IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 20, 2020, accepted March 15, 2020, date of publication March 24, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983117

Lightweight Authenticated-Encryption Scheme for Internet of Things Based on Publish-Subscribe Communication

ABEBE DIRO[1], HAFTU REDA[1], NAVEEN CHILAMKURTI[1], (Member, IEEE), ABDUN MAHMOOD[1], (Member, IEEE), NOOR ZAMAN[2], (Member, IEEE), AND YUNYOUNG NAM[3], (Member, IEEE)

[1]Department of Computer Science and IT, La Trobe University, Bundoora, VIC 3083, Australia
[2]School of Computer Science and Engineering (SCE), Taylor's University, Selangor 47500, Malaysia
[3]Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Yunyoung Nam (ynam@sch.ac.kr)

This work was supported in part by the Korea Institute for Advancement of Technology (KIAT) funded by the Korea Government (MOTIE: Ministry of Trade Industry and Energy) under Grant N0001791, HRD Program for ICT Convergence Smart Rehabilitation Industrial Education Program, and in part by the Soonchunhyang University Research Fund.

ABSTRACT The resource-constrained nature and large-scale adoption of Internet of Things (IoT) have a significant challenge for securing IoT applications. This necessitates a robust and lightweight security architecture and schemes as the existing traditional Internet security architecture and protocols require huge resources and lack of end-to-end security mechanism. In this research, a resource efficient end-to-end security scheme has been proposed by offloading computations and storage of security parameters to fog nodes in the vicinity. In addition, a symmetric-key payload encryption has been used to minimize the overhead of message communication in the resource-contested IoT environment. The analysis shows that the proposed scheme outperforms Transport Layer Security (TLS) in resource usage while it maintains equivalent authenticated end-to-end communication between communicating IoT nodes. The proposed end-to-end security scheme saves more communication bandwidth and incurs less overhead as compared to existing TLS-based security schemes. In particular, the proposed system uses less number of handshakes and achieves a decrease in the number of transmitted messages (approximately 184 bytes as compared to compared TSL message size of 332 bytes) for every handshake. Further, it has been demonstrated through experiments that the proposed security method incurs less overheads as compared to the TLS bandwidth consumption considering a single connection session during message subscription.

INDEX TERMS Encryption, authentication, cybersecurity, end-end security, Internet of Things, publish-subscribe systems.

I. INTRODUCTION

The Internet of Things (IoT) cannot adopt standard Internet architecture and protocol standards mainly due to scalability issues and the limited resources. A novel architecture and lightweight security protocol has to be adopted to minimize communication, computation and storage overheads in the IoT network. One of the most promising architectures is the publish-subscribe model [19], [20] in which clients exchange messages through a broker. The broker hosts topics that can be published and subscribed to by clients. Fig. 1 shows a typical architecture of the publish-subscribe computing model. Message Queue Telemetry Transport (MQTT) is a popular protocol of publish-subscribe applications that make use of IoT devices. It is a lightweight protocol envisioned to decouple the publisher and subscriber in machine-to-machine communications. Despite the recommendations of the MQTT standard, the protocol still lacks security considerations with its current implementations. Thus, IoT applications significantly benefit from the publish-subscribe architecture and protocols if security is taken into consideration.

Traditional security protocols, if implemented, require expensive computation of asymmetric cryptography and Public Key Infrastructure certificates before authentication occurs in publish-subscribe IoT arenas [32]. For this reason, there must be a lightweight authentication and encryption scheme to protect topics and the associated data. Furthermore, traditional protocols such as Transport Layer Security

The associate editor coordinating the review of this manuscript and approving it for publication was Tie Qiu.

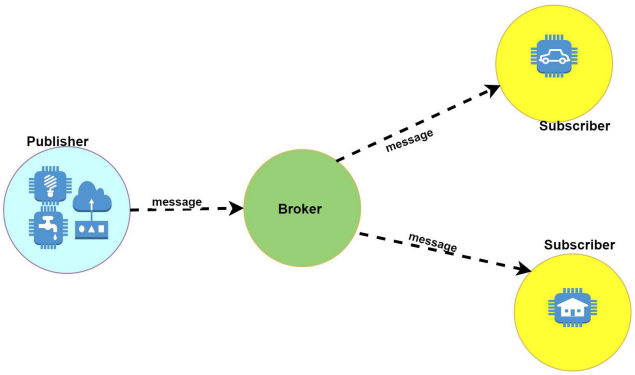

FIGURE 1. Publish-subscribe system.

(TLS) cannot be applied as a mechanism of an end-to-end security in decoupled systems. Even if the channel between publisher and broker or broker and subscriber can be secured, it is highly probable that the broker is left unsecured. This means that it is essential that payloads are encrypted before passing through a broker to guarantee end-to-end security. To provide complete security, authentication must be provisioned with a secure key exchange. As much as possible, key management and other cryptography computations should be offloaded to a registration authority or nearby nodes in the IoT environment. This inculcates that IoT devices cannot be effectively secured by traditional security protocols due to its extensive resource requirements. Thus, lightweight security schemes, specifically authenticated-encryption, that minimize storage, processing and communication overheads are essential to secure massive-scale deployment of the IoT.

The IoT is expected to enable large number of emerging technologies, for instance smart health, smart grid, smart city, smart agriculture, and the like. These scenarios require a secure end-to-end communication whose design considerations should be subject to a communication bandwidth bottleneck, computation resource, memory, and energy consumption. In this paper, an end-to-end security scheme is proposed in a resource-constrained IoT environment using a publish-subscribe communication system.

The main motivation of this paper is the resource constraints, in terms of communication overhead, computational complexity, and impact on storage, while considering an end-to-end security scheme for IoT devices over the publish-subscribe communication system. The staggering number of IoT nodes look for a secure communication channel. Existing research works on TLS/Datagram TLS (DTLS) provide successful end-to-end authentication thereby fulfilling confidentiality-and integrity-protected data exchange between the end points. However, such schemes are not good fit to resource-constrained IoT communications. The proposed security scheme incurs less communication overhead between the IoT end points, less memory, and less computation resource while providing an equivalent security solution compared to the TLS/DTLS security solutions.

Considering IoT environment, the end-to-end communication channel should be secured to provide data availability, data integrity, and user confidentiality subject to minimization of resources such as memory storage, communication overhead, and security computational complexity. To achieve this, we propose end-to-end security scheme for IoT devices considering publish-subscribe communication paradigm. The main contributions of this paper focuses on the end-to-end authentication scheme subject to resource constraints. Our main contributions are summarised as follows:

1) As compared to existing TLS-based security schemes, the proposed system shows a performance analysis in terms of:
   a) saving communication bandwidth of lightweight IoT nodes,
   b) a decrease in the number of communication handshakes, and
   c) a decrease in the message sizes in each handshake.
2) The proposed system enables less memory consumption for hardware-constrained IoT devices by offloading storage overheads to Fog brokers. For example, in TLS IoT clients need a memory storage of 321 bytes of credentials; however, only 80 bytes of memory is required in our proposed scheme.
3) Our proposed end-to-end security scheme also reduces the computational overheads, which is a key requirement for various IoT applications.

