (encrypted with AES-CCM)
ClientHello (SessionTicket Extention ⊠, R*)
Pre-Master Secret
Pre-Master Secret
ServerHello (EmptySessionTicketExtention, R")
NewSessionTicket (⊠)
Session key
Session key
CurrentMasterSecret:= PRF(R*, R", 🔒)
ChangeCipherSpec
Finished (encrypted with AES)
ChangeCipherSpec
Finished (encrypted with AES)
Secure end-to-end Communication

Figure 5: The proposed session resumption based end-to-end security for healthcare Internet of Things [20]

out server-side state) that offloads the encrypted session state of the tiny sensors towards the non-resource-constrained end-users/caregivers [18][37]. This is due to the asymmetry in resources between medical sensors and end-users/caregivers considering the constrained nature of sensors.

Before enabling secure end-to-end communication, as we presented earlier, a full certificate-based DTLS handshake needs to be performed once by the end-user and the smart e-health gateway (initialization phase). The protocol flow of the full certificate-based DTLS handshake while issuing a session ticket (to be used later in DTLS session resumption) is shown in Figure 3. Here, the client (i.e. end-user) indicates its support for session resumption with an empty session resumption extension in the *ClientHello* message. On the other hand, the server (i.e. medical sensor) indicates its support for session resumption with an empty session resumption extension in the *ServerHello* message. In addition, during the handshake procedure, the smart gateway needs to build a new session ticket which holds: (i) the key name that recognizes the key utilized to encrypt the state, (ii) the validation of the ticket, and (iii) the encrypted state. Once the full certificate-based

DTLS handshake between the aforementioned end-points is done successfully, the smart gateway updates the medical sensor about the validity of the end-user as well as the status of the DTLS handshake. This is done by encrypting the respective information using AES-CCM encryption algorithm. The AES-CCM algorithm ensures the confidentiality, integrity and authentication of the transmitted payloads. Here, the encryption key is used as secret key, which is shared between the smart gateway and the medical sensor and generated by utilizing the mutually agreed elliptic curve cryptographic algorithm. More details regarding the shared secret key generation can be found in [19]. This enables medical sensors to perform the session resumption with authorized and validated end-users.

To provide secure end-to-end communication between an end-user and a medical sensor, the end-user needs to initiate the session resumption mechanism with the sensor by sending a *ClientHello* message (Figure 5). This time, the *ClientHello* message comprises a session resumption extension maintaining the session ticket and a random value R^*. During this step, the medical sensor uses the received encrypted and authorized session update from the smart gateway in order to resume the DTLS connection which has previously been established between the end-user and the smart gateway. The protocol flow for the DTLS session resumption without server-side state used in this work is shown in Figure 5. Upon receiving the *SessionTicket* extension, the medical sensor which acts as a server needs to decrypt and verify the correctness of the ticket using the corresponding key which is the pre-master secret. When the session ticket is completely verified, the sensor responds with a *ServerHello* message holding an empty session resumption extension and a random value $R^{”}$. In the same flight, the sensor also issues a new session ticket, which contains the information of the current state, that is, the current master secret. The current master secret is computed using the Pseudo Random Function (PRF), that is, a HMAC-based secret expansion function, over the previous master secret key (pre-master secret) and the exchanged random values R^* and $R^{”}$, respectively. The random values provide the property of forward secrecy meaning that revelation of the current single key just allows access to the information of that session and does not threaten the security of the previous DTLS sessions. The new session ticket is conveyed through the *NewSessionTicket* message and kept by the end-user for a possible subsequent session resumption. This way the resource-constrained sensor offloads the computational and processing burden of its state towards the non-resource-constrained end-user. Later, by exchanging the *ChangeCi-*

pherSpec messages, the new keying material is utilized in order to secure the communication channel. Finally, by exchanging the *Finished* messages the correctness of the agreed keys and the integrity of all exchanged messages are verified. This concludes the handshake and provides the exchange of secured application data.

In this work, to generate the *SessionTicket*, the revised version of recommended ticket construction proposed in [37] is used. This is because the recommended ticket construction leads to an excessive ticket size for resource-constrained network environments. Therefore, it is necessary to provide a revised version of the recommended ticket construction that will take the constraints of the device/network into account with respect to the transmission overheads. The *NewSessionTicket* message includes a lifetime value and a session ticket. The lifetime value represents the number of seconds until the session ticket expires. The structure of the session ticket is opaque to the communicating peers and only the ticket issuer can access the session ticket information. The recommended ticket structure presented in [37] suggests to use AES-CCM for encryption with a 12 byte Initialization Vector (IV) and a 32 byte MAC based on HMAC-SHA-256. However, in this work, an 8-byte MAC based on HMAC-SHA-256 and a 12-byte IV are utilized, as they are the recommended cipher suites for secure CoAP over DTLS [17].

The major advantages offered by our scheme compared to the conventional end-to-end security solution [38] can be found in [20]. We applied our proposed session resumption-based end-to-end security scheme for healthcare IoT to the full system architecture shown in Figure 1. As can be seen from the architectural viewpoint, the end-to-end security is fulfilled by (i) using the full initial certificate-based DTLS between end-users and smart gateways and (ii) utilizing session resumption technique which enables end-users and sensors to directly communicate and transmit the encrypted health-related information. The full procedure considerably alleviates the processing load on tiny sensors in terms of authentication, authorization, certificate related functionalities, and public key cryptography operations.

6. Fog Layer-Based Mobility For The Proposed End-to-End Security Scheme

Mobility support is one of the most important issues in healthcare IoT systems. In such systems, improving patients' quality of life is essential. Providing patients with the possibility to walk around the hospital wards knowing that the monitoring of their health condition is not interrupted is an essential feature. Using a portable patient monitoring system offers a high quality of medical service by providing freedom of movement to patients. Mobility enables patients to go for a walk around the medical domain(s) while they are monitored. In addition, mobility allows the patient to move from his/her base MSN to other rooms for medical tests without loosing the continuous monitoring. This scenario can also be extended to other environments such as a nursing house or in-home patient monitoring. The main goal of the continuous monitoring in the healthcare IoT systems is to achieve a knowledge base from the patient which enables the remote server and the Knowledge Base System (KBS) to detect symptoms, predict, and manage the illnesses. Mobility can be categorized into two main topics denoted as macro-mobility and micro-mobility. The movement of medical sensors between various medical network domains distinguishes the macro-mobility. Micro-mobility assumes that medical sensors move between different MSNs within the same domain.

To achieve a continuous monitoring of patients considering the mobility support, it is essential to develop self-configuration or handover mechanisms which are capable of handling secure and efficient data transfers among different MSNs. A data handover mechanism is defined as the process of changing or updating the registration of a mobile sensor from its associated base MSN to the visited MSN, for example, when moving across the hospital's wards. Data handover solutions should enable the ubiquity when they need to work autonomously without human intervention. The handover mechanism should also offer medical sensors continuous connectivity, if there exist several gateways in the hospital or nursing/home environments.

Medical sensors carried by patients are utilized to collect various biological or physiological parameters. Healthcare IoT services are supposed to serve patients in a seamless and continuous way when they are moving in a hospital a nursing facility or at home. More precisely, the mobility support should be provided to the medical sensors ubiquitously from the upper layer (i.e. Fog layer) so that zero reconfiguration is needed in the sensor layer. Fog layer-based handover solutions try to endow healthcare IoT systems with ubiquitous features and provide continuous patient monitoring as well as mobility support.

6.1. Requirements of mobility support for a healthcare IoT System

In this subsection, we present different requirements that need to be fulfilled while offering mobility support for a healthcare IoT system.

