IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 March 2023, accepted 6 April 2023, date of publication 17 April 2023, date of current version 5 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3267718

RESEARCH ARTICLE

A Robust Security Scheme Based on Enhanced Symmetric Algorithm for MQTT in the Internet of Things

AHMED J. HINTAW[1,2,3], SELVAKUMAR MANICKAM[1], SHANKAR KARUPPAYAH[1], (Member, IEEE), MOHAMMAD ADNAN ALADAILEH[4], MOHAMMED FAIZ ABOALMAALY[5], AND SHAMS UL ARFEEN LAGHARI[1]

[1]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang 11800, Malaysia
[2]Applied Medical Sciences College, University of Kerbala, Karbala 56001, Iraq
[3]Department of Computer Techniques Engineering, Alsafwa University College, Karbala 56001, Iraq
[4]Cybersecurity Department, School of Information Technology, American University of Madaba (AUM), Amman 11821, Jordan
[5]Department of Computer Center, Al-Zahraa University for Women, Karbala 56001, Iraq

Corresponding author: Selvakumar Manickam (selva@usm.my)

ABSTRACT Message Queuing Telemetry Transport (MQTT) is expected to be the de facto messaging IoT standard. Therefore, MQTT must achieve efficient security. Nevertheless, the most significant drawback of the MQTT is its lack of protection mechanisms. Meanwhile, the existing approaches have added processing overhead to the devices and are still vulnerable to various attacks. Therefore, this research work presented an integrated scheme known as the Robust Security Scheme (RSS) to protect the MQTT against any exploitations that might result in sophisticated cyberattacks. The proposed RSS employs two cryptosystems: 1) a dynamic variant of the Advanced Encryption Standard (D-AES); and 2) Key-Policy Attribute-Based Encryption (KP-ABE). RSS introduces a new design architecture of the symmetric AES algorithm to encrypt the MQTT payload called D-AES. Additionally, the second part of the proposed hybrid cryptosystem is KP-ABE, which is utilized to cipher the private key of the proposed D-AES to avoid the computation overhead of bilinear maps. The performance of the proposed RSS is measured in terms of processing time and traffic overhead. Additionally, the security aspects are evaluated in terms of balance, avalanche effect, and hamming distance and compared to the existing works in a testbed environment. Results revealed that the proposed D-AES is more promising with improvements than the standard AES algorithm. The proposed scheme achieves polymorphism while maintaining interoperability. RSS exhibited improvements over the standard AES algorithm by 8.75%, 10.45%, and 6.81% in terms of balance, avalanche effect, and hamming distance, respectively.

INDEX TERMS MQTT, dynamic encryption, cybersecurity, end-end security, Internet of Things, publish-subscribe systems.

I. INTRODUCTION

Various emerging innovations, such as environmental sensors [1], smart Homes, industrial devices [2], and vehicles [3], are anticipated to benefit from the communications possibilities made possible by the Internet of Things. Of importance is that IoT objects are expected to be on the rise, as stated by Ericsson, where 24 billion IoT devices will be interconnected and active in 2050 [4]. Besides, over 30.9 billion IoT objects are expected to be linked by 2025, thus according to Statista forecasts [5]. The majority of these linked objects utilize data communication protocols such as the MQTT protocol to exchange data. This emerging ecosystem introduces new concerns for communication security [6]. Such scenarios end-to-end protection is required, the design of which must take into account the limits of available bandwidth, computing

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

power, memory, and power. In light of this, the primary concern of IoT systems is their level of protection.

New architecture and robust security protocols are required to be adopted in order to reduce storage, communication, and computation overheads within IoT networks. The paradigm of publish-subscribe is a promising design in the IoT arena because it allows clients to interact with one another through a centralized message broker [7], [8]. MQTT is widely utilized in publish-subscribe systems that employ IoT devices [9]. MQTT is considered as the data communications protocol that is most suited to the IoT because it can be simply deployed on devices that are low-cost, low-power, and low-storage [10]. MQTT broker contains several topics that its clients can use to publish data or subscribe to a particular topic.

The present deployments of the MQTT standard still do not adequately account for security concerns. Consequently, IoT systems benefit greatly from the paradigm of publish-subscribe and protocols when security is considered. MQTT authentication relies on the user/passcode approach, and the transmitted passcode is not encrypted. It is possible to utilize external mechanisms for authentication within the connecting messages of MQTT such as Kerberos [11] as well as SCRAM [12]. Furthermore, it is strongly suggested by the MQTT standard specification that the server utilize Transport Layer Security (TLS) [13], [14] while implementing the protocol. However, these methods are not well suited for IoT that utilized limited resource nodes.

There are, in the current bibliography, many comprehensive studies about applying cryptographic schemes, [15], [16], [17]. Besides, Authentication and encryption of the payload are two aspects of MQTT interaction that are addressed in these studies using a variety of techniques and cryptographic primitives. Existing encryption algorithms for MQTT are either computationally heavy or prone to various cyber-attacks. One of the most cryptographic algorithms adopted in IoT is AES which is standardized by the National Institute of Standards and Technology (NIST). The three most crucial features of AES-128 were conceived during its development: extraordinary performance via script compactness and minimized design complexity, robustness against all known attacks. Many various types of attacks, such as linear cryptanalysis, differential cryptanalysis, square attacks, interpolation attacks, and related key attacks, were taken into account during the construction of AES. However, AES has been extensively used in various systems since it was standardized in NIST FIPS-197, although its vulnerabilities have been thoroughly investigated. Besides, it has spawned novel forms of assault, such as algebraic attacks, SAT solver attacks, and hybrid attacks [18]. Due to the number of AES attacks has been growing, it is important to strengthen the algorithm so that it can be utilized in the MQTT. In order to achieve that, the non-linearity of AES must be enhanced which will enhance the prevalence of confusion and, in turn, the security of the algorithm will be more effective.

In this work, the focus will narrow to MQTT security issues to enable security features for the protocol. Besides, this research work enables MQTT as an application layer protocol with enhanced security features for encrypting the protocol payload with the proposed RSS which has a polymorphic encryption feature for IoT devices considering the publish-subscribe communication paradigm. The most significant benefits of the proposed RSS are simplicity, performance efficiency, polymorphic feature, and robust security. The idea is relying on utilizing a less computational structure of the cryptosystem to be suitable for MQTT nodes. Our main contributions are summarized as follows:

- Strengthening the unit of key expansion to improve the strength of the symmetric algorithm at a very good scale through new advanced construction of the key expansion unit.
- The symmetric algorithm's SubByte transformation is improved to be round key dependent, resulting in a key change that is easily detected in the cipher text.
- Improvement of the symmetric algorithm's ShiftRow transformation in order to overcome the corresponding drawback and function dynamically and not rely on a static offset. As a result of the improved design, a better diffusion is achieved.
- A new security option called RSS is designed to provide confidentiality, access control, collusion resistance, and broadcast encryption to secure publisher-subscriber messages of MQTT-based IoT. In particular, the proposed RSS is able to provide at the same time confidentiality of payload for publisher and subscriber of MQTT-based IoT using our modified AES symmetric algorithm, fine-grained access control, collusion resistance, and broadcast encryption by using the ABE algorithm.

The remaining part of this article is organized as follows. In Section II, we discuss the IoT protocol stack. The existing literature related to security solutions for the MQTT is discussed in Section III. Section IV presents the AES Algorithm and its essential transformations. Then, the methodology of the proposed solutions is presented in Section V. The core component of the proposed Secure Publisher-Subscriber messages of the MQTT is covered in Section VI. Besides, RSS Design is detailed in Section VII. Implementation of the RSS is presented in Section VIII. Further, experimental results and discussion is covered in Section IX. Finally, the conclusion is given in Section X.

II. OVERVIEW OF MQTT

MQTT is a data communication protocol utilized in an IoT context that operates on top of TCP [3]. IBM invented the protocol as an inexpensive Machine-to-Machine (M2M) interaction technique which was subsequently recognized by OASIS.

The basis of MQTT is the publish-subscribe interaction architecture; under this approach, updates are conveyed to

subscribers rather than clients directly, which reduces the resources needed to process messages and makes the protocol well-suited for usage in constrained environments. Additionally, the protocol also operates on a server-client architecture wherein a broker broadcasts changes to MQTT nodes of any updates. Because of the reliance on the broker, nodes will not be able to interact with one another directly. MQTT messages have a tree-like form that consists of a topic upon which users may subscribe or publish. When a node publishes a message with an action or data, the broker will forward it to all other clients that have subscribed to that topic. MQTT was built for asynchronous transmission, in which publishing or subscriptions to or from separate entities occur in tandem. The protocol can also enable trustworthy transfers by selecting from among three types of reliability mechanisms, often known as Quality of Service (QoS) [9]. Compared to conventional protocols such as Hypertext Transfer Protocol (HTTP), MQTT has a significantly lower footprint, enabling it, as previously noted, far more appropriate for contexts with limited resources. Whereas MQTT offers numerous benefits, not all MQTT-based brokers have the same or equivalent entity authentication and encryption capabilities. Mosquitto, an open-source utility developed by Eclipse, supports the majority of the MQTT implementation capabilities, including client certificates and SSL/TLS. The default configuration of the MQTT broker has not provided protection for its messaging system, and authentication data is delivered in plaintext; consequently, it lacks security methods to secure the transferred data.

MQTT does indeed have a number of built-in security capabilities, such as encryption and authentication of both entities and messages, but these features are not enabled by default and require further configuration. Authentication techniques, such as utilizing the device's physical address (MAC), are available and thus are managed mostly by the MQTT broker via enrolling a node's data when it wants to join. A broker could use a tool called an Access Control List (ACL) to manage who has access to what resources [19]. An ACL stores information like client identification and credentials, as well as details about which entities a given client is authorized to acquire and what actions the client is permitted to do concerning those entities. As per the research work of [20], ciphering the data which has to be broadcasted at the application layer is a necessary step toward ensuring the confidentiality of sensitive information. Such an encryption mechanism may be used in a node-to-broker configuration or with an end-to-end strategy. By contrast, with node-to-broker ciphering, the broker deciphers the data before sending it to all the other nodes, who then cipher the data they get from the broker. Brokers in E2E scenarios can't read the encrypted data being sent out across topics, so they just transmit it along in a ciphertext version. In those, the broker merely serves as a mediator and doesn't even need extra functionalities that could cipher/decipher data, hence it consumes negligibly energy as well as negligibly processing resources. However, other security methods could be added at lower tiers if necessary. As pointed out within studies [21], [22], [23], security protocols such as TLS or maybe even SSL can be employed to reliably preserve the confidentiality of a communication channel at the transport layer. Further, AES in Counter Block Mode or AES in Counter with CBC-MAC mode, commonly known as CCM mode, could be utilized for ciphering the communication at the link layer, as stated in [20]. Such protection solution has several benefits over others, such as greater efficiency thanks to hardware acceleration features present within chips.

III. LITERATURE REVIEW

A. MQTT PROTOCOL

Considering the importance of securing MQTT protocol in IoT over the years, different schemes have been introduced by researchers for protecting data communication protocol in the IoT arena and the amongst utilized protocol is MQTT. The default state of the protocol is the cause of basically all security vulnerabilities.