The remaining part of of this article is organised as follows. In Section II, we discuss the IoT and its communication protocol stack. The architecture of IoT considering different scenarios is discussed in Section III. In the following section, a number of applications of the IoT is discussed. Moreover, Section V discusses vulnerabilities of the IoT over the different communication layers. Then, existing literature related to security solutions for the IoT is presented in Section VI also comparing the advantages of our proposed technique with the current state-of-the-art researches. Further, the proposed security model is covered in Section VII. Here, details of the proposed security architecture, proposed algorithm of the end-to-end security, and performance analysis of the proposed scheme will be presented. Finally, conclusion is given in Section VIII.

II. THE IoT PROTOCOL STACK

As the Internet is the key platform for IoT devices to function, the communication protocol stack in the IoT environment should be compatible with the Internet Protocol (IP) stack. IoT devices typically communicate using the IP protocol stack, which requires extensive power, storage and computing resources. Protocols such as Bluetooth, RFID, and NFC can be used to conserve power though their communication range is limited to personal area networks (PAN). The outreach of these PANs, however, can be increased by modifying the existing IP stack to enable lower power consumption. In line with this, several energy conserving and low resource requirement protocols have been envisioned for IoT devices at all layers of the Transmission Control Protocol (TCP)/IP stack

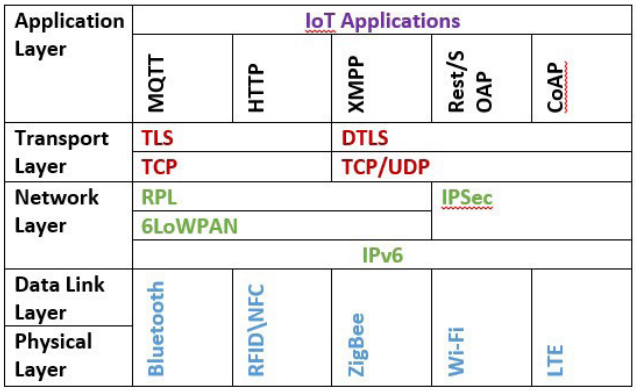

FIGURE 2. Protocol stack of the IoT [22].

of the Internet [1]. The total list of the IoT protocol stack can be seen in Fig. 2.

Low power and resource conserving network protocols play a significant role in addressing and routing functions of the IoT network [21]. One of the network layer protocols adopted by the IP stack to lower power consumption is the IPv6 over Low-Power Wireless PAN (6LoWPAN) standard, which integrates IPv6 with low power PANs [2]. This enables having a PAN as wide as a local area network (LAN) with improved power efficiency. The 6LoWPAN standard provides a large address space and defines a layer named adaptation layer between link layer and the transport layer to enable communication with the Internet. Routing protocol for low-power and lossy networks over IPv6 (RPL) is another low power network layer routing protocol based on distance vectors. The objective of this protocol is to minimize routing latency and memory requirements and save battery power. Thus, the surge in the number of connected entities could not have been imagined without lightweight network layer protocols.

Resource-efficient application protocols are also building blocks of message exchange and data delivery in the IoT environment. Application layer protocols such as Constrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP) and MQTT have been designed for messaging purposes [22]. CoAP is an optimized version of HTTP for resource management. For instance, it uses the EXI data format for storage instead of HTML/XML for efficiency. MQTT is a lightweight publish-subscribe protocol that uses a broker to decouple subscribers and publishers. The brokers handle subscription and authentication using topics. XMPP is a communications protocol based on XML that supports multiple communication patterns, including asynchronous messaging, publish-subscribe and request/response. Therefore, data communication requires optimized Internet application protocols or newly emerging resource-saving protocols.

III. ARCHITECTURAL SUPPORT FOR THE IoT

In the IoT-to-cloud continuum, the architecture of IoT varies from application to application. The basic architecture of the IoT consists of three-tiers: things, fog, and cloud [3]. Tier-1 comprises smart objects capable of data capturing,

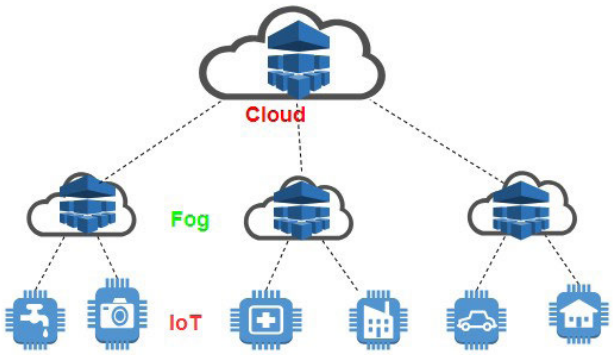

FIGURE 3. Basic architecture of fog network.

processing, and communication. This consists of IoT devices, typically wireless sensors and actuators. The second tier is a fog layer in which gateways and access points are the main components that collaborate in sharing resources such as storage and processing. In this level, intermediate operations are performed before the data moves to the cloud. The final tier is the cloud data centre which has huge storage and processing capabilities [4]. [5], [6] also discusses similar layers, namely, the perception, network and application layers, which correspond to the previous proposal by [3] as things, fog and cloud layers, respectively. Fig 3 shows the typical architecture of fog: interaction between IoT, fog nodes, and the cloud. It follows that the fog network supports IoT devices in the continuum of things-to-cloud.

Authors in [3] proposed a 4-tier IoT architecture: embedded systems and sensors, multi- service edge, core and the cloud/data centre as the information and computing architecture to support IoT services. According to this view, smart things consisting of sensors and actuators are located in the lowest level of the hierarchy while Wi-Fi and cellular nodes serve as distributed fog nodes. The architecture views the cloud as the data store. From the big data processing point of view, IoT architecture can be lined up with 4-level hierarchies comprising grid sensors and devices, operational or non-operational data, historical data and business data repository [7]. Based on hardware/software components, [8] proposed a 3-tier view of IoT architecture: physical resources/abstraction layer, fog orchestration layer, and IoT services. Despite its different perspective, these classification schemes also aim at supporting IoT devices.

As can be observed from the various classification, the three-level architectural model corresponding to the network perspective is the most widely accepted approach. It is aimed at deploying fog-level services for IoT applications for a quick response time and scalability while the cloud-level deals with intelligence and high-performance computing. This architecture is ideal for data analysis, security controls and network monitoring for latency-sensitive critical applications.

IV. APPLICATIONS OF THE IoT

The IoT has wide applications in today's modern world, particularly smart cities, smart homes, eHealth, and smart grids.

Smart cities benefit from IoT applications in the sectors of smart transportation and smart water systems. For instance, the CityOS project of Barcelona aimed at creating a single virtualized OS for all the smart city applications and services offered using IoT devices. Congestion [9] and accident [10] management are the two most important such use cases of applying IoT in the area of transportation in cities. On the other hand, the efficient utilization of water can be ensured by embedding sensors in water drains, tanks and supply lines for leakage management [11]. This smart water management can be integrated with weather and usage information as demonstrated in cities such as Barcelona and Stockholm. As a result, IoT applications can increase the efficient use of utilities in smart cities.

Smart home [12] is another application area that can harness the monitoring and control potential of IoT devices for utility and safety. Such connectivity using IoT devices has become popular due to advances in sensor and actuation technologies, and the desire of users to enhance their quality of life through technology. For this reason, IoT devices are used in smart homes by automating routine tasks, and conserving energy by automatically turning off lights in the absence of activity. Video surveillance and motion detection are used as a means of securing homes by recording and alerting events taking place in the vicinity [13]. Therefore, the use of IoT devices for home automation serves for the purpose of monitoring and security, and resource efficiency.