(1) In healthcare IoT, mobility must be supported in both star and mesh topologies including single- and multi-hop routing. Mesh networks are mostly formed by nodes with a high degree of mobility.

(2) Signalling must be minimized by removing the use of broadcast/multicast flooding as well as the frequency of link scope broadcast/multicast messages. Reduction of the mentioned mobility signalling messages mitigates the transmission overhead.

(3) Mobility solutions should be compatible and interoperable with the current IPv6 protocols such as Internet Control Message Protocol version 6 (ICMPv6) and Mobile Internet Protocol version 6 (MIPv6).

(4) In the fog layer, a local gateway must notify other available gateways about the presence of mobile sensors in its domain. The reason is that binding necessary updates about the network must be performed by gateways rather than the mobile sensors to unburden tiny sensors from performing heavy tasks.

(5) Global addressing must be supported in mobility solutions. Medical sensors must be addressable anytime needed independent of their current locations. In healthcare IoT, it is one of the main challenges to accomplish global connectivity with the devices using the current Internet infrastructure.

(6) Header information and payloads regarding data messages should be optimized carefully. This reduces fragmentation, the transmission overhead of data messages, and latency while roaming.

(7) Mobility solutions must be based on distributed storage of patients' medical information rather than conventional centralized approaches to support fault tolerance.

(8) The authentication and authorization of medical sensors, smart gateways and caregivers must be performed to ensure the protection of resources, confidentiality, and integrity of the medical information.

(9) Robust security solutions must be provided as healthcare IoT requires ensuring the protection of patients' medical information. Security support can be provided by the AES algorithm which is provided in the data link layer. However, stronger mechanisms to guarantee patients' privacy as well as the security of their medical data can be offered by IPSec in the network layer and DTLS in the transport layer.

(10) In real-time healthcare IoT, mobility detection must be agile so that it avoids delays, jitter, and interruptions of the communication during the data handover process. Data handover procedures (on the evaluation of specific metrics) can be categorized into two main groups: movement parameters and communication parameters. The movement parameters are based on the node position, and movement direction, and velocity. Such parameters are difficult to capture in resource-constrained sensors made to collect just physiological parameters. The second group utilizes the communication parameters in order to handle the requirements for the handover task. The wireless link between two devices can be evaluated using two different metrics: the Received Signal Strength Indicator (RSSI) and the Link Quality Indicator (LQI).

According to [39], the most frequently monitored parameter utilized to evaluate the handover decisions is the Received Signal Strength Indicator (RSSI). The RSSI represents the signal power of a message received by a node which is mostly measured in decibels (dB). The alteration of this value should be directly related to the distance between a sender and a receiver. However, the value of this metric suffers from interference from the surrounding environment and, thereby, this relation is not linear in most situations. The evaluation of RSSI can be performed in two different ways:

(i) Choosing the best value: In this approach, if a patient carrying medical sensors moves to an overlapped coverage area of two or more smart gateways, the one with the higher RSSI value is the one with which the medical sensor chooses to communicate. Due to the oscillation of the RSSI, this model can lead to unnecessary data handovers when a sensor is under several smart gateways' coverage zones. Despite this unpleasant behavior, this model is easy to be deployed and if optimized, it can minimize the data handover costs.

(ii) Making a decision based on comparison against a threshold value: To mitigate the number of unnecessary data handovers performed by the

> previous approach, this model recommends the use of a threshold value to decide the proper moment to switch to a new gateway. If a sensor moves out from the registered smart gateway's coverage area, the RSSI value will be decreased. If this value undershoots to a predefined threshold value, the sensor needs to be registered to another nearby smart gateway which can receive signals with satisfactory signal strength.

It should be noted that proposing an efficient policy for mobility support in fog-based architectures is beyond the scope of this article. Instead, the key contribution of this work is to present how our proposed session resumption-based end-to-end security scheme can be extended to be efficiently maintained and managed when mobility takes place. In other words, it can be considered as a sub-process of a full mobility procedure to address security aspects after it is decided by a policy making module that roaming should be performed from a smart gateway to another.

6.2. Mobility Scenario

Figure 6 presents the scenario where a patient wearing medical sensors decides to move from its room (base network) to other rooms (visited networks). We assume a mobility scenario which consists of several MSNs for remote patient monitoring in a hospital or nursing/home environment. In the considered scenario, patients may roam through the hospital wards or move to other rooms due to some medical tests (e.g., Laboratory or X-ray).

In the case that a moving sensor loses its connection with one of the smart gateways, he/she will stop being monitored by the caregivers. This condition is not favorable in situations where real-time and continuous monitoring is necessary. To enable seamless transitions of medical sensors, providing an efficient and robust data handover mechanism among smart gateways, considering the limitations of sensors, is of essential importance. The mobility scenario is discussed in three phases in the following subsections.

6.2.1. Message Exchange in patients' base MSN

This phase presents the initial state of the medical sensors where each sensor is connected to its base MSN via smart e-health gateway and exchange the required messages. These messages may consist of data frames requests, responses, and acknowledgments of data transmission between the medical sensors and the smart gateways.

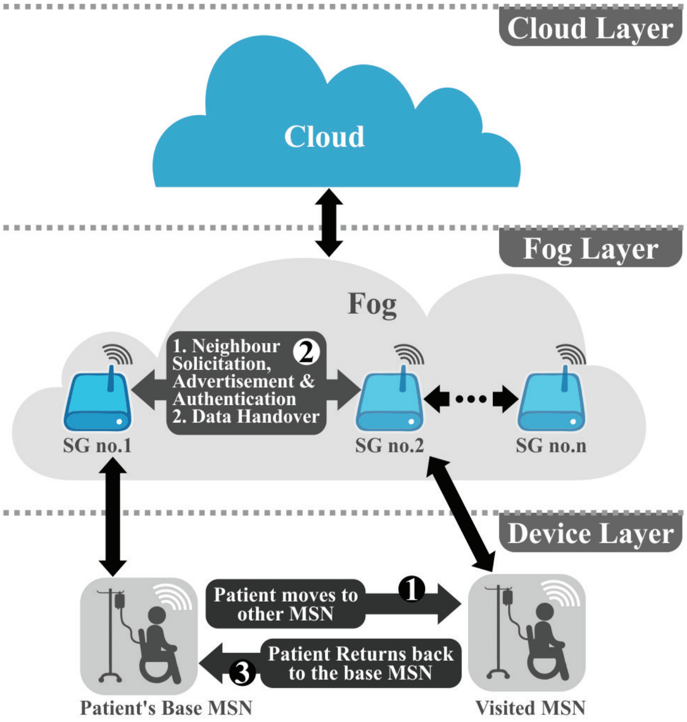
Cloud Layer
Cloud
Fog Layer
Fog
1. Neighbour Solicitation, Advertisement & Authentication
2. Data Handover
2
SG no.1
SG no.2
SG no.n
Device Layer
Patient moves to other MSN
1
Patient Returns back to the base MSN
3
Patient's Base MSN
Visited MSN

Figure 6: Mobility Scenario

The data frames include: (1) information regarding the DTLS session states for the subsequent DTLS session resumption and (2) information about the validity of remote caregivers. Information is exchanged between both peers using the aforementioned AES-CCM algorithm. Request messages are queries to the medical sensor to either get or change some values. Response messages include replies to the request messages where the results of the operation can be obtained. In addition, the request and response messages include information that needs to be transmitted between the sensor and the gateway during the DTLS handshake to perform mutual authentication.