Since the MQTT was developed primarily for usage in closed, air-gapped, and trustworthy IoT networks. Therefore, cybersecurity was not a significant concern during its development. Accordingly, it is imperative that the protection of the MQTT protocol from cyberattacks be ensured. Because this protocol was developed with a single goal in mind, performance before any things, the vast bulk of the protocol lacks security mechanisms and is susceptible to assaults. For the sake of secure communication within IoT, researchers are working on methods to protect the MQTT protocol.

Authors in [20] have highlighted the most applicable mechanisms in the area of the industrial IoT by presenting a comprehensive study of several MQTT security, the first evaluation option is to use AES-CCM to implement encryption of the Link layer to achieve hop-by-hop protection. In this scenario, the Link Layer Security (LLSEC) driver is the appropriate option to activate good security interactions between sensor and broker by ignoring the weakness of its "hope by hope" process to use instead of that single hope. The second evaluation indicates, offering an "end to end" option via using AES, AES-OCB, and AES-CBC to encrypt the payload. It is an attractive option to encrypt the payload with AES-OCB: it adds more security compared to AES-CCM and the AES-OCB option could not be handled when the payload amount is 64-byte, they have focused on the wind park scenario for their evaluation to impose it in the area of actual industrial use.

Bashir and Mir in [24], have utilized dynamic key-based XOR operation for securing publish-subscribe messages of MQTT. the intended mechanism introduced a lightweight cryptosystem solution for protecting the messages of the protocol malicious nodes within the IoT network.

The author in [25] proposed a new hybrid scheme, which allows using smaller size keys compared to existing cryptographic solutions by comparing different asymmetric

(RSA, ECC) and symmetric algorithms (AES 128, XTEA, HIGHT, RC5, and PRESENT) to investigate which lightweight algorithm is better to implement and modify this scheme due to data sending scenario. The hybrid scheme includes two types of cryptography at the same time: symmetric algorithms for encryption/decryption data; and asymmetric algorithms for key exchange. The optimal way to combine two different algorithms is to use ECC as the key manager and XTEA for encryption data. The proposed scheme can be applied in every IoT sector when data for sending has a small size.

The authors in [26], have presented a lightweight cryptosystem called Secure IoT (SIT). The proposed cryptosystem utilized two cryptography structures: substitution-permutation network and Feistel. Its 64-bit block size utilized 64-bits as an encryption key. but the 64-bit key is vulnerable to brute force attacks.

In [27], the authors have suggested a security solution for protecting MQTT in the IoT arena by utilizing ABE and Dynamic S-Box AES. They employ the KP-ABE scheme to make it lightweight [28] over lightweight ECC [29]. Due to these constraints possessed by existing security mechanisms. There has been the minimal implementation of security features for the MQTT protocol. As a result, the existing mechanisms that are being for securing MQTT Publish-Subscribe messages are still unprotected. They are typically computationally intensive mostly because of their complexity, need several rounds to encrypt, therefore squandering the limited energy of the devices, and are susceptible to cyber-attacks. Hence, a robust security option is required for securing MQTT Publish-Subscribe messages in IoT.

B. DYNAMIC ENCRYPTION

The existing approaches for protecting the MQTT network such as those proposed by [20], [27], [30], and [31] are still vulnerable to various attacks. All these approaches utilize the AES but cryptanalysis can be adapted and run successfully because the encryption algorithm in its design nature has less complexity in the key expansion. In addition, AES has a monomorphic design and every monomorphic cryptosystem performs the identical process with each input regardless of the key value. Thus, exposing the precise operations details of a cryptosystem in conjunction with weak implementation may lead to new ways to obtain significant details regarding the secret key by the attacker.

Many research works have been stated to enhance the AES algorithm to achieve a robust cipher. In addition to the obvious conflict with Kerckhoff's principle [32]. The authors [33] have modified the key of the AES to 320 bits in order to strengthen the key expansion unit by utilizing the Polybius square technique for deriving the secret keys from a password. This model has two significant issues. First, retrieving a password is easier than obtaining an encryption key. Because a user-selected password has low entropy, it may be obtained by employing simple approaches like social engineering [34]. Secondly, it is critical to consider the existing widespread implementations of the AES. Additionally, several authors proposed a variation of AES for protecting a certain data type. Especially, in [35], a cryptosystem was constructed primarily to improve the computational effort and security of the algorithm when used to encrypt data such as images. the study employed a chaotic mapping and Exclusive OR function in place of the MixColumns. This model is primarily intended to improve the performance of AES on images. Obviously, standard encryption must operate effectively with all possible types of data. Likewise, some various hybrid cryptosystems and protocols utilize the AES as stated in [36], [37], and [38]. Besides the complexity of designing hybrid cryptosystems, hybrids often require more processing, degrading cryptosystem performance.

Moreover, the authors, [39], [40], [41], [42] altered the transformation of the AES SubBytes. Instead of employing the conventional S-Box of the AES, they made it a dynamic object derived from the secret key. They have been stated that the constructed S-Box met the properties of the AES S-Box. As a result, each new key changes the structure of the cipher. This will definitely slow down attackers' ability to break the encryption. Manipulating the AES S-Box is deemed hazardous because of its careful selection to maintain maximum prop-ratio and input-output correlation values as minimal as possible. Thus, the propagation, as well as the correlation of input/output of the AES's S-Box are less than 2^6 and 2^3, respectively. These values determine the cipher's robustness to differential and linear assaults. Such 8-bit invertible S-Boxes generally have maximum prop-ratios of 2^{-5} to 2^{-4} and maximum input-output correlations of 2^{-2} [43]. As a result, the cipher's overall resistance to linear and differential assaults reduces.

In the work of [22], the dynamic S-box has been suggested in order to replace it with a fixed conventional AES S-box. They have utilized stream cipher Spritz as well as the hash function for generating the round key. However, the cryptosystem may be delayed due to the dynamic generation of the s-box, which requires more processing resources.

The study in [44], has detailed a Model for providing security solutions in the cloud that has been offered to be protected using a variation of the AES cryptosystem. They enhanced the AES in order to improve the encryption process speed such as 1000 blocks per second by utilizing the round key twice. In [45], tried to strengthen the AES cryptosystem for improving its security. The transformation of the ShifRow has been replaced with two new operations i.e., column permutation and row rotation, to have relied on the round key. But, the enhancement of the algorithm has added extra storage and substantial execution time.

Another dynamic cryptosystem scheme has been stated in [46]. The major goal of this scheme is to improve the cipher's performance by minimizing the number of cycles to one. The authors state that certain delay-sensitive systems cannot withstand the delay provided by common cryptosystems, such as AES. Moreover, the encryption must be secure enough

TABLE 1. The architecture and issues of the existing modification of the AES algorithm.

References	architecture	Issues
[33]	Increase the number of rounds to 16 and utilize the Polybius square technique in the key expansion	-The expense of implementation is significant -Retrieving a password is easier than obtaining an encryption key because the low entropy. -It is critical to consider the existing widespread implementations.
[35]	A chaotic mapping and Exclusive OR function in place of the MixColumns.	-The branch number of replaced Mixcolumn was not proved in order to ensure that any continuous four rounds of the AES have at least 25 active S-Boxes.
[39]–[42]	Altered the transformation of the AES SubBytes, made the S-Box a dynamic object derived from the secret key.	-Manipulating the AES S-Box is deemed hazardous because of its careful selection to maintain maximum prop-ratio and input-output correlation values as minimal as possible. -The cipher's overall resistance to linear and differential assaults reduces.
[22]	Dynamic S-box has been suggested in order to replace it with a fixed conventional AES S-box.	-The cryptosystem may be delayed due to the dynamic generation of the s-box.
[44]	In order to improve the encryption process speed, they utilized the round key twice.	-Disagreement with Kerckhoff's concept. -Issues of interoperability. -Uncertainty regarding security.
[45]	The ShifRow transformation has been replaced with two new operations i.e., column permutation and row rotation	-Added extra storage and substantial execution time. -Disagreement with Kerckhoff's concept. -Issues of interoperability.
[46]	Minimizing the number of cycles to one	-Rounds is the only way to reduce the propagation ratio and correlation for linear trails. As a result, there is no strong evidence in support of the assertion that linear or differential analysis is resistant in their approach

to withstand all known assaults. But it is well proven that repeating any cipher's transformations i.e., rounds is the only way to reduce the propagation ratio and correlation for linear trails. As a result, there is no strong evidence in support of the assertion that linear or differential analysis is resistant in their approach [47]. Table 1 depicted the architecture and issues of the existing modification of the AES algorithm.

Briefly stated, it could be observed that the majority of current research works seem to have at minimum one of these issues:

- The expense of implementation is significant.
- There is an obvious disagreement with Kerckhoff's concept, as well as issues of interoperability.
- Performance reduction is simply inappropriate.
- A monomorphic architecture in which the same actions are repeated for each new input.
- Uncertainty regarding security.

Therefore, the AES cryptosystem should be enhanced to make the algorithm function dynamically and work in different forms against these kinds of issues in order to design an RSS for protecting MQTT data.

IV. ADVANCED ENCRYPTION ALGORITHM

The symmetric algorithm AES operates on plaintext with a length of 128 bits and utilizes the same key for both encryption and decryption processes. The algorithm handles data in blocks of 128 bits and involves 10, 12, or 14 rounds employing a 128-bit, 192-bit, or 256-bit cipher secret, respectively. The block cipher of the AES algorithm forms as two diminution matrices 4 by 4 called state matrix. The state is employed to carry out the essential procedures of the algorithm.

The 128-bit block cipher of the AES may be broken down into three phases; the first phase is adding the initial round key, the second is rounds from one to nine, and the last phase is the last round. In the first phase of the algorithm, the 128-bit data is EXORed with the initial key of 16 bytes. The transformations such as SubBytes, ShiftRow, MixColumn, and AddRoundKey are executed nine times on a 4 by 4 twodimensional matrix known as a state matrix. In the final phase, only three transformations will be executed on the state matrix which are SubBytes, ShiftRows, and AddRoundKeys.

A. SUBBYTES TRANSFORMATION

The one and only non-linear and invertible bytes transformation is the SubBytes transformation. As a result,

$$s(y) = Affinetransformation(y^{(-1)}) \tag{1}$$

Affine Transformation

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} i_7 \\ i_6 \\ i_5 \\ i_4 \\ i_3 \\ i_2 \\ i_1 \\ i_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{2}$$

AES is sufficiently resistant to attacks. In this transformation of the algorithm, every byte in the state matrix is substituted with the stored value in the substitution box. There are 256 bytes in the lookup table tha1t makes up the S-box structure. The S-box values can be obtained by computing the multiplicative inverse there in the finite field GF (28), in which the input element has all bits set to zero, and then performing an affine translation over GF (2). Eq. (1) demonstrates the multiplicative inverse in the GF (28). Eq. (2) depicted the transformation over GF (2).

B. SHIFTROWS TRANSFORMATION

The transformation of the ShiftRows shifts most bytes in the row of the state matrix cyclically to the left, where rows 0,1,2, and 3 are shifted by 0,1,2, and 3 positions respectively. The value of the offset is row-dependent. Therefore, the first row does not alter at all. Moreover, within the AES, the diffusion feature is imparted through the cyclic rotation of rows.