IoT systems have also shown prominent importance in the healthcare systems and wellness domains [33]. For instance, wearable devices monitor a patient's health condition, and can report to doctors and parents. This is particularly important for falling detection due to chronic diseases and aging, and reporting case of emergencies. It is extremely useful to embed IoT devices into medical devices to continuously monitor, record and regulate health conditions and transmit status information such as warnings and suggestions. The fitness of humans can be monitored using IoT devices based on daily activities. For example, the number of steps taken and the amount of exercises done can be measured by using fitness trackers [14]. This suggests that IoT applications will revolutionize health-care industries in monitoring and fitness activities.

The generation, transmission, distribution and consumption of electricity has recently moved online to enable the flow of power between clients and suppliers. In this case, the emerging use of smart Grids helps save energy consumption in smart homes by having a smart meter for monitoring energy consumption, and provides dynamic pricing for customers. Smart grids can also integrate IoT devices to monitor online transmission lines for disaster prevention. Smart meters can predict power usage by analyzing the consumption patterns at regular and peak load times. This information is used by clients to adjust their power consumption to reduce costs [15]. Hence, the integration of IoT devices into smart grids provides efficiency in terms of saving power and detecting faults.

V. VULNERABILITIES OF THE IoT

Traditional Internet attack threats and their variants continue to be the major threats for the IoT [16]. The focus of Internet attacks is data manipulation whereas IoT attacks target controlling actuation. With limited protection, the scale and simplicity of attack targets are larger for IoT devices than traditional Internet devices.

In network security, a vulnerability is a point of weakness through which a threat actor, such as an attacker performs unauthorized actions within a network. The origin of the attacks could be external adversaries that intend to gain access to the internal network or insiders that have the motive and opportunity to misuse, attack or steal information. The impact and treatment of IoT vulnerabilities is different from the traditional Internet due to resource limitations.

The vulnerability of IoT devices can be at the network, device, interface or infrastructure level [17]. In this section, the vulnerabilities of IoT devices are discussed based on the OWASP list [18].

An insecure web interface can expose web-interfaced IoT devices to attackers in gaining unauthorised access to the device. These vulnerabilities include enumeration of accounts, weak and default parameters, credentials exposed in network traffic, cross-site scripting (XSS), SQL injection, session management and weak account lock settings.

Insecure network services is related to weaknesses in network services that enable to access the IoT applications, and that has a potential flaw to allow an intruder to gain unauthorised access to the device or associated data. Some of the vulnerabilities include flawed services, buffer-overflow, open-ports, exploitable UDP services, DoS, and DoS via network device fuzzing.

Insufficient authentication/authorisation represents ineffective authentication mechanisms being in place for authenticating IoT devices or authorisation mechanisms that lead to a higher level of privileged access than that allowed for users. For instance, lack of a strong password, unprotected device credentials, lack of multi-factor authentication, insecure credential such as password recovery, privilege escalation and lack of role-based access control.

Lack of transport encryption deals with plain data exchange with the IoT device that enables to intercept the data or the device. These specific weaknesses include plaintext services via the Internet, plaintext services via LAN, poorly implemented and configured Secure Sockets Layer (SSL)/TLS.

Privacy concerns are related to a lack of proper protection of personal or sensitive data during data collection. Insecure cloud interface is the vulnerability of the cloud interface at the point it meets/interacts with the IoT device. It can be poor authentication mechanisms or unencrypted data in transit, which can allow attackers access to the device or the underlying data. It tends to be similar to an insecure web interface, and the specific vulnerabilities are account enumeration, lack of account lockout and credentials exposed in network traffic.

Insecure mobile interface is similar to the insecure cloud interface regarding specific vulnerabilities. It is caused by lack of strong authentication or encryption which leads to an unauthorized access to the device or underlying data of an IoT device that is connected through vulnerable mobile interfaces.

Insufficient security configurability occurs when a user lacks the ability to alter security controls. For instance, the lack of a strong password enforcement or the lack of options for creating granular user permissions at the web interface can lead to the compromise of an IoT device and allow unauthorised access to the device or data. The particular vulnerabilities of this category are lack of layed permission, lack of strong password security, lack of monitoring and lack of logging.

Insecure software/firmware to the lack of an update capability when vulnerabilities are discovered. It can also include the insecurity of software/firmware updates when attackers target the updated files, and the underlying network connection. It is also possible that software/firmware can be insecure if sensitive data such as credentials are hard-coded into them. The challenges in the inability of a software/firmware to be updated leads to the state that the IoT devices remain vulnerable indefinitely to the security issue that the update is supposed to address. Further, if the devices contain hard-coded sensitive credentials, and if these credentials are exposed, then they remain so for an indefinite period of time. The specific vulnerabilities of this issue include lack of encryption when updates are retrieved, lack of an encrypted update file, lack of an update verification before retrieval and sensitive information contained in the firmware.

Poor physical security is a vulnerability which occurs when an attacker can disassemble a device to easily access the storage medium and any data stored on that medium. Weaknesses are also present when USB ports or other external ports can be used to access the device using features intended for configuration or maintenance.

VI. LITERATURE REVIEW

A security framework was proposed in [22] using password-only authentication and a key exchange scheme over the MQTT protocol. The protocol enables the establishment of session keys between a client and a broker. The obtained session key is used to encrypt data using a secured symmetric-key encryption algorithm. The protocol simplifies the burden of message exchange when compared to TLS. However, the scheme does not achieve end-to-end security and still involves expensive exponent computations.

A lightweight attribute-based encryption (ABE) over elliptic curves was used in [23]. This scheme achieved end-to-end encryption and access control, enabling only subscribed clients to decrypt the publisher's message. The data exchange among clients is protected from adversaries and the broker. Nevertheless, the broadcast nature of the protocol and the complex multiplicative inverse functions significantly increase the processing overhead.

The proposal in [24] discusses an end-to-end security mechanism of the MQTT protocol using a certification authority (CA). The authors used an RSA hybrid algorithm for performance gain. RSA has been used to speed up encryption while ECDSA has been employed for the digital signature. Though end-to-end encryption has been achieved, its overheads are not much less than that of TLS.

An authentication mechanism on RFID was proposed by the work in [25] using asymmetric security. The study considered the optimised design of NTRU as the fastest and most lightweight cryptographic mechanism because RFID is an extremely low-powered and resource-limited technology. The complex operations of cryptography are offloaded to a server, and the device only has to deal with the lightweight operations. The authors claim the system resists common attacks such as replay and man-in-the-middle attacks. This is an attractive approach to designing the security of constrained devices, such as low-powered IoT devices.

The work in [26] analyzed AES encryption techniques over the MQTT protocol. The research compared payload encryption and link layer encryption for time and memory efficiency. Furthermore, AES encryption modes with and without an accompanying MAC were investigated. It was found that link encryption is more efficient than end-to-end encryption. However, it cannot provide application-layer end-to-end encryption. The authentication scheme proposed by the authors is resource intensive.

A secure group communication scheme was proposed for publish-subscribe systems in [27]. Group-oriented key management in the secure communication of events in the publish- subscribe systems was also analyzed in [28]. This scheme exponentially increases the keys in heterogeneous and massive-scale IoT systems, hence it is a source of scalability bottleneck. It is also not as flexible as publish-subscribe systems.