6.2.2. Entering to a new medical subnetwork

Healthcare IoT services are supposed to be offered to patients in a seamless and continuous way when they are moving. When a patient moves out of his/her base MSN, the sensor detects that the quality of its connection with the associated smart gateway is reduced below a pre-defined threshold. We propose to provide mobility support to the sensors from the fog layer to alleviate processing and computation burden of the sensors. To do so, the smart gateway located in the base network needs to check, through the fog layer, whether the medical sensor is accessible from other gateways. This type of mobility (micro-mobility) is just provided to those sensors that are in the same domain/sub-network and their IP addresses do not change. This type of scenario is desirable for MSNs of a hospital as the entire network relies on the same domain.

To provide continuous monitoring of patients, efficient and seamless data handover mechanisms between smart e-health gateways are needed. These mechanisms should take the following features into consideration: 1) Data handover between smart gateways should be quick and seamless considering that the connection to the sensor needs to be preserved during the whole process. 2) After a successful data handover, the changes of routes to the moving medical senor should be spread quickly by the entire healthcare IoT system. 3) The number of messages which need to be exchanged among gateways should be kept minimal (transmission overhead). As a result, to enable mobility for healthcare IoT systems, the following functions need to be performed in the fog layer between smart gateways:

(i) Neighbour Solicitation, Advertisement, and Authentication: Neighbour solicitation and advertisement functions need to be done between the smart gateways in the fog layer to enable seamless mobility. The successful integration of multiple smart gateways on a shared backbone (i.e. fog layer) offers an efficient mobility support. To facilitate the security and the authorization of communication between available smart gateways, it is also required that gateways mutually authenticate one another. As presented earlier, smart gateways are non-resource-constrained devices and they are intelligent in the sense that they have been empowered to autonomously perform local data storage and processing, to learn, and to make decisions at the edge of the network. Hence, the mutual authentication between gateways can be done securely and efficiently using the ECDSA algorithm which was previously presented and analyzed in *SEA* [19].

(ii) Data Handover: Data handover is defined and considered as the process of changing/updating the registration of a sensor from one smart gateway to another one. For example, when moving across hospitals' different rooms. This mechanism enables the mobility support of medical sensors in healthcare IoT domains. In a case that a moving medical sensor loses its connection with one of the smart gateways or if it takes too long to be registered/updated by a new one, the desirable continuous communication and monitoring cannot be ensured. Thus, the smart gateway located in patients' base network needs to periodically send update messages to other gateways in the same domain (e.g., hospital). These messages may include information about the authorized sensors as well as caregivers. Thereby, when a patient enters to another MSN, due to some medical tests, no authentication needs to be done between the sensor and the new gateway. The reason is that the gateway located in the visited network has already been updated, with all necessary information regarding the communication, by the gateway in the base MSN. However, in the case that a new mobile sensor is detected in an MSN, the authentication needs to be performed. As a result, any malicious activity can be discovered and blocked before entering to an MSN.

6.2.3. Returning back to the base MSN

When the patient returns back to his/her base network, the medical sensor sends a re-association request to inform the home smart gateway regarding its new location.

As can be noticed from Figure 7, mobility is enabled in our proposed end-to-end security scheme using the fog concept. It is shown that by exploiting the fog layer, the mobility support can be provided to the medical sensors ubiquitously without compromising the end-to-end security.

7. Implementation and Evaluation

The system architecture illustrated in Figure 1 is implemented for experimental evaluation, with the main goal of secure and efficient authentication and authorization as well as providing mobility for the proposed end-to-end security scheme. To Implement our proposed architecture, we setup a platform that consists of medical sensors, UT-GATE smart e-health gateways, a remote server, and end-users. UT-GATE is constructed from the combination of a Pandaboard [40] and a Texas Instruments (TI) SmartRF06 board that

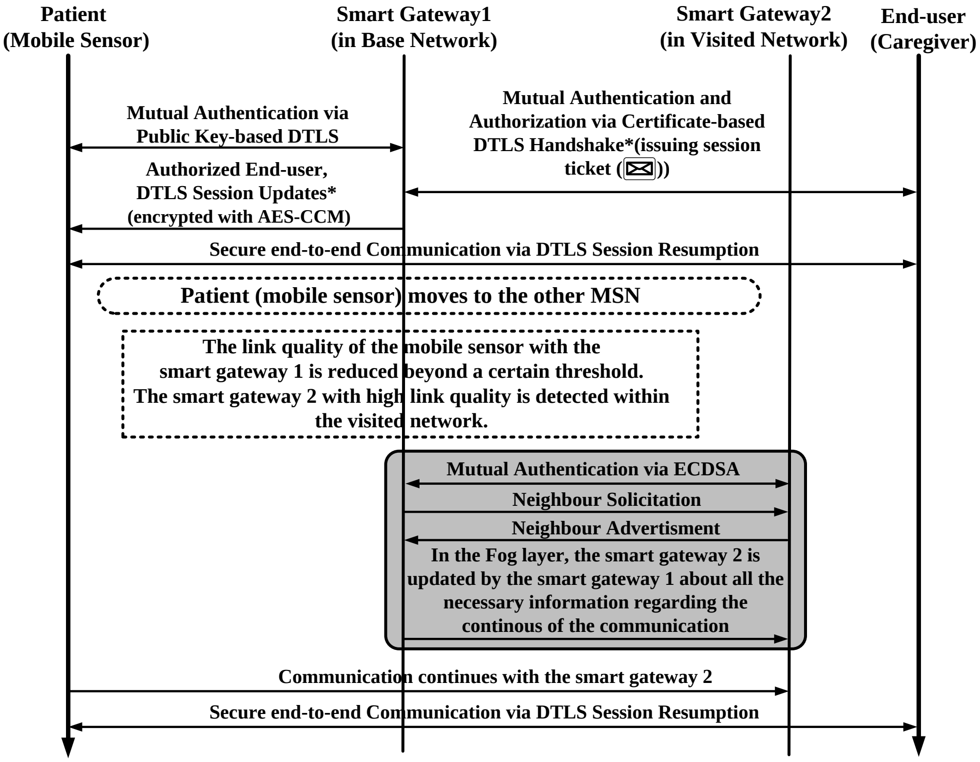
Patient
(Mobile Sensor)
Smart Gateway1
(in Base Network)
Smart Gateway2
(in Visited Network)
End-user
(Caregiver)
Mutual Authentication via
Public Key-based DTLS
Mutual Authentication and
Authorization via Certificate-based
DTLS Handshake*(issuing session
ticket (✉))
Authorized End-user,
DTLS Session Updates*
(encrypted with AES-CCM)
Secure end-to-end Communication via DTLS Session Resumption
Patient (mobile sensor) moves to the other MSN
The link quality of the mobile sensor with the
smart gateway 1 is reduced beyond a certain threshold.
The smart gateway 2 with high link quality is detected within
the visited network.
Mutual Authentication via ECDSA
Neighbour Solicitation
Neighbour Advertisment
In the Fog layer, the smart gateway 2 is
updated by the smart gateway 1 about all the
necessary information regarding the
continous of the communication
Communication continues with the smart gateway 2
Secure end-to-end Communication via DTLS Session Resumption