C. MIXCOLUMNS TRANSFORMATION

The transformation of the MixColumns conducts processes on the state matrix for every column one at a time. This transformation is a diffusion process that involves a linear trajectory. Thus, every column of state data is treated as a four-term polynomial over GF (28). The modulo $(y4 + 1)$ with a fixed polynomial a (y) is multiplied by the column as presented by Eq. (3).

$$a\,(y) = \{03\}\,y^3 + \{01\}\,y^2 + \{01\}\,y^1 + \{02\} \tag{3}$$

Eq. (3) can alternatively be expressed as matrix multiplication, which is represented by Eq. (4):

$$p'\,(y) = a\,(y) \times p\,(y) \tag{4}$$

and in matrix form as Eq. (5):

$$\begin{bmatrix} P'_{0,c} \\ P'_{1,c} \\ P'_{2,c} \\ P'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} P_{0,c} \\ P_{1,c} \\ P_{2,c} \\ P_{3,c} \end{bmatrix} \tag{5}$$

D. ADDROUNDKEY TRANSFORMATION

The transformation of the AddRoundKey is performed in every round at the end of the round. this transformation employed the EXOR operation to EXORed the state data with the acquired round key. The EXOR operation is performed on the Nb words from the key schedule of each Round Key and the state matrix columns, as exhibited in Eq. (6).

$$\left[P'_{0,c}, P'_{1,c}, P'_{2,c}, P'_{3,c}\right] = \left[P_{0,c}, P_{1,c}, P_{2,c}, P_{3,c}\right] \oplus \left[P_{round*NP+c}\right] \tag{6}$$

where those words taken from the key scheduling denote as $[P_{round}]$, the round is a number that falls in $0 \leq$ round $\leq N_r$, and the round number is denoted as N_r.

E. KEY EXPANSION MODULE

The module of the key expansion utilizes the actual 128-bit key to produce the 128-bit keys needed for each cycle of the AES algorithm. Accordingly, this model has three transformations which are SubBytes, ShiftRows as well as RCON. Subsections A and B are clarified SubBytes and ShiftRows respectively. The round constant (RCON) employs the EXOR operation by utilizing the RCON array. The RCON array has data stated by $x^{(i-1)}$ with $[x^{(i-1)}\{00\}, \{00\}, \{00\}]$ powers of x where x represented as $\{02\}$ in the GF (2^8). Consequently, Eq. (7) is employed to construct each round key.

$$N\,(r, c) = \begin{cases} N(r-1, c) + sbox\,[Rword(N(r-1, c+3))] \\ \quad +Rcon(r-1)c = 1 \\ N(r, c-1) + N(r-1, c) 2 \leq c \leq 4 \end{cases} \tag{7}$$

As stated in Eq. (7), N(r, c) represents c^{th}32-bit column of r^{th} round key, where r > 1and1 ≤ c ≤ 4. Before the first round, the plaintext 128-bit is EXORed with the initial key r = 1.

V. METHODOLOGY

The Hybrid scheme for securing MQTT protocol in IoT has been proposed to provide confidentiality, access control, and collision resistance to the Publisher-Subscriber messages of MQTT protocol in IoT. Whilst designing the proposed technique, a number of assumptions have been made to ensure that it met the requirements.

A. ASSUMPTIONS

The proposed secure hybrid approach for securing MQTT data in the context of the Internet of Things is dependent on the following assumptions:

- Assume that MQTT publishers, subscribers as well as a broker have an honest-but-curious threat model which means they follow the protocol rightly, but as much as possible they are curious to learn about the transmitted messages.
- The key Authority, which issues keys for the encryption and decryption operation which are utilized to protect the data from unauthorized access, is assumed to be external to the system. The authority of the system does not misconduct and therefore is trusted by all entities.

B. SECURE HYBRID SCHEME

The RSS proposes to secure Publisher-Subscriber messages of the MQTT protocol in the IoT. The hybrid cryptosystem includes two cryptography algorithms: the proposed D-AES and KP-ABE algorithms. Therefore, taking advantage of the characteristics including both algorithms is to design a secure mechanism that offers a significant security improvement in the arena of the IoT while maintaining the complexity at a reasonable level. The next sections will discuss the symmetric algorithm, KP-ABE, and Broadcast encryption in detail. Fig. 1. depicts the block diagram of the proposed RSS.

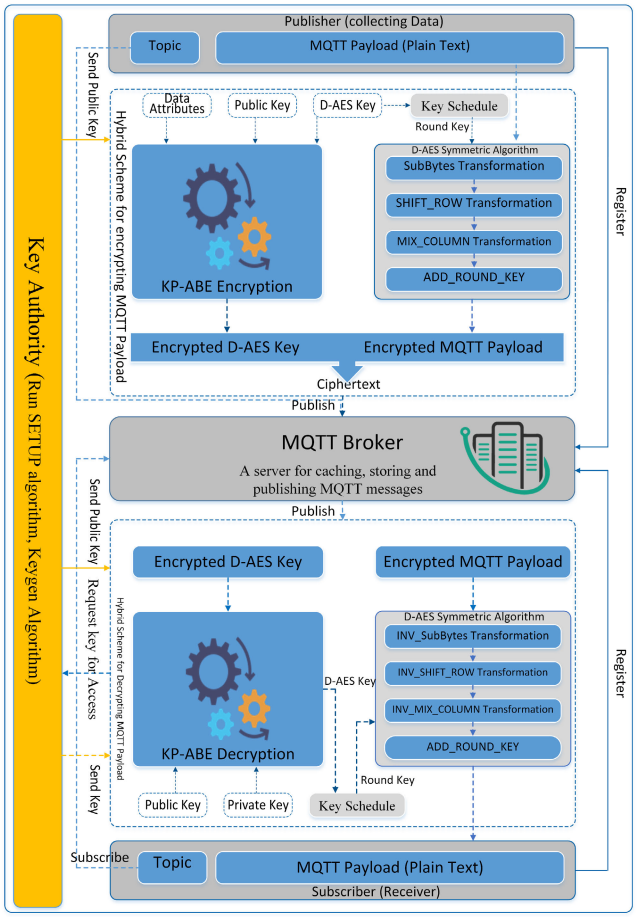

FIGURE 1. The block diagram of the proposed RSS.

VI. PROPOSED D-AES ALGORITHM WITH ENHANCED SECURITY

The core component of the proposed Secure Publisher-Subscriber messages of the MQTT protocol is enhancing the AES algorithm. The AES algorithm has been enhanced to be secure and employed to design a hybrid scheme to protect the MQTT message.

Without acquaintance of the key, a cryptosystem must be impenetrable. The intruder could employ the cryptoanalysis to acquire the plaintext or even the encryption key via utilizing linear as well as differential cryptoanalysis. Additionally, it will be more effective if the algorithm's complexity is weak. Cryptanalysis can be adapted and run successfully in case of the encryption algorithm in its design nature has less complexity for generating the ciphertext. Two strategies have been suggested by Claude Shannon, confusion as well as diffusion for preventing cryptanalysis by employing statistical approaches [48]. Diffusion hides the association between unencrypted data and encrypted data, and it represents the algorithm distribution nature. An encryption algorithm can have a high level of diffusion if a single bit is flipped in the original data, then most of the ciphertext data must be altered.

Consider a 128-bit block, if any bit flipped of the plaintext, then the generated encrypted text must have a sequence in which more than 64 bits had been changed. Whilst the confusion layer implies difficulty in order for identifying the correlation between the encrypted secret key as well as encrypted data. Substituting the encrypted data values with another value will attain confusion. Moreover, utilizing several replacements for the data will make the algorithm reach a heightened degree of confusion. The symmetric algorithm AES has a different stage to achieve confusion as well as diffusion.

In order to be effective, AES-128 has to possess three critical features. Resistance to all potential attacks, a high performance due to code compactness, and decreased design complexity. Moreover, attacks such as differential and linear cryptanalysis as well as interpolation, related key, and square attacks have not been taken into consideration throughout the algorithm's construction. Nevertheless, once NIST-FIPS-197 has announced that AES is becoming the standard cryptosystem and thus its popularization to be utilized in applications with high levels of security over a broad spectrum, there has been extensive investigation into the vulnerabilities in the algorithm. This has resulted in the development of extended types of threats and new kinds of an intruder, such as SAT solver and algebraic attacks, as well as hybrid and other attacks [18]. Therefore, it is important to deal with these attacks and must prevent them.

This research focuses on enhancing the standard algorithm to address its vulnerabilities and then employing the enhanced algorithm D-AES to design a robust scheme for securing MQTT data. The proposed enhancements will increase the confusion as well as the diffusion rate of the symmetric algorithm. To fulfill that, three different enhancements are made within the symmetric algorithm, as follows:

- A new design of the key expansion of the symmetric algorithm has been proposed for improving the algorithm security level in terms of diffusion, confusion, and complexity with enough performance level.
- To propose a new design of the SubBytes transformation of the AES algorithm to make it round key-dependent.
- Takes into account the downsides of the transformation of the shift row and proposes a new strategy for the ShiftRow transformation of the AES algorithm to make it round key-dependent and for improving the algorithm security level in terms of diffusion.

The key expansion unit's primary weakness is generating the subkey in a straightforward form. Operations such as bytes substitute, ShiftRow, and RCON XOR are involved within the first 32 bits of the key explanation unit. In the design, confusion could be achieved via diffuse the first word of a matrix within other words. The mapping of bytes to words in an expanded key is a very straightforward procedure. Because the EXOR action with zero yields the same key, it introduces a flaw within the architecture which enables the intruder for discovering the round keys, hence, the master secret key will be compromised. Therefore, the proposed enhancement of the algorithm key expansion employs round constants that are the exact length as the word, which means

that for all 10 rounds of the symmetric algorithm, a total of 40 round constants have been employed. Compared to the conventional AES, the proposed enhancements are slight changes that are introduced to transform components in the design in order to retain hardware simplicity while providing better security over the existing algorithm. The technique for the proposed key expansion improvement will be explained in more detail within the next section.

A. ENHANCED KEY EXPANSION

When it comes to the operations of encryption as well as decryption, thus the key is still the most critical element. Thus, is the key upon which the whole security of the system has relied; if an adversary obtains knowledge of this key, the secrecy of the system is compromised. Moreover, it is vital to take the required precautions in order to make the disclosure of the key as complex as possible. The process of generating keys can be run entirely on a decoder as stated by ([49], [50], [51], but in the IoT systems, the node itself serves as the network node, the functions implicated in the process of generating the key should correspondingly be taken into consideration to the significant extent that they guarantee the essential security in the IoT environment. As presented by the studies in [52] and [53], the key schedule process of the AES algorithm suffers vulnerabilities due to the forthright associations between sub-key bytes that are created as a consequence of the key schedule operation, an adversary can exploit this vulnerability in the rounds of the algorithm in order to crack the key.

The key expansion of the AES algorithm has a critical drawback in that the words that are produced from the key schedule are associated with one another. Once the overall key has been obtained via the differential technique or liner techniques of cryptanalysis, adversaries can be employed to crack the key in the AES round [52], [53].