Reference [29] investigated the security challenges in a content-based publish-subscribe system. Research work in [30] also showed the possibility of achieving security requirements in publish-subscribe networks. As these studies focus on the general Internet, their proposed methods are not suitable for the resource-constrained IoT.

A lightweight mutual authentication and payload encryption scheme was proposed in [31] for use over the CoAP protocol using a four-way handshake mechanism. It is a client-server model which enhances the security of Datagram Transport Layer Security (DTLS) by integrating security features into CoAP. The objective was to design an authentication process that was as lightweight as possible in the number of request/response message exchanges when the session key is shared among interacting entities. The study showed that the protocol significantly improved DTLS in handshake duration, memory consumption and average response time.

Our proposed security scheme achieves a robust end-to-end security solution for the IoT nodes interacting over the publish-subscribe model subject to

TABLE 1. Comparison of our proposed system with previous works.

Comparison attributes	Our proposed system	Previous works
Basic security requirements	Considered	Considered
Processing efficiency	Considered	Considered
Communication overhead	Our proposed scheme has better communication efficiency (see 9)	-
Storage overhead	Our proposed scheme has better storage efficiency (see 8)	-

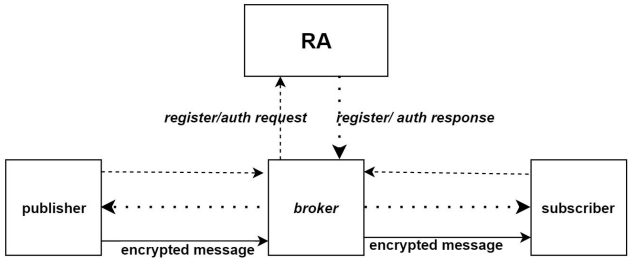

FIGURE 4. Security architecture for publish-subscribe system.

hardware-constraints. Generally, it is compelling that the proposed security scheme is suitable for IoT applications which seek for a secure end-to-end communications channel under affordable computation-and hardware-resources.

VII. PUBLISH-SUBSCRIBE END-TO-END SECURITY SCHEME

This section discusses Elliptic-curve cryptography (ECC) based publish-subscribe authentication schemes for IoT fog computing. We discuss security architecture, algorithms, analysis, and evaluations.

A. ARCHITECTURE

The architecture consists of publisher IoT, subscriber IoT, and the broker.

Publishers write data under a topic while subscribers read data from the topic to which they subscribe. In the scheme, the broker plays a paramount role in performing subscriptions, publications and information dissemination under a specific topic. The broker also handles authentication while the RA provides registration and secret key management services for subscribers and publishers. Credentials such as password, private key, and topics are provided by the RA at registration time. IoT devices are authenticated using the credentials received at the registration phase at the broker. The header of the MQTT packet is sent without encryption to provide packet routing information so that decryption is avoided at each intermediate node. The overall logical architecture of the proposed framework is depicted in Fig 4. An efficient and secure distributed architecture is designed for IoT-fog computing over the MQTT protocol. Hence, the design is aimed at decreasing processing, storage and message transmission from IoT devices while providing end-to-end security.

B. ALGORITHMS

The system provides authentication and encryption for data confidentiality and integrity in a resource-constrained environment. The encryption algorithm AES-CCM with 128-bits

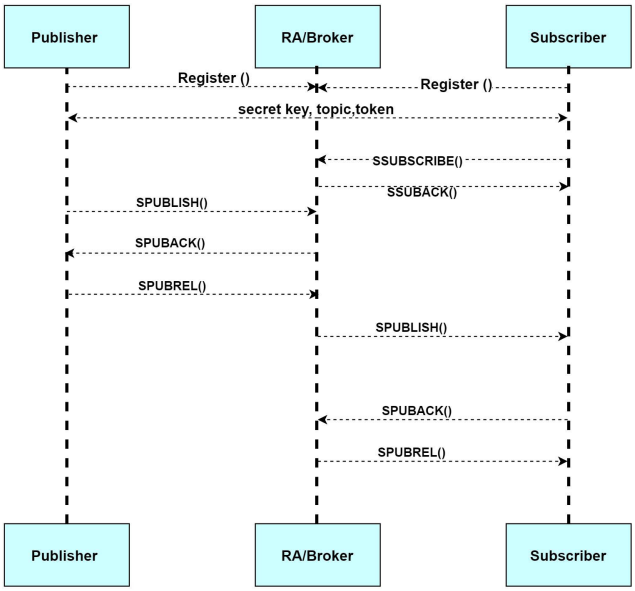

FIGURE 5. Interaction diagram of the system.

TABLE 2. Notations used in the ECC-based publish-subscribe authentication system.

Notation	Description
α	A point on EC
p	Large prime number
k	Random master secret key for a given IoT device
ID_i	Identity of device i
ID_b	Identity of broker b
$topic_i$	Subscription topic
K_i	Client-side Key for device i
K_b	Broker-side Key for device i
r_i, r_j, r_b	Nonce of publisher, subscriber and broker, respectively
$\|\|$	Concatenation
*	Changed by adversary
k_e	Encryption/decryption key
C	Comparison
H()	Hash function
token	Access token produced from registration attributes
$\oplus$	XOR operator
PM	Point multiplication
AES	Advanced Encryption standard

key is chosen based on the trade-offs in security, overhead and performance. A smaller key size and a fewer number of rounds contribute to lower overheads and greater performance in terms of the cost of security. However, increasing the number of rounds does not necessary increase security linearly whereas it decreases the speed of encryption by 40% [34]. The breakable key size using the current technology is 80 bits for symmetric cryptography [35], which is sufficient to adopt 128 bits for the encryption process. To provide message integrity, CCM mode (CBC-MAC) is used for its reduced message size and lack of padding.

This section discusses the algorithms of our system which consists of three steps: initialisation, registration, secure subscription, and secure publication. The interaction diagram of the the proposed security technique is shown in Fig 5. Table 2 lists notations used in the algorithms.

Fixed Header (2 bytes)				
Message Type	DUP	QoS	Retain	Remaining Length
Variable Header (60 bytes)				
Attributes		Packet ID		Session ID
Security Header (60 bytes)				
H (ID, Topic)	IoT ID		Nonce	
Ciphettext (62 bytes)				
Key ciphertext		Message ciphertext		MAC

SPUBACK (142 bytes)

Fixed Header (2 bytes)				
Message Type	DUP	QoS		Remaining Length
Variable Header (40 bytes)				
Attributes		Packet ID		
Security Header (100 bytes)				
H (ID, topic, token)	IoT ID	Broker ID	Nonce1	Nonce2

PUBREL (138 bytes)

Fixed Header (2 bytes)				
Message Type	DUP	QoS		Remaining Length
Variable Header (40 bytes)				
Attributes		Packet ID		
Security Header (96 bytes)				
Cipher of ID and topic hash	IoT ID	Broker ID	Nonce1	Nonce2

FIGURE 6. Secured publication packet formats.