Figure 7: The handshaking procedures of the proposed end-to-end security scheme for mobility enabled healthcare IoT

is integrated with a CC2538 module [41]. The Pandaboard is a low-power and low-cost single-board computer development platform based on the TI OMAP4430 system-on-chip (SoC) following the OMAP architecture and fabricated using 45 nm technology. The OMAP4430 processor is composed of a Cortex-A9 microprocessor unit (MPU) subsystem including dual-core ARM cores with symmetric multiprocessing at up to 1.2 GHz each. In our configuration, UT-GATE uses 8GB of external memory and is powered by Ubuntu OS which allows to control devices and services such as local storage and notification. To investigate the feasibility of our proposed architecture, the *Wismote* [42] platform, which is a common resource-limited sensor, is utilized in Contiki's network simulation tool Cooja [14]. Wismote is equipped with a 16MHz MSP430 micro-controller, an IEEE 802.15.4 radio transceiver, 128KB of ROM, 16KB of RAM, and supports 20-bit addressing. For the evaluation, we use the open source tool *OpenSSL* version 1.0.1.j to create elliptic curve public and private keys from the NIST P-256 (prime256v1) and X.509 certificates. X.509 certificates are the prevailing form of certificates and are employed in the certificate-based mode of DTLS [43]. The server association to the end-user is created using OpenSSL API which provides all necessary functions related to end-users including configuration, certificate, handshake, session state, and cipher suites to support session resumption. *TinyDTLS* [44] is used as the code-base of the proposed scheme, in this work. TinyDTLS is an open-source implementation of DTLS in symmetric key-based mode. We extend it with support for the certificate-based DTLS as well as session resumption. For the public-key functions, we utilize the *Relic-toolkit* [45] that is an open source cryptography library tailored for specific security levels with emphasis on efficiency and flexibility. The MySQL database is set up for static and non-static records. Static records which are managed by system administrators, include white tables, essential data required by the DTLS handshake, and an end-user authentication mechanism. Non-static records store up-to-date bio-signals that are synchronized between the Pandaboard database and a cloud server database. The cloud server database is processed using xSQL Lite which is the third party tool for data synchronization. With respect to the cryptographic primitives and to make a fair comparison, we followed similar cipher suites (which are current security recommendations for constrained network environments [17]) as employed in the most recently proposed authentication and authorization architecture for IP-based IoT [45]. In this regard, we utilize elliptic curve NIST-256 for public-key operations, $AES\_128\_CCM\_8$ (with an IV of 8 bytes) for symmetric-key, and SHA256 for hashing operations.

7.1. Energy-Performance Evaluation

In this subsection, we analyze our proposed end-to-end security scheme from the energy-performance point of view.

Transmission Overhead: To perform the certificate-based DTLS handshake, as shown in Figure 3, all message flights need to be transmitted to establish a DTLS connection. When transmitted over size-constrained IEEE 802.15.4 radio links, these messages must additionally be split into several packet fragments due to their extensive message size [14]. As Table 1 presents, the transmission overhead of the proposed SEA approach to the most recently proposed architecture for a successful certificate-based DTLS

Table 1: Performance comparison with the most recently proposed authentication and authorization approach for IoT

	Transmission-overhead (byte)	**Latency-GE (s)**	**Latency-NG (s)**
SEA approach (This Work)	1190	5.001	~ 15
Hummen et al. [14]	1609	6.08	~ 15
SEA approach improvements (%)	26	16	0

conncetion is compared. As the baseline for this evaluation, a simulation environment is implemented using Cooja. Then, the transmission overheads of the certificate-based DTLS protocol between two wirelessly connected WiSMotes is measured. To quantify the transmission overhead, the *pcap* tool in combination with the Cooja simulator is employed. The presented results signify averages over 100 measurement runs. In a delegation-based architecture, the measured transmission overhead of the certificate-based DTLS handshake is 1609 bytes which causes in total 24 fragments for the transmission of all handshake messages from the delegation server to the end-user [14]. In contrast, the proposed SEA architecture requires transmission of 1190 bytes and it causes 18 fragments totally. As a result, the transmission overhead in our proposed architecture is reduced by 26% compared to the delegation-based architecture.

Latency: Latency is defined as the time needed for a data packet to travel from one designated point to another. It is an essential metric for real-time applications. In this work, we calculate the latency from two perspectives: i) The communication latency from a smart gateway to an end-user for the authentication and authorization process, and ii) Data handover latency between two smart gateways for the proposed mobility enabled end-to-end security scheme. The communication latency and the data handover latency are estimated on a 20Mb/s broadband Internet connection.

(i) *Communication Latency*: To estimate the communication latency, the processing time which is spent from sensor node to the end-user (*NE*) is calculated. This processing time deduced from the summation of communication latency from sensor node to smart gateway (NG) and smart gateway to end-user can be written as: $Latency_{NE} = Latency_{NG} + Latency_{GE}$. In this work, to compute the communication latency from the UT-Gate to the end-user, a proxy server is adjoined to the network. Through the proxy server, the transmission latency between the end-user and the UT-Gate can be easily measured as the proxy

Table 2: Data handover latency between two smart gateways with different packet size

Packet Size (byte)	**Data Handover Latency (milliseconds)**
10	2.288
30	2.410
50	2.517
100	2.884
200	3.113
500	3.342
1K	3.685
5K	4.588

server listens to requests transmitted from the end-user to the UT-Gate and vice versa without tampering or modifying them. To compute the communication latency of *GE*, the Fiddle [4] proxy server, which is a desktop application, is employed to track requests and responses. Fiddle offers a large number of services including security testing and HTTP/HTTPS traffic recoding. According to our analysis, the proposed SEA architecture achieves an almost equivalent *NG* processing time to the delegation-based architecture [14]. However, the proposed SEA approach considerably reduces the processing time required for *GE* compared to the delegation-based architecture. As shown in Table 1, in SEA, the processing time required for *GE* is about 5.001 seconds whereas this time increases to about 6.08 seconds in the delegation-based architecture. Thus, regarding the latency from the gateway to the end-user, the proposed architecture obtains about 16% improvement compared to the delegation-based architecture.

(ii) *Data Handover Latency*: To demonstrate how our proposed end-to-end security scheme enables mobility, we implemented a real system in which two UT-GATE gateways with the configuration described above are employed. We assume that these gateways are connected through the fog layer where one of the gateways acts as a client and the other one acts as a server. In the experiments, we created a 100-byte lookup table for each gateway that consists of: i) control data which consists of the DTLS session resumption state, information about the authorized caregivers, medical sensors' IDs, and patients' IDs. ii) Patients' health data that includes heart rate, body temperature, and oxygen saturation. In our analysis, we calculated the latency of the data handover process between the gateways. To show the scalability of our method, we considered messages with different sizes which may need to be exchanged between the gateways for the data handover process. The results are shown in Table 6. As can be deduced from the Table, the data handover latency between two gateways is negligible and mobility is supported in an agile way without any computational and processing burden to the sensors. In addition, by increasing the packet size, latency marginally increases showing the scalability of our scheme. As mentioned before, seamless mobility is a necessity in healthcare IoT systems. The experiments show that our proposed end-to-end security scheme also provides support for this feature. It should be noted that proposing a novel mobility approach is orthogonal to the proposed idea. It means that any fog-based mobility solution can be combined with our security scheme.

Sensor-side Processing Time: For the evaluation, in Cooja, we configured two Wismotes as a client and a server. Once the booting process is performed, the client initiates the handshake by sending the *ClientHello* message. After a successful handshake, we measured the total processing time at the sensor-side (server). The results of our measurements using three different approaches are shown in Table 4. As can be seen from the Table, the symmetric key-based mode and our session resumption-based scheme require almost similar processing time. The proposed scheme requires 20 ms less processing time than the symmetric key-based mode. This is due to the fewer message flights needed to be exchanged in the session resumption (compared to the full symmetric key-based DTLS), resulting in less computations at the sensor-side. The processing time for the certificate-based DTLS handshake is considerably higher than both the symmetric key-based and the session resumption-based modes. The certificate-based DTLS requires about 5690 ms at the sensor-side which is mainly due to the expensive public key-based operations (i.e. ECDSA and ECDH).