Thus, the intruder could be capable to obtain SK_{13}^{0} which is represented by the byte 13th of the private key in the algorithm by employing Eq. (8) in case of the intruder has knowledge about the ciphertext as well as the original plaintext [53], as follows:

$$SK_{13}^{0} = SRK(P_{t13}; (SBI(ARK(C_{t13}; SK_{13}^{1})))) \quad (8)$$

It will then be possible to obtain SK_{9}^{1} by employing (9) in the form described below:

$$SK_{13}^{1} = SK_{13}^{0} \oplus SK_{9}^{1} \quad (9)$$

The true key length $SK_t \mathrm{Sk_t}$ must therefore be sufficiently long to prevent the adversary from performing 2^{SK_t-1} encryption methods in order to conduct key searching attacks. This is crucial to preserve protection against thorough search attacks. So, this portion of the algorithm needs to be addressed to make it strong adequately to remain inconspicuous during an attack while also using less processing during the key generation process.

Additionally, despite the fact that the S-boxes, XOR operation, as well as the shifting in the g function, are all contributed to the algorithm's confusing features, the reverse engineering technique may readily return to the original key space. More, the biased inputs in the key space disclose the distinctions between the words, allowing for the partial acquisition of the key space.

Therefore, in order to address this limitation, the unit of the key expansion is structured to utilize four-round constant values for every round in order to compensate for this vulnerability. The values of the RCON are used sequentially according to the RCON obtained by the proposal of Rijndael in GF (28). Fig. 2 depicts the structural layout of the improved key expansion unit via employing the sequential RCON. Utilizing these round constants within the key expansion component effectively improves AES' confusion feature.

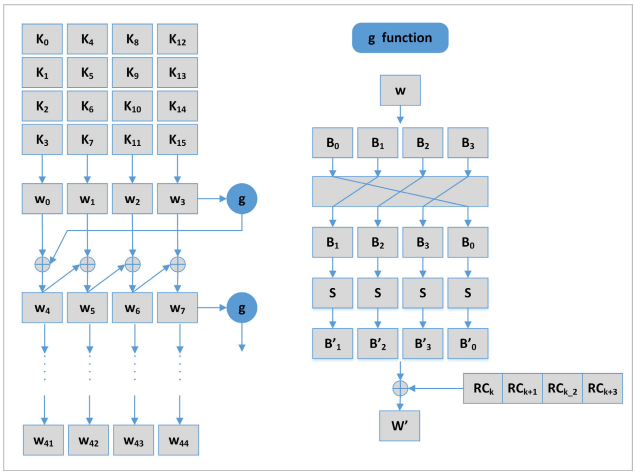

FIGURE 2. The architecture of the proposed key expansion block.

While taking into consideration the trade-off caused in the enhanced structure of the key schedule in terms of storage, due to utilizing the RCON values with each expanded key in the g function will resultant a slight increase in the terms of area. therefore, 24 registers are required for the purpose of holding the RCON value. Accordingly, the proposed technique for improving the key expansion security level has achieved a better level of security at the tradeoff of an insignificant increase in area. Consequently, the enhanced key expansion strategy yields a subkey that is not really the identical subkey that is utilized in Add Round Key (ARK). Thus, SK_{13}^{0} in (6) cannot be the same with SK_{13}^{0} in (7). As a result, the attacker is unable to compute the key by employing these formulas.

AES encryption techniques consist of numerous rounds, individually of which requires a unique key. The enhanced algorithm would be a 128-bit as block cipher size, which probably takes a 128-bit key employed to encrypt 128-bits of information and is limited to ten rounds; thus, ten individual keys are required for the stated purpose. The user is required to provide a cipher key SK_c with a length of 128 bits. Thus, this key will be utilized as the entry point for the key

expansion block. The next enhancement will be elaborated on in detail in the next section.

B. ENHANCED SUBBYTES TRANSFORMATION

This portion of the algorithm transformation of SubBytes has been enhanced to be key dependent on a round key which works dynamically with the round key of the algorithm in order to guarantee that any change in the key is readily detected within the encrypted message.

The enhanced transformation utilizes the 128-bits round key, where, all the bytes in the matrix of the 128-bits round key have been EXORed to get four keys EXORkey0, EXORkey1, EXORkey2, EXORkey3 each one has a size of eight-bit, as demonstrated by Eq. (10). Behind the getting of the EXORkeys, each EXORkeyi as stated in Eq. (11)–(14) was appended to all the other bytes within Rowi of the state-matrix prior to replacing the values within the substitution box. Theoretically, the state-matrix $S_{m(i,j)}$ is represented as well as a round key $K_{r(i,j)}$, depicted as a 4 by 4 matrix:

$$S_{m(i,j)} = \begin{bmatrix} S_{m(0,0)} & S_{m(0,1)} & S_{m(0,2)} & S_{m(0,3)} \\ S_{m(1,0)} & S_{m(1,1)} & S_{m(1,2)} & S_{m(1,3)} \\ S_{m(2,0)} & S_{m(2,1)} & S_{m(2,2)} & S_{m(2,3)} \\ S_{m(3,0)} & S_{m(3,1)} & S_{m(3,2)} & S_{m(3,3)} \end{bmatrix}$$

$$K_{r(i,j)} = \begin{bmatrix} K_{r(0,0)} & K_{r(0,1)} & K_{r(0,2)} & K_{r(0,3)} \\ K_{r(1,0)} & K_{r(1,1)} & K_{r(1,2)} & K_{r(1,3)} \\ K_{r(2,0)} & K_{r(2,1)} & K_{r(2,2)} & K_{r(2,3)} \\ K_{r(3,0)} & K_{r(3,1)} & K_{r(3,2)} & K_{r(3,3)} \end{bmatrix}$$

$$EXORK_{r(i)} = k_{r(i,0)} \oplus k_{r(i,1)} \oplus k_{r(i,2)} \oplus k_{r(i,3)} \quad \text{where } i = 0 \text{ to } 3 \tag{10}$$

Alternatively,

$$EXORK_{r(0)} = k_{r(0,0)} \oplus k_{r(0,1)} \oplus k_{r(0,2)} \oplus k_{r(0,3)} \tag{11}$$

$$EXORK_{r(1)} = k_{r(1,0)} \oplus k_{r(1,1)} \oplus k_{r(1,2)} \oplus k_{r(1,3)} \tag{12}$$

$$EXORK_{r(2)} = k_{r(2,0)} \oplus k_{r(2,1)} \oplus k_{r(2,2)} \oplus k_{r(2,3)} \tag{13}$$

$$EXORK_{r(3)} = k_{r(3,0)} \oplus k_{r(3,1)} \oplus k_{r(3,2)} \oplus k_{r(3,3)} \tag{14}$$

In order to obtain the new state matrix $S'_{m(i,j)}$, it was required to employ Eq. (15).

$$S'_{m(i,j)} = S'_{m(i,j)} \oplus EXORK_{r(i)} \text{where } i, j = 0 \text{ to } 3 \tag{15}$$

below the following matrix exhibits the operations evidently

$$S'_{m(i,j)} = \begin{matrix} S_{m(0,0)} \oplus K_{r(0)} S_{m(0,1)} \oplus K_{r(0)} S_{m(0,2)} \oplus K_{r(0)} S_{m(0,3)} \oplus K_{r(0)} \\ S_{m(1,0)} \oplus K_{r(1)} S_{m(1,1)} \oplus K_{r(1)} S_{m(1,2)} \oplus K_{r(1)} S_{m(1,3)} \oplus K_{r(1)} \\ S_{m(2,0)} \oplus K_{r(2)} S_{m(2,1)} \oplus K_{r(2)} S_{m(2,2)} \oplus K_{r(2)} S_{m(2,3)} \oplus K_{r(2)} \\ S_{m(3,0)} \oplus K_{r(3)} S_{m(3,1)} \oplus K_{r(3)} S_{m(3,2)} \oplus K_{r(3)} S_{m(3,3)} \oplus K_{r(3)} \end{matrix}$$

The following is the state matrix $S'_{m(i,j)}$ that was produced:

$$S'_{m(i,j)} = \begin{bmatrix} S'_{m(0,0)} & S'_{m(0,1)} & S'_{m(0,2)} & S'_{m(0,3)} \\ S'_{m(1,0)} & S'_{m(1,1)} & S'_{m(1,2)} & S'_{m(1,3)} \\ S'_{m(2,0)} & S'_{m(2,1)} & S'_{m(2,2)} & S'_{m(2,3)} \\ S'_{m(3,0)} & S'_{m(3,1)} & S'_{m(3,2)} & S'_{m(3,3)} \end{bmatrix}$$

upon acquiring the matrix S'_m, then the S-Box will be utilized to perform the byte substitution through doing conventional SubBytes transformation as illustrated in Eq. (16)

$$S'_{m(i,j)} = substititutionBox\left[S'_{m(i,j)}\right] \quad \text{where j=0 to 3 for each i =0 to 3} \tag{16}$$

C. ENHANCED INVSUBBYTES TRANSFORMATION

The invertibility of the SubBytes transformation has been verified in order to achieve the enhanced inverse SubBytes process, as indicated in Eq. (17).

Assumed $f(k)$ and $f(x)$ within $GF\left(2^8\right)$ is operated, in such a way:

$$\begin{aligned} f\ (h) &= (f\ (k) \oplus f\ (x)) \\ f(x) &= ((f(k) \oplus f(x)) \oplus f(k)) = f(k) \oplus f(h) \end{aligned} \tag{17}$$

as the following is the evidence to proof that:

Two numbers of hexadecimal are assumed CE and EF represented as 11001110 and 11101111 in binary respectively, the value x, that is the EXOR of CE and EF, could well be generated by adding CE to EF while performing an exclusive OR function on the two numbers. The process is illustrated, in the manner described below:

It is possible to describe EF and CE as polynomial functions $f(k)$ and $f(x)$, at Galois Field, to be exact $2^8(GF(2^8))$, in such a way that:

$$f\ (x) = x^7 + x^6 + x^5 + x^3 + x^2 + 1 = EF \tag{18}$$

$$f\ (k) = x^7 + x^5 + x^4 + x^3 + x^2 + x^1 + 1 = CF \tag{19}$$

Consequently,

$$\begin{aligned} f(x) &= f(k) \oplus f(x) \\ &= x^7 + x^5 + x^4 + x^3 + x^2 + x^1 + 1 \oplus x^7 \\ &\quad + x^5 + x^3 + x^2 + x^1 + 1 \\ &= x^6 + x^4 + x^1 \end{aligned} \tag{20}$$

$f(h) = x^6 + x^4 + x^1 = 21$ in hexadecimal represented as 00100001 in binary. in order to prove this $f(x) = (f(k) \oplus f(x) \oplus f(k)) \oplus f(k) = f(k) \oplus f(h)$ via utilizing the polynomials $f(k) \oplus f(h)$, this will resultant:

$$\begin{aligned} f(x) &= f(k) \oplus f(h) \\ &= x^7 + x^5 + x^4 + x^3 + x^2 + x^1 + 1 \oplus x^6 + x^4 + x^1 \\ &= x^7 + x^6 + x^5 + x^3 + x^2 + 1 \end{aligned} \tag{21}$$

Since $f(x) = (f(k) \oplus f(x)) \oplus f(k = f(k) \oplus f(h)$

It has been proved that the enhanced transformation of the SubBytes is invertible. While performing the inverse transformation of the SubBytes, the substitution is performed prior to EXORing the state matrix with the EXORkeys via:

$$S'_{m(i,j)} = InverseSubstititutionBox\left[S'_{m(i,j)}\right] \quad \text{where } j = 0 \textit{ to } 3 \text{ for each } i = 0 \text{ to } 3. \tag{22}$$