Procedure 1 Initialisation

Input: security parameter 1^n

Output: public elliptic curve parameters *pk*

1) Two prime numbers p and q are generated in such a way that $q = \frac{(p-1)}{2}$, $and |q| = n$.
2) A base curve point α is generated such that cyclic group G is the unique order q subgroup of $Z^*{}_q$
3) Broadcast public parameters pk = (G,α,q)

The security procedures make use of MQTT packets such as publish, subscribe, puback, suback, and pubrel. We adopt these packets to extend the headers and payloads. Accordingly, the secured versions of the packets are renamed Spublish, Spuback, Spubrel, Ssubscribe and Ssuback. The first three are used in publication while the last two are engaged in subscription. The secure publication and subscription packet formats are shown in Fig 6 and Fig 7, respectively.

C. SECURITY ANALYSIS

The main objective of the security framework is to provide a robust security mechanism for message exchange among IoT devices through a broker such as a fog node using the MQTT protocol.

The threat model is composed of authentication (publisher and subscriber), encryption and broker-mediated key exchange processes. While it is assumed that the fog node is semi-trusted, IoT devices should be given an appropriate access control for their subscriptions. In this section, security concepts and requirements are evaluated and proved, and the capacity to resist prevalent cyber attacks are explained.

SSubscribe (102 bytes)

Fixed Header (2 bytes)				
Message Type	DUP	QoS	Retain	Remaining Length
Variable Header (40 bytes)				
Attributes		Packet ID		
Security Header (60 bytes)				
H (ID, Topic)	IoT ID	Nonce		

SSUBACK (142 bytes)

Fixed Header (2 bytes)			
Message Type	DUP	QoS	Remaining Length
Variable Header (40 bytes)			
Attributes		Packet ID	
Security Header (100 bytes)			
H (ID, topic, token)	IoT ID	Broker ID	Nonce1	Nonce2

FIGURE 7. Secured subscription packet formats.

Procedure 2 Registration

Input: Identity of IoT device ID_i and attributes A = $\{a_1, a_2, a_3, \ldots, a_n\}$
Output: IoT secret key k_i, $topic_i$, $token$

1) Publisher/Subscriber presents identity ID_i and attributes A to RA..
2) Then, RA performs the following operations:
   - chooses a master secret key k uniformly at random from a field Z^*_q for IoT ID_i.
   - chooses a random IoT secret key k_i from a field Z_p
   - calculates broker side secret key as $k_b = k \oplus k_i$ for IoT ID_i.
   - securely provides $\{k_i, topic_i, token\}$ to IoT. It stores the credentials $\{ID_i, k_i, topic_i, token, k_b\}$ for IoT ID_i.

Procedure 3 Secure Subscription

Input: Identity of IoT device r_j, ID_j, $topic_j$
Output: subscription

1) Subscriber presents $(r_j, ID_j, H(ID_j, topic_j, r_j))$ to broker subscription.
2) Broker validates $H(ID_j, topic_j, r_j)$ from the received and stored parameters. Then, acknowledges by $(r_j, r_b, ID_j, ID_b, H(ID_j, topic_j, r_b))$.
3) Subscriber calculates $H(ID_j, topic_j, r_b)$ locally, and compares with the received digest. If it is valid, subscription is successful.

1) AUTHENTICATION

Man-in-middle and replay attacks are potential threats in wireless connected IoT environment. It is highly probable that interceptors or eavesdroppers can target message communications between the IoT and fog nodes, while these entities can also impersonate either of the parties by replaying communications used in the old session. A good scenario is that the fog node could be hijacked and the subsequent message communication with IoT devices is a potential attack. The mutual authentication between the IoT devices and the fog node is achieved through the established trust by the combination of random numbers, hashed credentials $(ID_i, topic_i)$ and the broker side key k_b supplied by the RA. Random numbers are used in each session as a nonce. A Man-in-the-middle attack is prevented using the MAC functionality of the AES-CCM encryption.

Suppose adversary A intercepts $SPublish = (r_i, ID_i, H(ID_i, topic_i, r_i), (r\alpha, k_e + rk_i\alpha), AES_{k_e}(m))$ message in the *secure publication* phase. If it wants to replay to the broker, upon receiving the message, the adversary has to produce message send $(r_i, ID_i, H(ID_i, topic^*_i, r_i), AES_{enc}(m^*), r\alpha, k_e + rk^*_i\alpha)$ to submit to the broker. The broker computes $H(ID_i, topic^*_i) \neq H(ID_i, topic_i)$, and fails to be authenticated as the strong hash function cannot be reversed. Thus, the adversary cannot get the broker side key k_b to continue in the encryption key exchange. The encryption key cannot

Procedure 4 Secure Publication

Input: Message m, secret key k_i, encryption key k_e
Output: client side ciphertext

1) Publisher presents $SPublish = (r_i, ID_i, H(ID_i, topic_i, r_i), (r\alpha, k_e + rk_i\alpha), AES_{k_e}(m))$ to the broker.
2) Broker checks the existence of $H(ID_i, topic_i, r_i)$ in the RA. If exists, it saves the sessions and:
   - computes *intermediate encryption for publisher*: $rk_b\alpha + rk_i\alpha + k_e = rk\alpha + k_e$
   - computes *intermediate encryption for subscriber*: $rk\alpha + k_e - rk_b\alpha = rk_j\alpha + k_e$
   - generates a session key s_i. Acknowledges the publisher by replying $PUBACK = (r_i, r_b, ID_i, ID_b, AES_{token}(H(ID_i, topic_i, r_i)||s_i))$.
3) Publisher decrypts $AES_{token}(H(ID_i, topic_i)||s_i)$ using the existing token to get a session key s_i. It then, validates the received credentials $H(ID_i, topic_i, r_i)$, and replies with $PUBREL = (r_i, r_b, ID_i, ID_b, AES_{s_i}(H(ID_i, topic_i, r_i)))$.
4) Broker sends $SPublish = (r_b, ID_b, H(ID_j, topic_j, r_b), (rk_j\alpha + k_e), AES_{k_e}(m))$ to subscribers with $topic_i$ and ID_i.
5) Subscriber decrypts key by computing $rk_j\alpha + k_e - rk_j\alpha = k_e$. Then, it decrypts the cipher $AES_{k_e}(m)$ using the key k_e. It generates a session key s_j, and acknowledges message reception by replying $PUBACK = (r_j, r_b, ID_j, ID_b, AES_{token}(H(ID_j, topic_j, r_b)||s_j))$ to Broker.
6) Broker decrypts $AES_{token}(H(ID_j, topic_j, r_b)||s_j)$using the existing token to get a session key s_j. It then, validates the received credentials $H(ID_j, topic_j, r_b)$, and replies by $PUBREL = (r_j, r_b, ID_j, ID_b, AES_{s_j}(H(ID_j, topic_j, r_b)))$.

be obtained from ECC point $(k_e + rk_i\alpha)$ due to the difficulty of breaking ECDLP. If it does not get an encryption key, it cannot decrypt the message encrypted by the AES algorithm. Similar procedures occur during the *secure subscription* phase. The combination of ECDLP, random number and a strong hash function increases the security of the system against the adversary. Hence, it is compelling that the proposed algorithm protects against replay and man-in-the-middle attacks, as the two communicating parties are identified.

2) CONFIDENTIALITY

The message is destined only to intended IoT devices. For in- stance, a patient being monitored by an IoT device should only disseminate the information on their heart condition via the for node to only subscribed stakeholders. All communicating entities should be registered, authenticated and verified before publishing or accessing any message. After authentication, the any message communication between the IoT and any fog node is encrypted, ensuring the confidentiality and privacy of the participating entities. The attack on the broker reveals no information about the clients as the credentials are stored securely on the RA.