Client-side Processing Time: The total processing time at the client (end-user) side using three different approaches is shown in Table 4. For the client-side, we used a machine with $IntelCore^{TM}i5-4570$ CPU operating at 2.2 GHz and having 6 GB of RAM. The processing time of the proposed scheme using DTLS session resumption is 45 ms, where as the conventional symmetric key-based requires 49 ms. This is due to the lesser

Table 3: Client-side processing time and total run-time performance of different DTLS modes to provide end-to-end security

	Client-side Processing Time (ms)	**Run-time Performance) (ms)**
DTLS Session Resumption Without Server-side State (*DTLS_Session_Resumption_WITH_AES_128*) (This Work)	45	205
Certificate-Based DTLS (*DTLS_ECDHE_ECDSA_WITH_AES_128_CCM_8_SHA_256*)	3744	9434
Symmetric key-Based DTLS (*DTLS_PSK_WITH_AES_128_CCM_8*)	49	229

Table 4: Sensor-side processing time and energy consumption of different DTLS modes to provide end-to-end security

	Sensor-side Processing Time (ms)	**Energy Consumption (mJ)**
DTLS Session Resumption Without Server-side State (*DTLS_Session_Resumption_WITH_AES_128*) (This Work)	160	8.87
Certificate-Based DTLS (*DTLS_ECDHE_ECDSA_WITH_AES_128_CCM_SHA_256*)	5690	315.79
Symmetric key-Based DTLS (*DTLS_PSK_WITH_AES_128_CCM_8*)	180	9.99

number of control messages needed for session resumption, compared to the full symmetric key-based DTLS. The processing time for certificate-based DTLS handshake, is considerably higher than both the symmetric key-based and the session resumption-based modes. The certificate-based DTLS requires approximately 3744ms at the client-side which is mainly due to the expensive public key-based operations. Compared to symmetric key-based and certificate-based DTLS, our session resumption-based scheme has 8.1% and 98.7% improvements in terms of client-side processing time, respectively.

Run-time Performance: In this work, run-time refers to the time it takes for the handshake between the medical sensor and the end-user to be done successfully. To provide end-to-end security, we calculate the total run-time performance of three different DTLS modes. The results are presented in Table 3. As can be seen from the Table, our scheme which utilizes the DTLS session resumption technique is about 97% and 10% faster than certificate-based and symmetric key-base DTLS handshake, respectively.

Energy Consumption: To measure the consumed energy of each sensor, we

Table 5: Memory footprint of different DTLS modes to provide end-to-end security

	RAM overhead (KB)	**ROM overhead (KB)**
DTLS Session Resumption Without Server-side State ($DTLS\_Session\_Resumption\_WITH\_AES\_128$) (This Work)	3.51	14.29
Certificate-Based DTLS ($DTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8\_SHA\_256$)	7.8	41.1
Symmetric Key-Based DTLS ($DTLS\_PSK\_WITH\_AES\_128\_CCM\_8$)	2.96	13.49

utilize the equation: $E(mJ) = U(V) \times I(mA) \times t(ms)$ where U represents the supply voltage, I is the current draw of the hardware, and t is the time. We calculate the energy consumption of the Wismote sensor when performing the DTLS session resumption, the symmetric key-based DTLS handshake, and the certificate-based DTLS handshake. According to the datasheet available in [42], the Wismote has a current consumption of 18.5 mA and a supply voltage of 3 V. The results are presented in Table 4. As can be seen from the Table, our techniques are considerably more energy efficient in comparison to the certificate-based DTLS handshake technique. It saves 11% of energy compared to the symmetric key-based DTLS.

Memory Requirement: To calculate total RAM and ROM requirements of the utilized session resumption technique, we used the msp430-size tool which is provided by the MSP430-gcc compiler. We evaluated RAM and ROM requirements using three different modes of DTLS handshake: (i) DTLS session resumption used in our proposed scheme, (ii) symmetric key-based DTLS handshake, and (iii) certificate-based DTLS handshake. As shown in Table 5, the certificate-based DTLS consumes about 2.6 times more RAM and 3 times more ROM resources than what is required by the symmetric key-based DTLS handshake. These overheads are considerable for devices having limited resources particularly in terms of memory. In [19], we presented that our proposed IoT-enabled healthcare architecture enables the constrained medical sensor to unburden all certificate-related and public-key operations to the distributed smart e-health gateway. Thus, the memory burden of the medical sensors is considerably alleviated. Compared to the symmetric key-based mode, our proposed session resumption-based scheme adds a negligible memory overhead (RAM and ROM overheads are only increased by 0.5 kB and 0.8 kB, respectively). This minor increase is due to the session resumption extension and the storage of the session tickets.

7.2. Security Evaluation

In this section, we analyze our proposed end-to-end security scheme from the security perspective. We conclude this section by comparing our work with the most recently proposed schemes found in the literature.

Data Confidentiality: In this work, to provide confidentiality, 128-bit AES-CCM with a 16 byte initialization vector is employed to protect patients' information that needs to be transmitted between communicating peers. In the proposed scheme, even if an adversary eavesdrops on some or all of the transmitted patients' health data, he/she cannot access those data easily as they are encrypted using the secure and robust 128-bit AES encryption algorithm. A brute force attack on 128-bit AES would require $3.4*10^{38}$ years [36].

Data Integrity: In this work, to ensure that the transmitted data is received in the exact same way as it is sent, a 8 byte Message Authentication Code (MAC) based on HMAC-SHA-256 is employed. This is done by creating the MAC of a message m (that needs to be transmitted) using the SHA-256 hash function and a shared secret key K (*SessionKey*) over m which can be written as: $HMAC(m) = SHA256(K,m) = HMAC(K,m) = D$. The MAC is a cryptographic checksum on message m that uses the *SessionKey* to detect both accidental and intentional modifications of the message. Based on the above equation, the secure HMAC generates a fixed length hash digest D from the message m. It has the characterestics of being simple to compute, while infeasible to retrieve the m from the given hash digest D. The small changes in m result in a different hash value. Such features are specified as preimage and collision resistant, respectively. Thus, our proposed scheme ensures the property of data integrity.

Mutual Authentication and Authorization: In SEA [19], we presented that sensors used in medical applications are highly resource-constrained for which reason they cannot cope with cryptography techniques demanding heavy computations. To overcome this limitation, we proposed to employ non-resource-constrained smart e-health gateways in distributed fashion to perform the authentication and authorization of end-users mutually on behalf of the sensors. The proposed architecture relied on the certificate-based DTLS handshake and the employed cipher suite was $TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8\_SHA\_256$. The name indicates the use of elliptic cryptography, particularly- *Elliptic Curve Diffie-Hellman* (*ECDH*) and *Elliptic Curve Digital Signature Algorithm* (*ECDSA*). We proved that, within the certificate-based DTLS handshake, from one hand, the smart e-health gateway authenticates the remote end-user through certificates. On the other hand, the smart gateway either authenticates to the remote end-point through certificates within the DTLS handshake mechanism or based on an application-level password once the handshake is terminated. Therefore, mutual authentication and authorization of peers is fulfilled in our work.

Forward Security: As mentioned earlier, the property of forward security ensures that the revelation of current encrypted patients' health data should not threaten the security of previously transmitted data. In this work, using the certificate-based DTLS handshake, the shared *SessionKey* between peers is derived using ECDH. For this, as Figure 3 presents, each of the peers, the smart gateway and the end-user, produce their own pair of private and public keys on an already agreed elliptic curve. *(a,b)* for the smart gateway and *(c,d)* for the end-user. Then, the peers exchange their public keys and the DTLS session key over the elliptic curve is calculated as: $a \times b = SessionKey = c \times d$ where × is the scalar multiplication on elliptic curve. Elliptic Curve Cryptography (ECC) relies on the general hypothesis that the elliptic curve discrete logarithm problem is infeasible or at least it cannot be solved in a reasonable time. Once the *SessionKey* is derived using ECDH, the x-coordinate value of *SessionKey* serves as a shared secret between the end-user and the smart gateway. The derived shared secret is utilized further to protect the communication/data transmitted between the peers. As shown in Figure 3, since b and d are public values of the peers, their exchange through an unencrypted channel does not compromise or provide any information concerning the *SessionKey.* This is because obtaining the *SessionKey* implies the computation of elliptic curve discrete logarithm problem (ECDLP). Solving this problem is not easily possible. The reason is that ECDLP is believed to be much harder to solve than its counterpart over finite fields (DLP) or the integer factorization problem (FP), the two main alternatives for public key cryptography.