After the substitutions, the state S'_m could be derived in the following manner:

$$S'_{m(i,j)} = \begin{bmatrix} S'_{m(0,0)} & S'_{m(0,1)} & S'_{m(0,2)} & S'_{m(0,3)} \\ S'_{m(1,0)} & S'_{m(1,1)} & S'_{m(1,2)} & S'_{m(1,3)} \\ S'_{m(2,0)} & S'_{m(2,1)} & S'_{m(2,2)} & S'_{m(2,3)} \\ S'_{m(3,0)} & S'_{m(3,1)} & S'_{m(3,2)} & S'_{m(3,3)} \end{bmatrix}$$

Hence, the initial state of the matrix S_m could be recovered with EXORing S'_m together within EXORkeys, as demonstrated within Eq. (23):

$$S'_{m(i,j)} = S'_{m(i,j)} \oplus EXORK_{r(i)} \quad \text{where } j = 0 \text{ to } 3 \text{ for each } i \text{ from 0 to 3} \tag{23}$$

Below is a matrix demonstrating the transformation of the inverse SubBytes:

$$S'_{m(i,j)} = \begin{matrix} S'_{m(0,0)} \oplus K_{r(0)} S'_{m(0,1)} \oplus K_{r(0)} S'_{m(0,2)} \oplus K_{r(0)} S'_{m(0,3)} \oplus K_{r(0)} \\ S'_{m(1,0)} \oplus K_{r(1)} S'_{m(1,1)} \oplus K_{r(1)} S'_{m(1,2)} \oplus K_{r(1)} S'_{m(1,3)} \oplus K_{r(1)} \\ S'_{m(2,0)} \oplus K_{r(2)} S'_{m(2,1)} \oplus K_{r(2)} S'_{m(2,2)} \oplus K_{r(2)} S'_{m(2,3)} \oplus K_{r(2)} \\ S'_{m(3,0)} \oplus K_{r(3)} S'_{m(3,1)} \oplus K_{r(3)} S'_{m(3,2)} \oplus K_{r(3)} S'_{m(3,3)} \oplus K_{r(3)} \end{matrix}$$

The equation is as follows for obtaining the resulting matrix S, which represents the initial state:

$$S_{m(i,j)} = \begin{bmatrix} S_{m(0,0)} & S_{m(0,1)} & S_{m(0,2)} & S_{m(0,3)} \\ S_{m(1,0)} & S_{m(1,1)} & S_{m(1,2)} & S_{m(1,3)} \\ S_{m(2,0)} & S_{m(2,1)} & S_{m(2,2)} & S_{m(2,3)} \\ S_{m(3,0)} & S_{m(3,1)} & S_{m(3,2)} & S_{m(3,3)} \end{bmatrix}$$

It is exemplified in the above demonstration that the transformation of the SubBytes is invertible, since

$$SubstititutionBox\left[S_{m(i,j)} \oplus XORK_{r(i)}\right] = InvSubstititutionBox[S'_{m(i,j)} \oplus XORK_{r(i)}] \quad \text{where j} = 0 \text{ to } 3, \text{ for each i from 0 to 3.}$$

D. ENHANCED SHIFTROWS TRANSFORMATION

Another improvement has been achieved throughout the ciphering technique. The transformation of the ShiftRows has been enhanced in terms of the security level by making the entire transformation randomized. The process of ShiftRows in the standard algorithm is dependent on a static number, known as offset, that specifies which position of the elements in the state matrix should be shifted. Therefore, the transformation process in this enhancement does not rely on the static offset but became a dynamic transformation process that relied on a dynamic value called Position Number P_n. This P_n could be determined by manipulating every row of the derivative round key and the corresponding row of the matrix. Following that, the rows will be shifted in the state matrix based on the obtained position value. The following below describes how the P_n will be obtained by utilizing the round key matrix $K_{r(i,j)}$ and state matrix $S_{m(i,j)}$.

$$S_{m(i,j)} = \begin{bmatrix} S_{m(0,0)} & S_{m(0,1)} & S_{m(0,2)} & S_{m(0,3)} \\ S_{m(1,0)} & S_{m(1,1)} & S_{m(1,2)} & S_{m(1,3)} \\ S_{m(2,0)} & S_{m(2,1)} & S_{m(2,2)} & S_{m(2,3)} \\ S_{m(3,0)} & S_{m(3,1)} & S_{m(3,2)} & S_{m(3,3)} \end{bmatrix},$$

$$K_{r(i,j)} = \begin{bmatrix} K_{r(0,0)} & K_{r(0,1)} & K_{r(0,2)} & K_{r(0,3)} \\ K_{r(1,0)} & K_{r(1,1)} & K_{r(1,2)} & K_{r(1,3)} \\ K_{r(2,0)} & K_{r(2,1)} & K_{r(2,2)} & K_{r(2,3)} \\ K_{r(3,0)} & K_{r(3,1)} & K_{r(3,2)} & K_{r(3,3)} \end{bmatrix}$$

The first step is adding every row in the matrix of the round key with the state matrix corresponding row utilizing the EXOR to calculate a new vector called a key state K_{sv}. Then, EXORing the obtained vector that has a size K_{sv} of 4-bytes with the calculated Position Value P_v. The next step is storing the 8-bits position value $P_{v(i)}$ in the state matrix, particularly in Row_i. Repeat the above process again for other rows Row1–Row3. Now, the position number P_n is attached to the position value $P_{v(i)}$ that has been calculated previously for every state row and set with ascending ordering only with the lowest position value holding one as the position number and the highest position value holding four as a position number. This procedure could be represented mathematically as described in the following:

Eq. (24) is utilized to calculate a vector of the key state K_{sv}, such that:

$$K_{sv(i)} = ((S_{m(i,0)} \oplus k_{r(i,0)}), (S_{m(i,1)} \oplus k_{r(i,1)}), (S_{m(i,2)} \oplus k_{r(i,2)}), (S_{m(i,3)} \oplus k_{r(i,3)})) \tag{24}$$

Besides that, Eq. (24) will be further simplified even more into Eq. (25), which yields the following:

$$K_{sv(i)} = (SK_{i,0}, SK_{i,1}, SK_{i,2}, Sk_{i,3}), \quad where\ Sk_{i,j} = (S_{m(i,j)} \oplus k_{r(i,j)}) \tag{25}$$

As an alternative, Eq. (25) could well be broken down into the following equations: (26)–(29):

$$K_{sv(0)} = (SK_{0,0}, SK_{0,1}SK_{0,2}, SK_{0,3}) \tag{26}$$

$$K_{sv(1)} = (SK_{1,0}, SK_{1,1}SK_{1,2}, SK_{1,3}) \tag{27}$$

$$K_{sv(2)} = (SK_{2,0}, SK_{2,1}SK_{2,2}, SK_{2,3}) \tag{28}$$

$$K_{sv(3)} = (SK_{3,0}, SK_{3,1}SK_{3,2}, SK_{3,3}) \tag{29}$$

The position value $P_{v(i)}$ is then calculated employing Eq. (30), in such a way that:

$$P_{v(i)} = \left(SK_{i,0} \oplus SK_{i,1} \oplus SK_{i,2} \oplus SK_{i,3}\right), \text{ Where } i = 0 \text{ to } 3 \tag{30}$$

Instead, Eq. (30) may be split down into the following separate equations describing every row in the state exhibited in Eq. (31)– (34):

$$P_{v(0)} = (SK_{0,0} \oplus SK_{0,1} \oplus SK_{0,2} \oplus SK_{0,3}) \tag{31}$$

$$P_{v(1)} = (SK_{1,0} \oplus SK_{1,1} \oplus SK_{1,2} \oplus SK_{1,3}) \tag{32}$$

$$P_{v(2)} = (SK_{2,0} \oplus SK_{2,1} \oplus SK_{2,2} \oplus SK_{2,3}) \tag{33}$$

$$P_{v(3)} = (SK_{3,0} \oplus SK_{3,1} \oplus SK_{3,2} \oplus SK_{3,3}) \tag{34}$$

Then the position number P_n will be attached to the values of the position value $P_{v(i)}$ in the ascending order, where P_n of 1 will be attached to the smallest value in P_v and the P_n of 4 will be attached to the biggest value of P_v. Table 3 depicted the association between both the P_n and P_v to execute how many elements will be shifted for every row.

Now, every byte in the state matrix will be split up into tetrad, where every state element will be represented as two states.

$$\begin{bmatrix} S_{m(0,0)} & S_{m(0,1)} & \cdots & S_{m(0,3)} \\ \vdots & \vdots & \vdots & \vdots \\ S_{m(3,0)} & S_{m(3,1)} & \cdots & S_{m(3,3)} \end{bmatrix} = \begin{bmatrix} S_{m(0,0(4))} & S_{m(0,0(4))} & \cdots & S_{m(0,3(4))} & S_{m(0,3(4))} \\ \vdots & \vdots & \vdots & \vdots \\ S_{m(3,0(4))} & S_{m(3,0(4))} & \cdots & S_{m(3,3(4))} & S_{m(3,3(4))} \end{bmatrix}$$

The i^{th} row, as well as a j^{th} column of each state matrix, are denoted as $S_{(i,j)k}$, with K equal to 4 to represent a tetrad-based operation on the state element rather than performing a byte operation within the state matrix, similar to the usual form. The state rows utilize as a tetrad as well as the data is EXORed according within P_v with the constant binary number corresponding to each tetrad. Table 2 presented the bite-sized splitting of the byte, including its operations and binary values. The same procedure is followed for each and every column. Fig.3 depicts the decision flowchart for the enhanced shiftrows transformation. On average, the confusion rate will be increased when the enhanced method is executed ten times for the 128-bits block to generate the encrypted data utilizing the algorithm as well as increasing its resistance to cryptanalysis. The association between P_n and P_{n-1} of the enhanced transformation is depicted in Table 3.

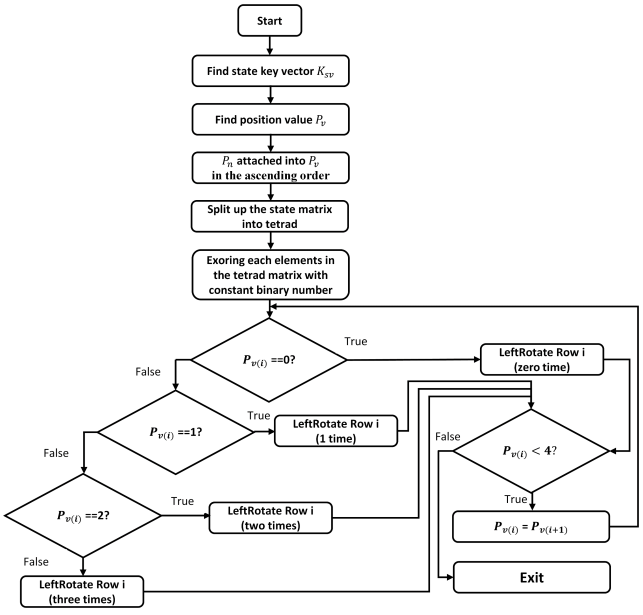

FIGURE 3. The decision flowchart for the enhanced shiftrows.

TABLE 2. Bite-sized splitting of the byte including its operations and binary value.