TABLE 3. Message sizes of TLS.

Message	Size	Remarks
ClientHello	160-170 bytes	Depends on parametsr such as cipher suites, Client Hello extensions and session resumption
Session ID	132 bytes	
ServerHello	70-75 bytes	Varies with Server Hello extensions
Certificate	4 x(800-1500 bytes)	varies due to certificate chain and size. Sometimes 4 cert are needed
ClientKeyExchange	130 bytes	RSA cert is popular for this
ChangeCipherSpec	1byte	Fixed (2 times)
Finished	12 bytes	
TLS Record Header	5 bytes	
TLS Handshake Header	4bytes	

TABLE 4. Message sizes of our scheme.

Message	Size	Remarks
client ID	20 bytes	this size can accommodate massive number of IoT devices
packet ID	20 bytes	this size can cover huge number of IoT packets
k_i, k_b, k	20 bytes each	this can be sufficient to produce a strong key
k_e	16 bytes	
topic	16 bytes	the size of subscription topic
nonce	20 bytes	random number size
MAC	8 bytes	message authentication size
m	50 bytes	IoT device can send a message size between 50 and 100 bytes
hash	16 bytes	the size of hash function
token	20 bytes	the size of a token produced when a client is subscribed
Spublish	184 bytes	customized size of secure publish
Spuback	142 bytes	customized size of secure publish acknowledgement
Spubrel	138 bytes	customized size of secure publish release
Ssubscribe	102 bytes	customized size of secure subscribe
Ssuback	142 bytes	customized size of secure subscribe acknowledgement

D. PERFORMANCE ANALYSIS

The main advantages of the proposed scheme compared to the existing scheme is the reduction of handshake frequencies and message sizes in each handshake. It is impossible to achieve this by using TLS because of the number of handshakes required with larger message sizes. Since the TLS system has multiple variations of a handshake, the most widely used TLS handshake version of the protocol (13 handshakes) is considered for comparison. The variation in the size of message in each handshake, it is reasonable to take average approximations for overhead computation. Accordingly, Table 3 approximates the SSL/TLS message size in each handshake connection.

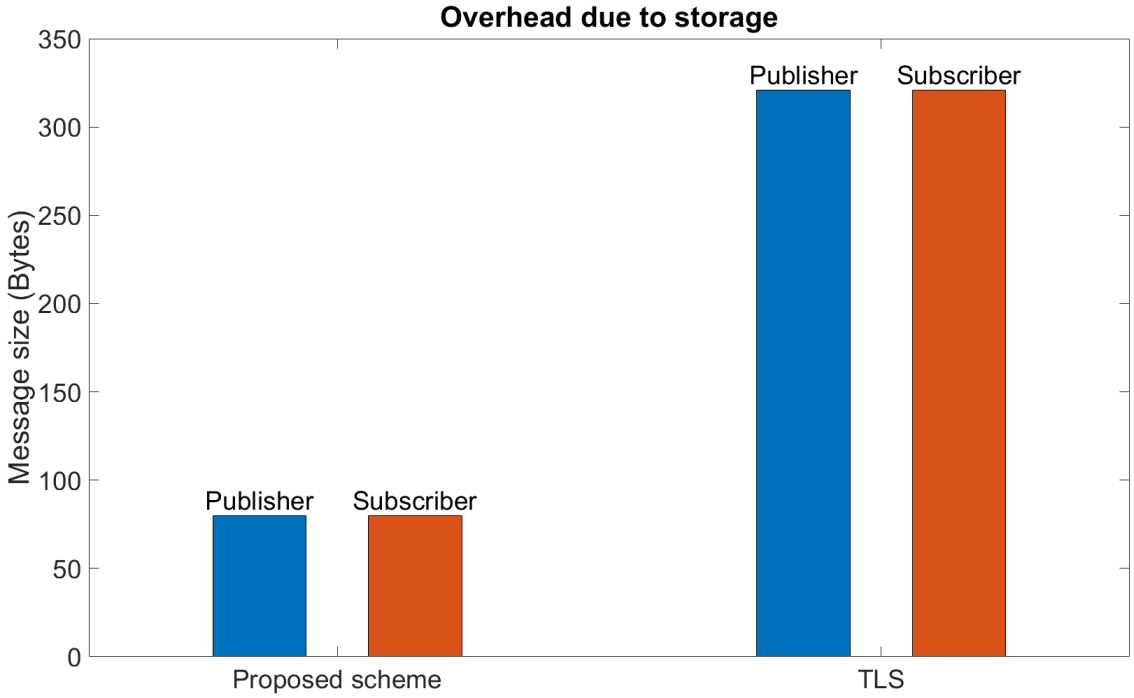

FIGURE 8. Overhead due to storage.

One of the important protocol evaluation is the extent to which the protocol offloads communication overheads from resource-constrained IoT devices. Table 3 shows the approximation of the average size of message in each handshake, and hence, the overhead of establishing a new TLS connection can be estimated from message sizes in each handshake, record header (5 bytes each) and handshake header (4 bytes each) messages. To established a new SSL/TLS connection, it requires 4 record headers (20 bytes) and 7 handshake headers (28 bytes). Hence, the overall overhead of a single new TLS connection is approximately $4 \times 5 + 7 \times 4 + 170 + 32 + 75 + 4 \times 1500 + 130 + 2 + 2 \times 12 = 6481$ bytes to the maximum (having considered 4 certificates, which are usually 1). However, if TLS is enabled with session resumption, the overhead declines to $15 + 16 + 202 + 75 + 2 + 24 = 332$ bytes. This applies for both publication and subscription. On the other hand, as shown in Table 4, our scheme incurs three handshakes namely, Spublish, Spuback and Spubrel which would be $184 + 142 + 138 = 464$ bytes without session resumption. However, TLS consumes about 6.5k bytes to set up a connection. Similarly, the proposed algorithm consumes 708 bytes of bandwidth for establishing a connection during subscription; however, TLS takes a bandwidth of over 6481 bytes. While TLS consumes 332 bytes of communication channel with session resumption enabled, our scheme incurs 184 bytes for the same process, as shown in the fig. 9. This means the system saves much communication bandwidth by reducing the number of handshakes compared to TLS. Thus, our system is more efficient than TLS systems.

TABLE 5. Performance comparisons between Our scheme and TLS with session resumption.

	Our Scheme		TLS	
Overhead Type	Pub	Sub	Pub	Sub
Storage	80 bytes	80 bytes	321 bytes	321 bytes
Comm	184 bytes	184 bytes	332 bytes	332 bytes

The resource scarcity in IoT ecosystems also calls for a solution that decreases the memory consumption of IoT devices. In TLS, clients store at least 321 bytes of credentials while the maximum storage space required in our protocol is about 80 bytes, as shown in fig. 8. While TLS imposes massive storage overheads, in the proposed algorithm, most of the storage overheads are offloaded to the fog broker or RA which are richer in resources than publisher/subscriber IoT devices.

Reducing computational overheads is also another essential requirement in the IoT environment as the devices lack processing capacity. For this reason, less expensive computations such as XOR, hash function and EC point additions have been used. The resource efficiency (storage, computations and communications) of the proposed scheme is manifested from Table 5 as it is superior to SSL/TLS. Further, Fig. 8 and Fig. 9 illustrate results of storage efficiency and communication efficiency respectively of our proposed system and previous TLS-based schemes.