Scalability and reliability: In SEA [19], we proposed a new architecture for IoT-enabled healthcare system (i.e. in-home/hospital environments) which relies on distributed smart e-health gateways. In our proposed architecture, we also discussed that in a multi-domain smart home/hospital environment, if an attacker runs a DoS attack or compromises one of the smart gateways, only the associated medical sub-domain is disrupted. However, in most of the recently proposed delegation-based architectures, if an attacker performs a Denial of Service (DoS) attack or compromises the delegation server, a large quantity of stored patients' health data can be retrieved. Specifically, in multi-domain networks, a DoS attack can disrupt all the available constrained medical domains as the functionality of those IoT-based domains depends on the centralized delegation server. Hence, compared to most recently proposed delegation-based architectures [14][38][46], our proposed IoT-enabled healthcare architecture is more scalable and reliable as the architecture is changed from being centralized to distributed.

Lightweight Solutions: In the previous section, we noted that conventional security and protection mechanisms including existing cryptographic solutions, secure protocols, and privacy assurance cannot be re-used due to resource constraints, security level requirements, and system architecture of IoT-based healthcare systems. To alleviate the constrained medical sensors from all heavy processing burdens: (i) we exploit the non-resource-constrained distributed smart gateways to perform the authentication and authorization of remote end-users securely and efficiently on behalf of medical sensors. (ii) to provide secure end-to-end communication between the end-user and the tiny medical sensor, we used the lightweight DTLS session resumption technique. This is because session resumption has an abbreviated form of a full DTLS handshake that relies on the previously established security context, which neither requires heavy-weight certificate-related nor public-key cryptography operations.

Access Control: In our scheme, as we discussed earlier in the mutual authentication and authorization section, the validation and authorization of data and end-user access control are handled by smart e-health gateways instead of the resource-constrained medical sensors. Thus, any malicious activity is blocked at the smart gateway before an unauthorized users get access to the medical network domain(s).

Smart Gateway and sensor Spoofing: In the proposed architecture, if an adversary pretends to be a trusted smart e-health gateway/medical sensor, from one hand, he/she can get access to all information related to the DTLS sessions. On the other hand, patients' encrypted health data can also be revealed to the attacker. In this work, as Figure 3 and Figure 5 present, the smart e-health gateway and the end-user as well as the medical sensor and the smart e-health gateway share a symmetric *SessionKey* between each other. As it was presented earlier in the forward security section, this shared *SessionKey* is generated using ECDH and solving this algorithm is not easily possible [23]. Thus, by spoofing the smart gateway/sensor, an attacker cannot deceive the end-user for access to data concerning the DTLS session.

Denial of Service Attack (DoS): In SEA [19], we discussed in more detail about the drawbacks of the state-of-the-art architectures proposed for IoT-based systems. To give an example, in the most recently proposed delegation-based architecture developed by Hummen *et al.* [47], if an adversary performs a DoS attack or compromises the centralized delegation server, a large number of stored security context related to constrained domains can be retrieved. Specifically, in multi-domain networks, a DoS attack can disrupt all the available medical domains as the functionality of the IoT-based healthcare systems still relies on the centralized delegation server. However, in our proposed IoT-enabled healthcare system, in a multi-domain smart home/hospital network, if an attacker runs a DoS attack or compromises one of the smart e-health gateways, just the associated medical sub-domain can be disrupted. The reason is that in our proposed architecture, the authentication and authorization tasks of a centralized delegation server is broken down to be performed by distributed smart e-health gateways.

Stolen DTLS Session Tickets: In a DTLS handshake, an eavesdropper may attempt to obtain the ticket and to utilize it to establish a session with the server. However, a stolen ticket does not help the adversary to resume the session as the session ticket is encrypted and the adversary does not have any knowledge about the secret key. To minimize the feasibility of success of this attack, in this work (as proposed by IETF [17]), the lightweight 128-bit AES in CCM mode and the HMAC-SHA-256 algorithms are used by the DTLS server to provide confidentiality and integrity, respectively. This prevents an adversary from successfully executing a brute force attack to obtain the tickets' contents.

Forged DTLS Session Tickets: A malicious adversary can alter or forge the session ticket in order to resume a DTLS session, to impersonate as a valid user, to extend the lifetime of a session, or to obtain additional privileges. To avoid the forged ticket attack, we used the strong integrity protection algorithm HMAC-SHA-256 to protect the session ticket. In the data integrity section, we described in detail more how the integrity requirements can be fulfilled using HMAC-SHA-256.

End-to-End Security: In our proposed scheme, during the initialization phase, the smart e-health gateways' main tasks are transmitting the information related to the DTLS sessions as well as the necessary security contexts to the medical sensors. However, the only performers of both the encryption and decryption of patients' health data (in DTLS session resumption) are the end-user and the medical sensor. Thus, both end points directly communicate with each other without the necessity of a smart gateway as an

Table 6: Security comparison of different schemes providing end-to-end security ("✓" indicates that the scheme supports the mentioned security feature, and "✗" indicates that the scheme does not support the feature.)

Security Features	**Hummen *et al.***[14]	**Granjal *et al.***[38]	**Kang *et al.***[46]	**This Work**
Data Confidentiality	✓	✓	✓	✓
Data Integrity	✓	✓	✓	✓
Mutual Authentication and Authorization	✓	✓	✓	✓
Forward Security	✓	✓	✗	✓
Architecture Scalability	✗	✗	✗	✓
Lightweight Solutions	✓	✓	✓	✓
Access Control	✗	✗	✓	✓
Smart Gateway and Sensor Spoofing	✗	✗	✓	✓
Denial of Service (DoS) Attack	✗	✗	✓	✓
End-to-End Security	✓	✗	✗	✓

intermediary node. Thus, end-to-end security is ensured in our scheme.

The security comparisons of our proposed end-to-end security scheme and the most recently proposed approaches are presented in Table 6. The state-of-the-art end-to-end security approaches proposed for IoT are presented by Hummen *et al.* [14], Granjal *et al.* [38], and Kang *et al.* [46]. However, we distinguish the following major advantages offered by our scheme compared to their approaches. We believe that the approaches presented by Granjal *et al.* [38] and Kang *et al.* [46] do not provide comprehensive end-to-end security. Rather, they can be considered *semi end-to-end* security. The main reason is that in these works, the 6LoWPAN Borader Router (6LBR) acts as an intermediary node located between the sensor and the end-user. Every time these two end-points try to communicate with each other, all the secret information related to the communication needs to pass through the 6LBR. Whilst, the smart gateway utilized in our work is only used during the initialization phase (Figure 5), and then afterwards, both end-points directly communicate with each other through a channel secured by the DTLS session resumption. Therefore, *end-to-end* security is guaranteed in our work.