Tetrad-based	Operation	Binary value
$S_{m(0,4)}$	$\oplus$	0000
$S_{m(1,4)}$	$\oplus$	0001
$S_{m(2,4)}$	$\oplus$	0010
$S_{m(3,4)}$	$\oplus$	0011

TABLE 3. Association between P_n and P_{n-1} of the enhanced D-AES.

S/No	P_n	P_{n-1}
1	1	0
2	2	1
3	3	2
4	4	3

E. ENHANCED INVSHIFTROWS TRANSFORMATION

The enhanced InvShiftRows transformation differs slightly from the enhanced ShiftRows transformation; where, every byte within the state matrix is splatted up to tetrads such that in the enhanced ShiftRow transformation, where every state element is represented as two states and the data is EXORed according to the P_v with the constant binary number corresponding to each tetrad as demonstrated in Table 4. The same procedure was done for each and every column. where every state element will be represented as two states.

$$\begin{bmatrix} S'_{m(0,0)} & S'_{m(0,1)} & \cdots & S'_{m(0,3)} \\ \vdots & \vdots & \vdots & \vdots \\ S'_{m(3,0)} & S'_{m(3,1)} & \cdots & S'_{m(3,3)} \end{bmatrix} = \begin{bmatrix} S'_{m(0,0(4))} & S'_{m(0,0(4))} & \cdots & S'_{m(0,3(4))} & S'_{m(0,3(4))} \\ \vdots & \vdots & \vdots & \vdots \\ S'_{m(3,0(4))} & S'_{m(3,0(4))} & \cdots & S'_{m(3,3(4))} & S'_{m(3,3(4))} \end{bmatrix}$$

The 16-bytes round key will be taken in reverse order through the encryption manner and the P_n the calculation for the InvShiftRows transformation will remain the same such that in the enhanced ShiftRows transformation. The only thing that is different is the direction in that the state matrix rows are shifted. Where relying on the P_n, the rows within the state matrix will be shifted into the right for the inverse process. The block diagram of the enhanced symmetric D-AES algorithm for the 128-bis block is illustrated in Fig. 4.

TABLE 4. Bite-sized splitting of the byte, including its operations and binary values for the Invshift transformation.

Tetrad-based	Operation	Binary value
$S'_{m(0,4)}$	$\oplus$	0000
$S'_{m(1,4)}$	$\oplus$	0001
$S'_{m(2,4)}$	$\oplus$	0010
$S'_{m(3,4)}$	$\oplus$	0011

VII. DESIGNED RSS

MQTT's implementation includes no protections against cyberattacks on its interaction. In addition, the current implementations of standard MQTT support only simple authorization policies and basic authentication. The most important drawback of the MQTT protocol is it has only a minimal set of security protections. Consequently, when communication is exploited threat actors potentially interrupt it and conduct cyberattacks as a result. this work introduces the design of the proposed security system, namely, an RSS to secure publish-subscribe messaging of the MQTT protocol in the IoT.

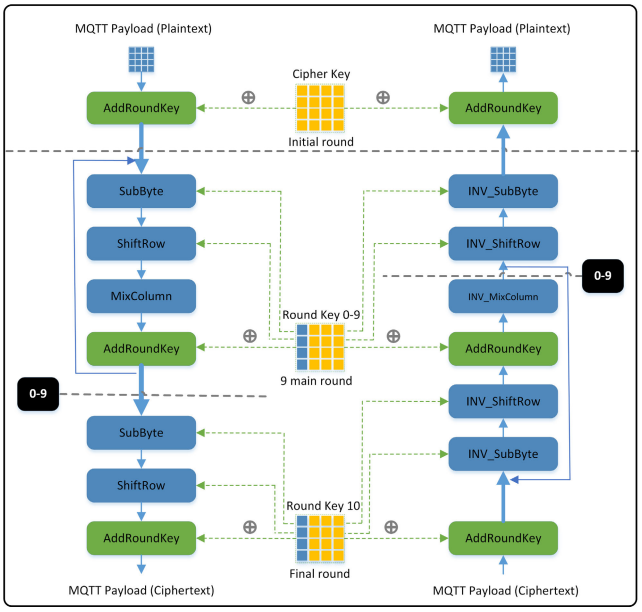

FIGURE 4. The block diagram of the enhanced D-AES for 128-bits.

The proposed hybrid cryptosystem employs two different cryptosystems that include: Symmetric Key Cryptosystem (SKC) and Asymmetric Key Cryptosystem (AKC). The SKC cryptosystem is utilized for encryption as well as decryption of the MQTT payload that used a symmetric secret key. The AKC cryptosystem is utilized in order to distribute the symmetric key that is utilized by the MQTT publisher for encrypting the payload as well as to provide confidentiality of the MQTT payload, broadcast encryption, fine-grained access control, and collusion resistance. AKC algorithm uses a public parameter as well as a list of data attributes.

KP-ABE was selected to be used for AKC due to its promising approach that enables expressive and offers features such as fine-grained data access control that is governed by rules or implies conditions constructed from attributes. Additionally, AES 128-bit has been chosen to be used with SKC based to its performance on a variety of systems. Due to the obvious anticipated significant cost of KP-ABE as compared to symmetric key encryption, the actual MQTT payload was not explicitly encrypted utilizing KP-ABE in this implementation. Rather than that, encrypting the MQTT payload with the proposed D-AES algorithm which uses128-bit secret key; thus D-AES key is encrypted by employing KP-ABE and dispatched along with the cipher text to the MQTT broker under the specified topic. As with many other public key encryption techniques, this is a typical practice to adopt.

Moreover, KP-ABE mentioned in [28] based on lightweight ECC [29], [54] was selected to be used for AKC. Note, CP-ABE has been skipped in the proposed flow as in [55] and [56], authors have demonstrated that KP-ABE exceeds CP-ABE and that it is important to limit the length of time and emphasis mostly on high-level encrypted data for IoT. Normally, an access policy is stated as an expression that has a collection of attributes and boolean constructions (OR, AND). The secure hybrid cryptosystem involves six algorithms: Setup, SKC_Encryption, AKC_Encryption, Key generation, AKC_Decryption, and SKC_Decryption.

Setup: This is the initiation phase of the proposed scheme, which entails the setup algorithm to be run by authority, which is a randomization process for generating the public key parameters (PK) as well as a master key (MK). Public key parameters should be available to be dispatched but the authority retains the master key secretive.

•**SKC_Encryption**(P, S_k): The D-AES algorithm is run by the MQTT publisher to encrypt the MQTT payload and outputs the encrypted payload EP by taking the payload P to be encrypted and the D-AES private key S_k as input. Therefore, the outputs of this process will be the encrypted MQTT payload.

•**AKC_Encryption**(S_k, γ, Params): In this stage, the MQTT publisher performs KP-ABE encryption which takes the D-AES key ES_k, the attributes set γ, and the public key parameters PK as input in order to generate the encrypted D-AES key S_k, then the generated ciphertext will be dispatched to the MQTT broker under a particular topic, along with encrypted MQTT payload and the attributes set.

•**Key-Generation** (MK, Params, Γ): The algorithm of the key generation performs by the authority, and it is also a randomized process. In this stage, when an MQTT subscriber sends a key request to the authority along with its access policy, the authority would then run the keygen algorithm, which will take a master secret key, public key, and the output of the policy decision Γ from the authority as input in order for generating a decryption key access rights to the MQTT subscriber. After that, the key authority dispatches the private key access rights into the subscriber. During this stage, the authority does partial decryption of the data. Therefore, the output of this process will be the decryption key of the KP-ABE that will be used for the decryption process by the MQTT subscriber.

•**AKC_Decryption** (ES_k, d, Params): In this stage, the MQTT subscriber first sends a key request to the authority along with its access policy in order to get the decryption key access rights. Secondly, the subscriber would then perform the decryption process by running KP-ABE in order for decrypting the encrypted symmetric key of the D-AES ES_k that was encrypted by the publisher node under a set of

attributes γ, which will take the encrypted D-AES key, public parameter, and the received KP-ABE decryption key from the authority as input to decrypt the encrypted private key of the D-AES. After that, in order for a subscriber to decipher the encrypted private key of the D-AES, the access policy that is included within the decryption key of KP-ABE must meet that of the encrypted data, in another word, If $\Gamma\ (\gamma) = 1$. Therefore, the outputs of this process will be the symmetric D-AES key S_k that will be used for further decryption by the next process.

•**SKC_ Decryption**(EP, S_k): In this stage, the MQTT subscriber performs the D-AES decryption algorithm in order to decrypt the actual payload data by taking the decrypted D-AES private S_k from the previous process of KP-ABE and the encrypted MQTT payload EP as input to decrypt the payload data of the MQTT. Therefore, the output of this process will be the MQTT payload.

VIII. IMPLEMENTATION

In order to implement the proposed RSS, the Eclipse Java platform was used in this experiment. Besides, the elements of the experimental setup are MQTT Publisher, Subscriber, Broker, and key authority. The MQTT protocol version 3.1.1 has been utilized [57]. MQTT Publisher and Subscriber nodes are implemented utilizing the Eclipse Paho Java Client version 1.1.0 [58], while the MQTT broker is implemented using the free source Mosquitto version 2.0.14 [59]. MQTT clients, i.e., Publisher and Subscriber nodes, are present on the same machine, whereas the MQTT broker is shown on a separate machine. Fig. 5 depicts the architecture of the test-bed environment setup that has been designed.

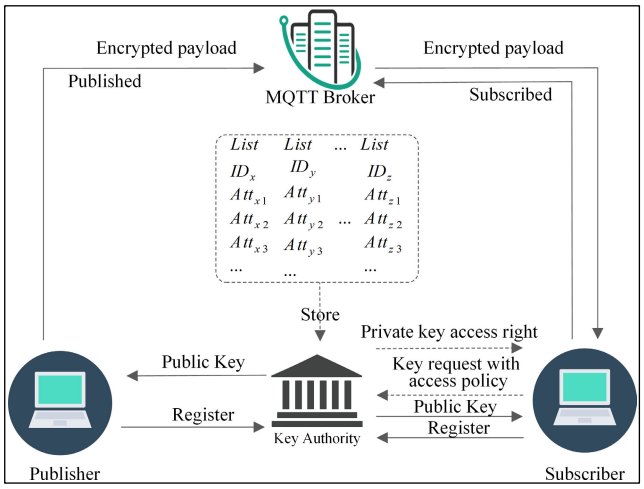

FIGURE 5. Test-bed Environment setup.

This test-bed demonstrates a high-level architecture of the proposed cryptosystem, which will identify the critical components involved and their interdependencies. Besides, it represents a standard MQTT topology, with Publishers and Subscribers nodes interacting simultaneously.

MQTT Publisher, Broker, Subscriber, and key authority are four elements that make up the MQTT test-bed environment. Moreover, the key authority serves as the key element in the proposed scheme for the MQTT nodes involved in this implementation in order to generate the public parameters as well as the decryption key for the MQTT subscriber. Besides, partial decryption is achieved by the key authority of the proposed RSS because the decryption process of the KP-ABE algorithm will request the decryption key from the authority and the authority will perform the keygen algorithm which is run on the authority. Furthermore, as the MQTT protocol has a centralized structure, which means that every node does have a direct dependence mainly on the broker, hence it is the most essential element in the protocol implementation.