In summary, a comparison of our scheme with similar studies in the literature is summarized in Table 6.

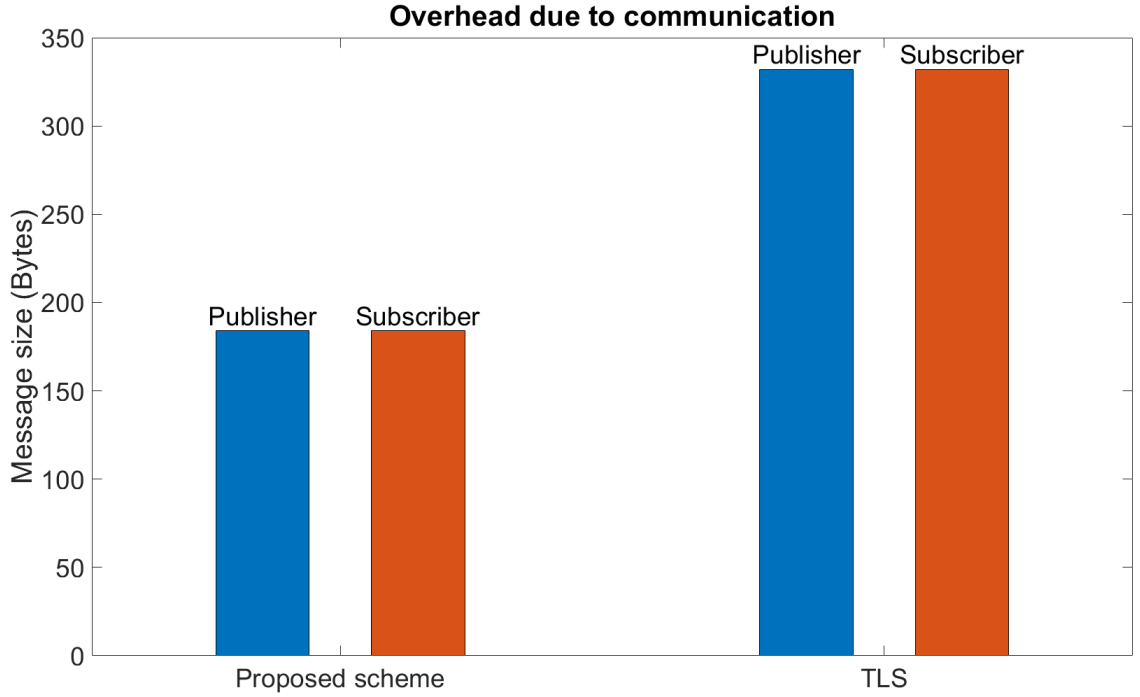

FIGURE 9. Overhead due to communication.

TABLE 6. Comparison of our scheme with the papers in the literature.

Attribute of comparison	Previous works	Our proposed scheme
Security	[29] considered security in terms of privacy of IoT services and encryption scheme for encrypting published events to protect data confidentiality	Considered lightweight security architecture and end-to-end security mechanism for IoT devices
Scalability	[30] considered scalbility from the perspective of routing	Considered scalability in resource-constrained IoT communications
Overhead related to communication bandwidth	Considered in existing techniques but performs less than our proposed scheme	Our proposed scheme outperforms the exiting techniques
Overhead related to storage	Considered in existing techniques but performs less than our proposed scheme	Our proposed scheme outperforms the exiting techniques

VIII. CONCLUSION

This article presented authenticated encryption mechanism based on publish-subscribe protocol for resource-constrained IoT environment. The proposed system uses a fog node as an intermediate broker in providing lightweight solution by offloading computations and storage for security parameters. In addition, while key exchange and authentication were achieved by public key cryptography (ECC), a private key encryption mechanism (AES-CCM) was used to minimize the overhead of message communication in the resource-constrained IoT network. The analytical comparison shows that the proposed scheme outperforms TLS in resource-usage and scalability while it maintains equivalent authenticated end-to-end communication between communicating IoT nodes. In particular, in each resumed session, the proposed system uses less number of handshakes, which enabled to achieve a decrease in the size of transmitted message (184 bytes) when it is compared TSL message size of 332 bytes. The same trend holds for the message storage incurred by the devices. This is an indication that ECC can be combined with symmetric algorithms to provide a lightweight security scheme for smart applications. In the future, implementing cryptographic elements on IoT platforms such as Raspberry Pi and Arduino will be considered for practical IoT security.

REFERENCES

[1] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the Internet of Things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.

[2] T. Gomes, F. Salgado, S. Pinto, J. Cabral, and A. Tavares, "A 6LoWPAN accelerator for Internet of Things endpoint devices," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 371–377, Feb. 2018.

[3] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018.

[4] G. Shanmugasundaram, V. Aswini, and G. Suganya, "A comprehensive review on cloud computing security," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Mar. 2017, pp. 1–5.

[5] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing architecture, evaluation, and future research directions," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 46–52, May 2018.