The approaches presented by Granjal *et al.* [38] and Kang *et al.* [46] also lack scalability and reliability as their proposed system architectures rely on the centralized 6LBR. The main reason is that their proposed architectures cannot be extended to be utilized in multi-domain infrastructures, such as large hospital environments. For example, if a malicious adversary performs a DoS attack or compromises the 6LBR, a large quantity of stored information concerning the constrained domain can be retrieved. More precisely, in multi-domain networks, a DoS attack can disrupt all the available medical networks as the functionality of the IoT-based healthcare system still depends on the centralized 6LBR. However, these issues are solved in our proposed scheme as the architecture is distributed. To be more specific, in our scheme, in a multi-domain smart home/hospital environment, if an attacker runs a DoS attack or compromises one of the smart gateways, only the associated medical sub-domain is disrupted. Although Hummen *et al.*'s [14] proposed delegation-based architecture offers end-to-end security, it is still not secure against the DoS attack due to the use of a centralized delegation server. Their presented architecture also suffers from shortcomings in scalability and reliability which is mainly due to the reasons mentioned above.

Based on the discussion above, our proposed scheme fulfills the aforementioned requirements of secure and efficient communication for healthcare IoT systems and can efficiently provide end-to-end security.

8. Conclusions

We presented an end-to-end security scheme for mobility enabled healthcare IoT systems. Based on literature, we determined that our scheme has the most extensive set of security features in comparison to related approaches. Our three-tier system architecture consists of the device layer, the fog layer, and the cloud layer. We leveraged the strategic position and the distributed nature of smart gateways in the fog layer to provide seamless mobility for medical sensors and to alleviate the sensors' processing loads. In our scheme, ubiquitous mobility is possible without requiring any reconfiguration at the device layer. The end-to-end security scheme was specified and designed by employing the certificate-based DTLS handshake between end-users and smart gateways as well as utilizing the session resumption technique. Our testbed platform demonstration showed that, compared to existing end-to-end security approaches, our scheme reduces the communication overhead by 26% and the communication latency between smart gateways and end users by 16%. Our scheme performed approximately 97% faster than certificate-based and 10% faster than symmetric key-based DTLS. In terms of memory requirements, certificate-based DTLS consumes about 2.2 times more RAM and 2.9 times more ROM resources than our approach. In fact, the RAM and ROM requirements of our scheme are almost as low as in symmetric key-based DTLS. Taking into account that the handover latency caused by mobility is low and the handover process does not incur any processing or communication overhead on the sensors, we summarize that our scheme is a very promising solution for ensuring end-to-end security and secure ubiquitous sensor-level mobility for healthcare IoT.

Acknowledgment

The authors wish to acknowledge the financial support by the Finnish Cultural Foundation, HPY Foundation, Nokia Foundation, Ulla Tuominen Foundation, and University of Turku Graduate School (UTUGS) during the course of this project.

References

[1] European Commission Information Society. Internet of Things Strategic Research Roadmap, 2009.

[2] L. Da Xu, W. He, and S. Li. Internet of Things in Industries: A Survey. *Industrial Informatics, IEEE Transactions on*, 10(4):2233–2243, 2014.

[3] S. Li, L. Da Xu, and S. Zhao. The Internet of Things: A Survey. *Information Systems Frontiers*, 17(2):243–259, 2015.

[4] A.-M. Rahmani, N.K. Thanigaivelan, Tuan Nguyen Gia, J. Granados, B. Negash, P. Liljeberg, and H. Tenhunen. Smart e-Health Gateway: Bringing Intelligence to IoT-Based Ubiquitous Healthcare Systems. In *12th Annual IEEE Consumer Communications and Networking Conference*, pages 826–834, 2015.

[5] C.E. Koop, R. Mosher, L. Kun, J. Geiling, E. Grigg, S. Long, C. Macedonia, R. Merrell, R. Satava, and J. Rosen. Future Delivery of Health Care: Cybercare. *IEEE Engineering in Medicine and Biology Magazine*, 27(6):29–38, 2008.

[6] R. Mueller. Demo: A Generic Platform for Sensor Network Applications. In *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–3, 2007.

[7] W. Shen, Y. Xu, D. Xie, T. Zhang, and A. Johansson. Smart Border Routers for eHealthCare Wireless Sensor Networks. In *7th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, 2011.

[8] Intel® IoT Gateway, 2014. http://www.intel.com/content/products [accessed 2014-01-22].

[9] S. Kumar and C. Paar. Are Standards Compliant Elliptic Curve Cryptosystems Feasible on RFID? In *Workshop on RFID Security*, 2006.

[10] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu. Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services. *IEEE Transactions on Industrial Informatics*, 10(2):1578–1586, 2014.

[11] G. Yang, L. Xie, M. Mantysalo, X. Zhou, Z. Pang, L. Da Xu, S. Kao-Walter, Q. Chen, and L. Zheng. A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor, and Intelligent Medicine Box. *IEEE Transactions on Industrial Informatics*, 10(4):2180–2191, 2014.

[12] H. Yan, L. Da Xu, Z. Bi, Z. Pang, J. Zhang, and Y. Chen. An Emerging Technology – Wearable Wireless Sensor Networks With Applications in Human Health Condition Monitoring. *Journal of Management Analytics*, 2(2):121–137, 2015.

[13] K. Malasri and L. Wang. Addressing Security in Medical Sensor Networks. In *Proceedings of the 1st International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*, pages 7–12, 2007.

[14] R. Hummen, H. Shafagh, S. Raza, T. Voig, and K. Wehrle. Delegation-based Authentication and Authorization for IP-based Internet of Things. In *11th IEEE International Conference on Sensing, Communication, and Networking*, pages 284–292, 2014.

[15] X. Hung, M. Khalid, R. Sankar, and S. Lee. An Efficient Mutual Authentication and Access Control Scheme for WSN in Healthcare. *Journal of Networks*, 6(3):355–364, 2011.

[16] R. Chakravorty. MobiCare: A Programmable Service Architecture for Mobile Medical Care . In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2006.

[17] C. Bormann Z. Shelby, K. Hartke. Constrained Application Protocol (CoAP), draft-ietf-core-coap-18, IETF. 2013.

[18] N. Modadugu E. Rescorla. Datagram Transport Layer Security (DTLS) Version 1.2. In *RFC 5238*, 2012.

[19] S. Rahimi Moosavi, T. Nguyen Gia, A.M. Rahmani, E. Nigussie, S. Virtanen, J. Isoaho, and H. Tenhunen. SEA: A Secure and Efficient Authentication and Authorization Approach for IoT-Based Healthcare Systems Using Smart Gateways. In *The 6th International Conference on Ambient Systems, Networks and Technologies*, pages 452–459, 2015.

[20] S. Rahimi Moosavi, T. Nguyen Gia, E. Nigussie, A.M. Rahmani, S. Virtanen, H. Tenhunen, and J. Isoaho. Session Resumption-Based End-to-End Security for Healthcare Internet-of-Things. In *IEEE International Conference on Computer and Information Technology*, 2015.

[21] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. CodeBlue: An Ad hoc sensor Network Infrastructure for Emergency Medical Care. In *Wearable and Implantable Body Sensor Networks*, pages 12–14, 2004.

[22] K. Lorincz, D. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor Networks for Emergency Response: Challenges and Opportunities. *IEEE Pervasive Computing*, 3(4):16–23, 2004.

[23] N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[24] C. Karlof, N. Sastry, and D. Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 162–175, 2004.

[25] G. Kambourakis, E. Klaoudatou, and S. Gritzalis. Securing Medical Sensor Environments: The CodeBlue Framework Case. In *The Second*

International Conference onAvailability, Reliability and Security, pages 637–643, 2007.

[26] J. Ko, J. Lim, Y. Chen, R. Musvaloiu, A. Terzis, G. Masson, T. Gao, W. Destler, L. Selavo, and R. Dutton. MEDiSN: Medical Emergency Detection in Sensor Networks. *Association for Computing Machinery Transactions on Embedded Computing Systems*, 10:11:1–11:29, 2010.