IX. EXPERIMENTAL RESULTS AND DISCUSSION

This section illustrates the findings of the implementation work which was based on the proposed hybrid cryptosystem to protect the MQTT data in the IoT. The proposed methodology was evaluated following standardized evaluation criteria. Two different scenarios are explained: the ordinary scenario and the upgraded scenario. This is to ensure that all the design parameters of the proposed symmetric D-AES algorithm, as well as the proposed secure hybrid cryptosystem, are satisfied as was previously stated. Consequently, an ordinary scenario was performed to measure the proposed secure hybrid cryptosystem performance in terms of execution time, traffic overhead, and memory usage. The achieved findings were compared to those existent methods as well as to standard MQTT in order to justify the deployment of the proposed Hybris Scheme to secure the MQTT protocol in the IoT. Similarly, an upgraded scenario of the enhanced D-AES symmetric algorithm was performed to measure the strength of the proposed D-AES algorithm in terms of balance, hamming distance, and avalanche effect criterion in addition to the performance analysis of the algorithm in terms of processing time. The achieved findings were compared to those of previous works as well as to conventional AES in order to justify the implementation of the enhanced D-AES symmetric algorithm, which would be employed in the proposed secure hybrid system.

A. KNOWN ANSWER TEST

The NIST Cryptographic Algorithm Validation Program (CAVP) provides validation testing of Approved (i.e., FIPS-approved and NIST-recommended) cryptographic algorithms and their individual components. Cryptographic algorithm validation is a prerequisite of cryptographic module validation. Therefore, the enhanced symmetric algorithm has been analyzed by utilizing the Known Answer Test (KAT). Bunches of plaintexts, ciphertexts, as well as keys, are utilized by known answer test vectors to evaluate the efficiency of AES implementations, as the name implies. By utilizing the samples of the KAT vectors, non-linearity, as well as the security strength of the enhanced D-AES algorithms, have been investigated by means of hamming

distance, balance, and avalanche effect rate, in which the plaintext is randomized and has a key value of zero. Table 5 presents the KAT vectors utilized throughout the security analysis.

B. BALANCE CRITERIA

The concept of balance is a fundamental precept of cryptographic schemes, and being crucial in establishing the non-linearity of Boolean processes. The possibility of zeros and ones is almost equivalent in the balanced function's output.

TABLE 5. Known Answer test vectors for analysis.

No. of samples	The original text and its ciphertext counterpart
PT-CT-1	F34481EC 3CC627BA CD5DC3FB 08F273E6
	0336763E 966D9259 5A567CC9 CE537F5E
PT-CT-2	9798C464 0BAD75C7 C3227DB9 10174E72
	A9A1631B F4996954 EBC09395 7B234589
PT-CT-3	96AB5C2F F612D9DF AAE8C31F 30C42168
	FF4F8391 A6A40CA5 B25D23BE DD44A597
PT-CT-4	6A118A87 4519E64E 9963798A 503F1D35
	DC43BE40 BE0E5371 2F7E2BF5 CA707209
PT-CT-5	CB9FCEEC 81286CA3 E989BD97 9B0CB284
	92BEEDAB 1895A94F AA69B632 E5CC47CE
PT-CT-6	B26AEB18 74E47CA8 358FF223 78F09144
	459264F4 798F6A78 BACB89C1 5ED3D601
PT-CT-7	58C8E00B 2631686D 54EAB84B 91F0ACA1
	08A4E2EF EC8A8E33 12CA7460 B9040BBF

Table 6. exhibits the balance rates of the standard algorithm ciphertext as well as the enhanced algorithm ciphertext for the plaintext that is presented in Table 5. On average, as compared to the conventional AES, the enhanced key expansion of the D-AES provides a better balance of 1s for the plaintext vectors in Table 5. A comparison is made between the ciphertext's overall balance in reference to a standard size of 128 bits for both the conventional symmetric algorithm and the proposed algorithm enhancements. The ciphertext that is generated by the conventional symmetric algorithm has a lower balance, where the algorithm achieved only 46.09% for plaintext sample 7 and 48.44% for sample 2.

TABLE 6. Balance comparison of enhanced symmetric D-AES.

		Avalanche effect of the proposed enhancements		
KAT samples	Conventional AES (%)	Enhanced key expansion (%)	key expansion with ShiftRow (%)	Enhanced D-AES(%)
1	50.78	46.09	50.00	53.13
2	52.34	53.13	58.59	54.69
3	51.56	49.22	50.78	57.03
4	45.31	46.88	53.13	57.81
5	43.75	53.13	46.88	51.56
6	49.22	46.88	46.09	51.56
7	50.78	50.00	51.56	53.91
Average	49.11	49.33	51.00	54.24

Considering 128-bit AES, a reasonable balance in the ciphertext ought to be 64 bits. Referring to Table 6, it can be observed that the enhanced Key Expansion obtains an average balance of 53.56% and that the enhanced Key Expansion along with the enhanced ShiftRow Transformation obtains an average balance of 55.13%, and that the overall enhancement obtains a balance on the average by 55.46%. Whereas the conventional algorithm achieves balance on average by 51.00%. As a result, for such plaintext-key pairs as depicted in Table 5, thus, proposed enhancements in the symmetric algorithm have been achieved a better balance compared to the conventional AES.

C. STRICT AVALANCHE CRITERIA

The most significant aspect of the security evaluation process in cryptography is the Strict Avalanche Criteria (SAC); it measures overall complexities for transforming plaintext-into-ciphertext [60]. Besides, when a Boolean function's single input bit is flipped, those resultant bits should be changed within 0.5 probability. All of that is referred to as the strict avalanche criterion. The effect of the avalanche assists to determine the level of diffusion and confusion of a cipher. To specify the degree of change required to obtain the avalanche effect, a plaintext-ciphertext pair has been adopted, simply flipping each bit within plaintext by one, and afterward comparing the resulting ciphertext to the original. The most significant bit (MSB) has been flipped in the plaintext for the analysis. Equation (35) has been used to calculate the avalanche effect.

$$Avalanche\ effect = \frac{change\ in\ ciphertext}{size\ of\ \ ciphertext} \tag{35}$$

TABLE 7. The avalanche effect (%) comparison of the ciphertext.

		Enhanced Algorithm		
PT-CT pair	Standard AES (%)	key expansion (%)	expansion with ShiftRow (%)	D-AES (%)
1	54.69	51.56	51.56	50.00
2	48.44	54.69	50.00	60.94
3	52.34	50.78	53.91	50.00
4	51.56	52.34	61.72	53.91
5	53.91	50.00	57.03	64.06
6	50.00	59.38	50.00	52.34
7	46.09	55.47	61.72	57.03

Table 7 depicts the avalanche's effect. Each of the plaintext samples 1-7 in Table 5 has had the MSB flipped, then the effects of the changes have been analyzed utilizing ciphertexts generated for each algorithm. Table 7 represents the differences changes in ciphertext bits for a 128-bit key length. Strict Avalanche Criterion comparison describes how many bits had been transformed as illustrated in Fig. 6 due to the avalanche in the ciphertext. The MSB has been flipped in plaintext for analysis. Table 7 depicts the avalanche's effect. Each of the plaintext samples 1-7 in Table 5 has had the MSB flipped, then the effects of the changes have been analyzed utilizing ciphertexts generated for each algorithm.

Table 7 represents the differences changes in ciphertext bits for a 128-bit key length. Strict Avalanche Criterion comparison describes how many bits had been transformed as illustrated in Fig. 6 due to the avalanche in the ciphertext.

As depicted in Table 7, D-AES attained a greater effect of the avalanche compared to the conventional algorithm

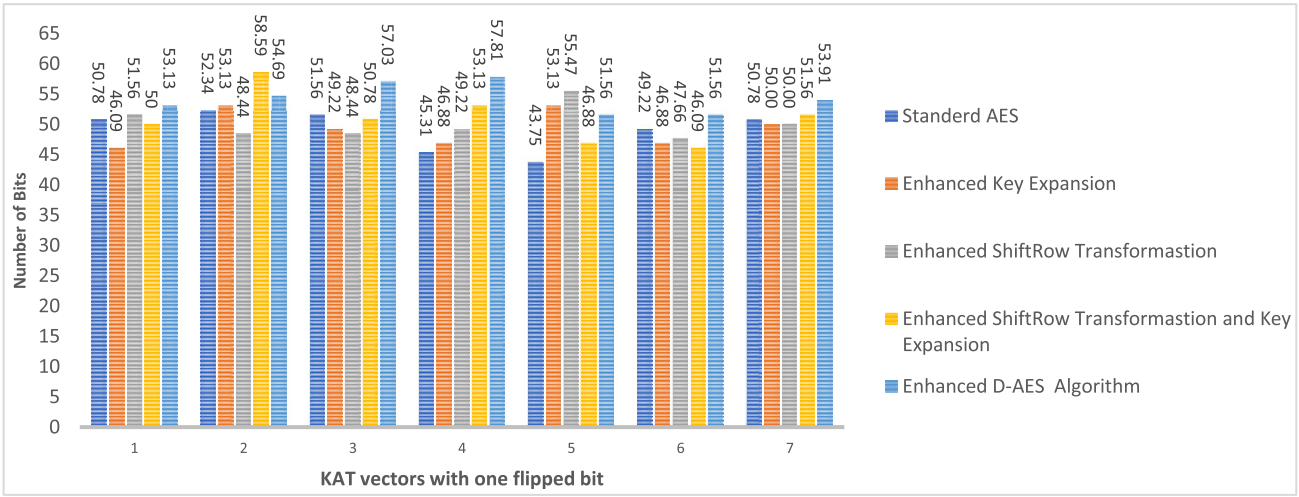

FIGURE 6. Strict Avalanche Criterion comparison of the ciphertext.

for all sample vectors. The enhanced Key Expansion with enhanced ShiftRow Transformation recorded an average SAC of 51.00%, which is much higher than conventional AES which is increasing the avalanche effect by 3.48%, and the overall enhanced symmetric algorithm has an average avalanche effect by 54.24%, as well as higher than the conventional algorithm by increasing the avalanche effect by 9.95%.

D. HAMMING DISTANCE

The nonlinearity could be determined by utilizing the hamming distance which is also a critical metric to quantify the nonlinearity in the cryptosystem. The positions at which the corresponding symbols differ within two Boolean functions are known as the hamming distance.

1) CIPHERTEXT ANALYSIS

The analysis of hamming distance bit-by-bit of the resultant ciphertexts has been performed in order to quantify the nonlinearity in the enhanced cryptosystem. Thus, the hamming distance between the resultant ciphertexts via enhanced symmetric encryption and the conventional AES is calculated bit by bit. Fig. 7 illustrated the analysis of hamming distance bit-by-bit. The distance has been determined between the ciphertext of the standard algorithm for the vectors 1-7 and the ciphertext resulting from the enhanced algorithm. According to Fig. 7, the enhanced symmetric algorithm scored an average Hamming distance by 67.14 bits. Moreover, Key Expansion enhancement, ShiftRow enhancement, Key Expansion with ShiftRow enhancements, and overall enhanced D-AES algorithm have scored a hamming distance of 63.14%, 64.14%, 55.29%, and 67.14% respectively.