[6] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[7] S. Chen, T. Zhang, and W. Shi, "Fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 4–6, Mar./Apr. 2017.
[8] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1826–1857, 3rd Quart., 2018.
[9] A. Abhimane, P. Limkar, B. Barate, and V. G. Puranik, "IOT based vehicle traffic congestion control and monitoring system," in *Proc. 2nd Int. Conf. for Converg. Technol. (ICT)*, Apr. 2017, pp. 1220–1223.
[10] S. Javaid, A. Sufian, S. Pervaiz, and M. Tanveer, "Smart traffic management system using Internet of Things," in *Proc. 20th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2018, pp. 393–398.
[11] A. S, S. M N, A. S, K. Natarajan, K. R. Shobha, and A. Paventhan, "An IoT based 6LoWPAN enabled experiment for water management," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2015, pp. 1–6.
[12] S. El Jaouhari, "A secure design of WoT services for smart cities," Ph.D. dissertation, Dept. Netw. Internet Archit., Nat. School Mines-Telecom Atlantique, Nantes, France, 2018.
[13] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 426–434, Nov. 2017.
[14] A. H. Akpa, M. Fujiwara, H. Suwa, Y. Arakawa, and K. Yasumoto, "A smart glove to track fitness exercises by reading hand palm," *J. Sensors*, vol. 2019, May 2019, Art. no. 9320145.
[15] M. M. Rana, W. Xiang, E. Wang, and M. Jia, "IoT infrastructure and potential application to smart grid communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.
[16] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 180–187.
[17] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2014, pp. 1–8.
[18] OWASP. (2016). *Top IoT Vulnerabilities*. Accessed: Jul. 2018. [Online]. Available: https://www.owasp.org/index.php/Top-IoT- Vulnerabilities
[19] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, L. Skorin-Kapov, and R. Herzog, "Openiot: Open source Internet-of-Things in the cloud," in *Interoperabilityand Open-Source Solutions for the Internet of Things*. Cham, Switzerland: Springer, 2015, pp. 13–25.
[20] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for MQTT-based Internet of Things," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, May 2016, pp. 290–295.
[21] H. T. Reda, P. T. Daely, J. Kharel, and S. Y. Shin, "On the application of IoT: Meteorological information display system based on LoRa wireless communication," *IETE Tech. Rev.*, vol. 35, no. 3, pp. 256–265, May 2018.
[22] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in *Proc. 4th Int. Conf. Electr. Eng., Comput. Sci. Informat. (EECSI)*, Sep. 2017, pp. 1–6.
[23] W.-T. Su, W.-C. Chen, and C.-C. Chen, "An extensible and transparent Thing-to-Thing security enhancement for MQTT protocol in IoT environment," in *Proc. Global IoT Summit (GIoTS)*, Aarhus, Denmark, Jun. 2019, pp. 1–4.
[24] J. Prabaharan, A. Swamy, A. Sharma, K. N. Bharath, P. R. Mundra, and K. J. Mohammed, "Wireless home automation and security system using MQTT protocol," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 2043–2045.
[25] S. W. Park and I. Y. Lee, "Mutual authentication scheme based on lattice for NFC-PCM payment service environment," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 7, 2016, Art. no. 9471539.
[26] S. Katsikeas, K. Fysarakis, A. Miaoudakis, A. van Bemten, I. Askoxylakis, I. Papaefstathiou, and A. Plemenos, "Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 1193–1200.
[27] P. Porambage, A. Braeken, C. Schmitt, A. V. Gurtov, M. Ylianttila, and B. Stiller, "Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for iot applications.," *IEEE Access*, vol. 3, no. 1, pp. 1503–1511, 2015.
[28] W. Peng, S. Liu, K. Peng, J. Wang, and J. Liang, "A secure Publish/Subscribe protocol for Internet of Things using identity-based cryptography," in *Proc. 5th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dec. 2016, pp. 628–634.
[29] L. Duan, C.-A. Sun, Y. Zhang, W. Ni, and J. Chen, "A comprehensive security framework for Publish/Subscribe-based IoT services communication," *IEEE Access*, vol. 7, pp. 25989–26001, 2019.
[30] G. Siegemund and V. Turau, "A self-stabilizing Publish/Subscribe middleware for IoT applications," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 2, p. 12, Jun. 2018.
[31] M. A. Jan, F. Khan, M. Alam, and M. Usman, "A payload-based mutual authentication scheme for Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 1028–1039, Mar. 2019.
[32] S. Nolan. *Authenticated Payload Encryption Scheme for Internet of Things Systems over the MQTT Protocol*. Accessed: Apr. 2019. [Online]. Available: https://scss.tcd.ie/publications/theses/diss/2018/TCD-SCSS-DISSERTATION-2018-003.pdf
[33] J. Kharel, H. T. Reda, and S. Y. Shin, "Fog computing-based smart health monitoring system deploying LoRa wireless communication," *IETE Tech. Rev.*, vol. 36, no. 1, pp. 69–82, Jan. 2019.
[34] Y. Lindell and J. Katz, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: CRC Press, 2014.
[35] J. H. Kong, L.-M. Ang, and K. P. Seng, "A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments," *J. Netw. Comput. Appl.*, vol. 49, pp. 15–50, Mar. 2015.

ABEBE DIRO received the M.Sc. degree in computer science from Addis Ababa University, Ethiopia, in 2010, and the Ph.D. degree from the Department of IT Computer Science and IT, La Trobe University, Australia. From 2007 to 2013, he worked at Wollega University as the Director of ICT Development, and a Lecturer in computer science. His research interests include software-defined networking, the Internet of Things, cybersecurity, advanced networking, machine learning, and big data.

HAFTU REDA received the B.Sc. degree in electrical engineering from Bahir Dar University, Bahir Dar, Ethiopia, in 2007, and the M.Eng. degree (research) from the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea, in 2017. He is currently a Ph.D. research student with the Department of Computer Science and IT, La Trobe University, Melbourne, Australia. He was an R&D Radio Frequency Hardware Engineer at VOIXATCH, a smartwatch solutions company in South Korea. He was also with Ethio Telecom, Addis Ababa, Ethiopia, from 2007 to 2014, in different positions, including Radio Access Engineer and Radio Frequency Planner, mainly on network planning and optimization of multiple radio access technology, such as GSM, CDMA, and WCDMA. His research interests include the Internet of Things, wireless sensor networks, cognitive radio networks (spectrum sensing and spectrum management), future radio access technology, machine learning, optimization, and bioinspired computing.

NAVEEN CHILAMKURTI (Member, IEEE) received the Ph.D. degree from La Trobe University. He is currently the Cybersecurity Program Coordinator of computer science and information technology with La Trobe University, Melbourne, VIC, Australia. His current research areas include intelligent transport systems (ITS), smart grid computing, vehicular communications, vehicular cloud, cybersecurity, wireless multimedia, wireless sensor networks, and mobile security.

ABDUN MAHMOOD (Member, IEEE) received the B.Sc. degree in applied physics and electronics, and the M.Sc. (research) degree in computer science from the University of Dhaka, Bangladesh, in 1997 and 1999, respectively, and the Ph.D. degree from the University of Melbourne, Australia, in 2008. He is currently with the Department of Computer Science, School of Engineering and Mathematical Sciences, La Trobe University. Previously, he worked at UNSW, RMIT, Melbourne University, and the University of Dhaka. He has been working as an Academic in computer science, since 1999. In 2011, he joined UNSW as a Lecturer in computer science, until he joined La Trobe University, Melbourne, in 2017, where he is currently working as a Senior Lecturer in cybersecurity. His research interests include data mining techniques for scalable network traffic analysis, anomaly detection, and industrial SCADA security. He has published his work in various IEEE Transactions and A-tier international journals and conferences.

NOOR ZAMAN (JHANJHI) (Member, IEEE) received the Ph.D. degree in IT from Universiti Teknologi PETRONAS (UTP), Malaysia. He has great international exposure in academia, research, administration, and academic quality accreditation. He was with ILMA University, King Faisal University (KFU), for a decade, and currently with Taylor's University, Malaysia. He has 19 years of teaching and administrative experience. He has been awarded as the top reviewer 1% globally by WoS/ISI (Publons) recently, for the year 2019. He has edited/authored more than 11 research books with international reputed publishers, earned several research grants, and a great number of indexed research articles on his credit. He has supervised several postgraduate students, including master's and Ph.D. He is an Associate Editor of IEEE Access, a Keynote speaker for several IEEE international conferences globally, External Examiner/Evaluator for Ph.D. and master's for several universities, a Guest Editor of several reputed journals, a member of the editorial board of several research journals, and an active TPC member of reputed conferences around the globe.

YUNYOUNG NAM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, South Korea, in 2001, 2003, and 2007, respectively. From 2007 to 2010, he was a Senior Researcher with the Center of Excellence in Ubiquitous System. From 2010 to 2011, he was a Research Professor with Ajou University. He also spent time as a Post-Doctoral Researcher at the Center of Excellence for Wireless and Information Technology, Stony Brook University, NY, USA, from 2009 to 2013. From 2013 to 2014, he was a Post-Doctoral Fellow at the Worcester Polytechnic Institute, Worcester, MA, USA. In 2017, he was the Director of the ICT Convergence Rehabilitation Engineering Research Center, Soonchunhyang University, where he is currently an Assistant Professor with the Department of Computer Science and Engineering. His research interests include multimedia database, ubiquitous computing, image processing, pattern recognition, context-awareness, conflict resolution, wearable computing, intelligent video surveillance, cloud computing, biomedical signal processing, rehabilitation, and the healthcare systems.