[27] C. Tan, H. Wang, S. Zhong, and Q. Li. IBE-Lite: A Lightweight Identity-Based Cryptography for Body Sensor Networks. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):926–932, 2009.

[28] S. valenzuela, M. Chen, and V. Leung. Mobility Support For Health Monitoring at Home Using Wearable Sensors. *IEEE Transactions on Information Technology in Biomedicine*, 15(4):539–549, 2011.

[29] A. Jara, M. Zamora, and A. Skarmeta. An initial approach to support mobility in hospital wireless sensor networks based on 6LoWPAN (HWSN6). *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 1(2/3):107–122, 2010.

[30] A. Jara, M. Zamora, and A. Skarmeta. HWSN6: Hospital Wireless Sensor Networks Based on 6LoWPAN Technology: Mobility and Fault Tolerance Management. In *International Conference on Computational Science and Engineering*, volume 2, pages 879–884, Aug 2009.

[31] A. Jara, M. Zamora, and A. Skarmeta. Intra-mobility for Hospital Wireless Sensor Networks Based on 6LoWPAN. In *6th International Conference on Wireless and Mobile Communications*, pages 389–394, Sept 2010.

[32] H. Fotouhi, M. Alves, M. Zuniga Zamalloa, and A. Koubaa. Reliable and Fast Hand-Offs in Low-Power Wireless Networks. *IEEE Transactions on Mobile Computing*, 13(11):2620–2633, 2014.

[33] S. Li, L. Da Xu, and X. Wang. Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186, 2013.

[34] S. Li, L. Da Xu, and X. Wang. A Continuous Biomedical Signal Acquisition System Based on Compressed Sensing in Body Sensor Networks. *IEEE Transactions on Industrial Informatics*, 9(3):1764–1771, 2013.

[35] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog Computing and Its Role in The Internet of Things. In *Proceedings of the Workshop on Mobile Cloud Computing*, pages 13–16, 2012.

[36] J. Daemen and W. Rijmen. Specification of Rijndael. pages 31–50, 2002.

[37] R. Hummen and J. Gilder. Extended DTLS Session Resumption for Constrained Network Environments. Technical report, 2013.

[38] J. Granjal, E. Monteiro, and J. Sa Silva. End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication. In *International Conference on Networking*, pages 1–9, 2013.

[39] J. Caldeira, J. Rodrigues, and P. Lorenz. Intra-Mobility Support Solutions for Healthcare Wireless Sensor Networks, Handover Issues. *IEEE Sensors*, 13(11):4339–4348, 2013.

[40] PandaBoard Platform Information. http://pandaboard.org/ [accessed 2015-09-27].

[41] SmartRF06 Evaluation Board. http://www.ti.com/lit/ug/swru321a [accessed 2015-09-27].

[42] Arago Systems. Wismote. http://www.aragosystems.com/en/document-center [accessed 2015-09-27].

[43] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W.polk. Internet X.509 Public Key Infrastructure Certificate Profile. http://tools.ietf.org/html/rfc5280 [accessed 2015-09-27].

[44] O. Bergmann. TinyDTLS. http://sourceforge.net/p/tinydtls [accessed 2015-09-27].

[45] D. Aranha and C. Gouv. RELIC is an Efficient Library for Cryptography. http://code.google.com/p/relic-toolkit/ [accessed 2015-09-27].

[46] N. Kang, J. Park, H. Kwon, and S. Jung. ESSE: Efficient Secure Session Establishment for Internet-Integrated Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, pages 1–12, 2015.

[47] R. Hummen, J. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle. Towards Viable Certificate-based Authentication for the Internet of Things. In *Proceedings of the 2nd Workshop on Hot Topics on Wireless Network Security and Privacy*, pages 37–42, 2013.

Biographies

Sanaz Rahimi Moosavi: received her B.Sc. (Tech.) degree in Computer Software Engineering from the Department of Electrical and Computer Engineering, University of Imam Reza, Mashhad, Iran in 2006, and M.Sc. (Tech.) degree in Information Technology, Networked Systems Security from the Department of Information Technology and Communication Systems, University of Turku, Finland in 2013. She is currently working towards her Ph.D. degree at University of Turku, Finland. Her research interests include security and privacy, Internet of Things (IoT), smart healthcare systems, and lightweight cryptography techniques. She is a Student Member of IEEE.

Tuan Nguyen Gia: received his B.Sc. (Tech.) degree in Information technology from Department of Information Technology, Helsinki Metropolia University of Applied Sciences, Helsinki, Finland in 2012, and M.Sc. (Tech) degree in Information Technology, Embedded Computing from the Department of Information Technology and Communication Systems, University of Turku, Finland in 2014. He is currently working towards his Ph.D. degree at the University of Turku, Finland. His research interests include Internet of Things (IoT), Smart Healthcare, and Medical Cyber Physical System, FPGA and Wireless Body Sensor Networks.

Ethiopia Nigussie: Dr. Ethiopia Nigussie is a University Lecturer at the University of Turku, Finland. She obtained her PhD degree in Communication Systems from University of Turku in 2010 and M.Sc. degree in Electrical Engineering from Royal Institute of Technology (KTH), Sweden in 2004. Her current research interests are energy saving strategies, adaptive design approaches and security for low-power wireless networks, self-aware design, and cognitive radio networks. Dr. Nigussie is the author of "Variation Tolerant On-Chip Interconnects" book (Springer) and she has about

50 international peer-reviewed journal and conference articles. She is Senior Member of IEEE since March 2015.

Amir M. Rahmani: received his Master's degree from Department of Electrical and Computer Engineering, University of Tehran, Iran, in 2009 and Ph.D. degree from Department of Information Technology, University of Turku, Finland, in 2012. He also received his MBA jointly from Turku School of Economics and European Institute of Innovation Technology (EIT) ICT Labs, in 2014. He is currently a University Teacher (Faculty Member) at the University of Turku, Finland, and visiting researcher at KTH Royal Institute of Technology, Sweden. He is the author of more than 100 peer-reviewed publications, is supervising eight PhD students. He is currently co-leading three Academy of Finland projects entitled "MANAGE", "SPA", and "InterSys".

Seppo Virtanen: Dr. Seppo Virtanen received his M.Sc. in electronics and information technology in 1998 and D.Sc. (Tech.) in Communication Systems in 2004 from the University of Turku, Finland. Since 2009, he has been an adjunct professor of Embedded Communication Systems at University of Turku where he also heads the Master's Programme in Information Security and Cryptography. He is a senior member of the IEEE. Currently the focus in his research is on information security issues in the communication and network technology domain, specifically focusing on design and methodological aspects of reliable and secure communication systems and networks.

Hannu Tenhunen: received the diplomas from the Helsinki University of Technology, Finland, 1982, and the PhD degree from Cornell University, Ithaca, NY, 1986. In 1985, he joined the Signal Processing Laboratory, Tampere University of Technology, Finland, as an associate professor and later served as a professor and department director. Since 1992, ha has been a professor at the Royal Institute of Technology (KTH), Sweden, where he also served as a dean. He has more than 600 reviewed publications and 16 patents internationality. He is a member of the IEEE.

Jouni Isoaho: received his M.Sc. (Tech.) in Electrical Engineering, and his Lic. Tech. and Dr. Tech. in signal processing from Tampere University of Technology, Finland in 1989, 1992 and 1995, respectively. Since 1999 he has been the professor of communication systems at University of

Turku, Finland, where he heads the communication systems laboratory. His research interests include future communication system concepts, applications and implementation techniques. His current special interests are in dynamically reconfigurable self-aware systems for future communication and interdisciplinary applications including information security and dependability aspects.