It can be observed that the enhancements made in the block cipher of the symmetric algorithm achieved a better result in balanced, SAC, and hamming distance improvements.

TABLE 8. Test samples of the round analysis.

No.	Plaintext-samples	Key-samples
PT-1	F34481EC 3CC627BA	00000000 00000000
	CD5DC3FB 08F273E6	00000000 00000000
	F34481EC 3CC627BA	80000000 00000000
	CD5DC3FB 08F273E6	00000000 00000000
PT-2	00000000 00000000	10A58869 D74BE5A3
	00000000 00000000	74CF867C FB473859
	00000000 00000000	90A58869 D74BE5A3
	00000000 00000000	74CF867C FB473859

The proposed enhancements perform better than the standard algorithm in terms of balance, there exists a 4.82%, 8.09%, and 8.74% difference for the enhanced unit of the key expansion, Key Expansion with ShiftRow transformation, and Key Expansion with ShiftRow and SubByte transformation respectively. While in terms of the avalanche effect, there exists a 0.44%, 3.84%, and 10.44% difference for the enhanced unit of the key expansion, the unit of the key expansion along with ShiftRow transformation, and the unit of the key expansion along with ShiftRow and SubByte transformation respectively.

Furthermore, the results of the enhanced key expansion have been analyzed and compared to those others enhancements of the algorithm in terms of hamming distance. it has been observed that there is a significant improvement of 3.40% difference within an enhanced unit of the key expansion along with ShiftRow transformation, and a 6.33% difference within an enhanced unit of the key expansion along with ShiftRow and SubByte transformation. Besides, it has been observed that there exists a 2.83% difference within an enhanced unit of the key expansion along with ShiftRow and SubByte transformation compared within an enhanced unit of the key expansion along with ShiftRow transformation.

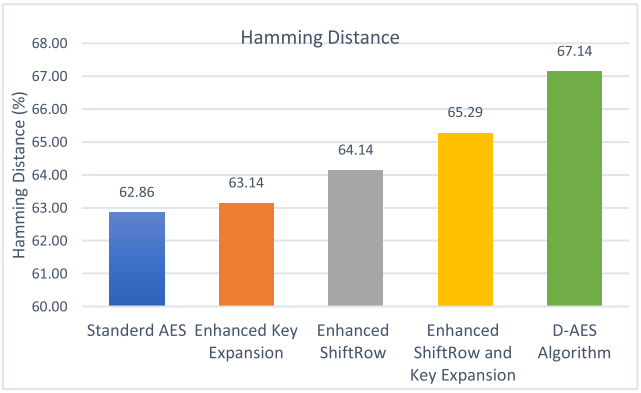

FIGURE 7. Hamming distance (%) of the Enhanced Algorithm.

2) EXPANDED KEY ANALYSIS

The results of the enhanced key expansion for the internal round are analyzed compared to those of the standard algorithm. To investigate the impact of the change of the key expansion unit in the internal rounds, keys, as well as plaintext, have been utilized.

The avalanche effect has been determined by analyzing the results of consecutive key expansion rounds by utilizing plaintext-key pairs and flipping the highest-order place of the binary number in the plaintexts and keys. Particularly, for all of these intermediate key investigations, the KAT vectors with such as zero plaintext-variable key and variable plaintext-zero key as exhibited in Table 7 are employed. Two scenarios have been employed, the first one is variable key with zero plaintext and the second one is zero key with variable plaintext. The effect of the avalanche has been found by flipping the MSB of the encryption key which is the PT Key pair1 and flipping the MSB of zero plaintext which is the PT Key pair2. Table 9 presents that the enhanced method has a better avalanche effect for each round of expansion unit, exhibiting greater confusion for PT key pair1 as well as PT key pair2 when compared to the standard algorithm. A better avalanche has been achieved due to the proposed enhancements techniques which illustrate an adequate degree of confusion. Fig. 8 and 9 depict the change in the percentage of the avalanche in the enhanced algorithm.

X. PERFORMANCE ANALYSIS

The main goal of this section is to calculate the overall processing time including the implementations of standard MQTT as well as the proposed method.

A. PROCESSING TIME

Since the standard implementations of the MQTT do not possess any security protections. Hence, the processing time of the protocol is definitely the shortest. Regardless, the overall processing time has been measured on both sides of the MQTT nodes i.e., Publisher and Subscriber. For example, the processing time has been measured to the overall publication message as well as the subscription upon receipt. Within that way, the computation time of a Publish/Subscribe is thus determined in two distinct phases, once during the publication phase and once during the subscription phase of a message's lifecycle. Based on the findings of the experiments, the proposed methodology processing time is taking more time than that of standard implementations of the MQTT protocol. This is due to the fact that the MQTT payload was encrypted utilizing the improved symmetric D-AES algorithm, and then the D-AES key was encrypted utilizing the KP-ABE algorithm,

TABLE 9. Avalanche effect comparison of the internal rounds.

PT-CT pair	PT-1 (Variable plaintext - zero key)		PT-2 (Zero plaintext - variable key)	
	Standard AES	Enhanced key expansion	Standard AES	Enhanced key expansion
Init. Key	0.78	0.78	0.78	0.78
Round-1	3.13	3.13	3.13	10.94
Round-2	10.94	10.94	20.31	20.31
Round-3	21.88	12.50	17.19	20.31
Round-4	28.91	44.53	22.66	41.41
Round-5	39.84	50.00	21.88	48.44
Round-6	32.03	49.22	26.56	53.91
Round-7	38.28	43.75	24.22	39.06
Round-8	35.16	49.22	30.47	51.56
Round-9	22.66	61.72	25.78	38.28
Round-10	29.69	37.50	27.34	40.63

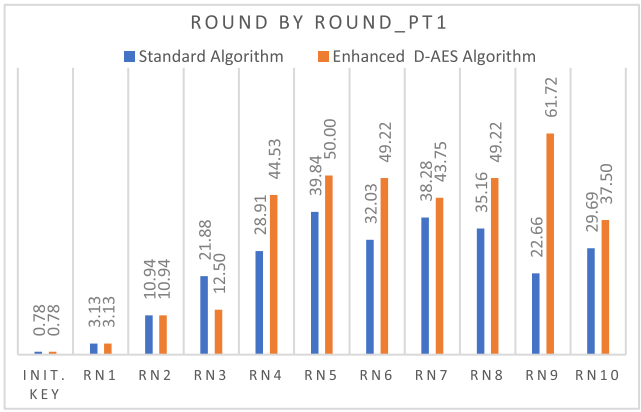

FIGURE 8. The avalanche effect between expanded keys of PT1.

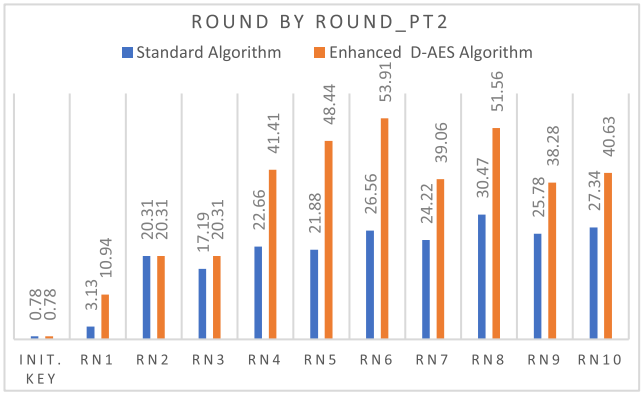

FIGURE 9. The avalanche effect between expanded keys of PT2.

which raised the computational cost to achieve the desired level of security; nevertheless, the significant advantage is somewhat secure MQTT data as well as protection against cyberattacks.

Moreover, the overall processing times for standard MQTT and the proposed hybrid scheme are depicted in Fig. 10, respectively. Due to the implementation of MQTT does not include any security options by default, it will take less processing time, which is almost one millisecond on average. The time needed for publishing/subscribing data is computed including both sides. Typically, the Dispatching messages consist only of the header.

Regardless, data packets differ according to the MQTT payload as well as the data to be obtained. The computation time was recorded by all MQTT messages utilizing the quality of service zero level. The computation time per message has been specified by dispatching nodes as well as the subscription nodes. That is because the dispatcher calculates the ciphered payload such that it may be securely transmitted to the endpoint and deciphered by the recipient. whilst it is essential to calculate the execution time of the same message on the endpoint in order to estimate the difference time between dispatching and subscribing methods.

The findings exhibit that such processing time for dispatching messages is higher than for subscribing messages. This is because the publisher is responsible for encrypting and packing the messages before dispatching them. But in the case of the RSS, the result indicates that the decryption in the subscriber node takes longer to process than encryption in the publish node because the decryption process needs to get the private key associated with the subscriber access policy which is required to run keygen algorithm for generating the privet key of the hybrid scheme which will add more time to process.

Tables 10 and 11 depict the differences in descriptive statistical findings for the conventional MQTT and the proposed RSS, accordingly. Each conventional MQTT message has a maximum computation time of slighter than 0.5 milliseconds. It is evident that it incurs no overhead as it does not transmit any data other than the MQTT payload. Conversely, for that very same messages, the encrypted payload with standard KP-ABE and AES has, on average, 177.1 and 270.15 milliseconds overhead for the MQTT publisher and subscriber respectively. Moreover, the proposed scheme with an enhanced symmetric algorithm has, on average, 204.45 and 297.7 milliseconds overhead for MQTT publisher and subscriber respectively, as a result of the protections introduced to stop cyberattacks. In addition, Fig. 11 depicts the total average execution time for two operations Encryption and Decryption for the proposed approach.

The encryption operation of the proposed approach includes the encryption time of two cryptography algorithms, the first one is utilized to encrypt the MQTT data payload by employing the enhanced AES algorithm, and the second algorithm is used to encrypt the secret key of the enhanced symmetric algorithm by adapting the KP-ABE scheme.

Besides, the decryption operation also possesses the decryption time of both algorithms the KP-ABE as well as the enhanced symmetric algorithm. The results were achieved using an average of twenty iterations for each operation and using 10 attributes. As expected, the time required to complete each operation grows linearly relying on how many attributes are utilized. With up to ten attributes, most operations take less than a second, apart from the decryption key generation. Therefore, MQTT nodes will not notice any significant delay as a result of ABE, because most of the IoT applications utilize less than ten attributes, as well as the decryption key in this work, is generated by the key authority. Besides, in the proposed work, the KP-ABE encrypts only the D-AES key and the actual MQTT payload is encrypted by the D-AES algorithm.

TABLE 10. Standard MQTT: processing time (in milliseconds).

	MQTT Standard		Existing KP-ABE scheme	
	Publisher	Subscriber	Publisher	Subscriber
Min	140	137	328	395
Max	191	180	362	475
Mean	155.75	152.45	340.05	430.4
St. Dev.	12.34	11.16	10.05	27.23
Overhead	-	-	184.3	277.95
Overhead %	-	-	54.20%	64.58%

TABLE 11. Proposed Scheme: Computation time (in milliseconds).

	MQTT Standard		RSS	
	Publisher	Subscriber	Publisher	Subscriber
Min	140	137	321	418
Max	191	180	358	479
Mean	155.75	152.45	343.45	443.75
St. Dev.	12.34	11.16	10.63	16.32
Overhead	-	-	187.7	291.3
Overhead %	-	-	54.65%	65.65%

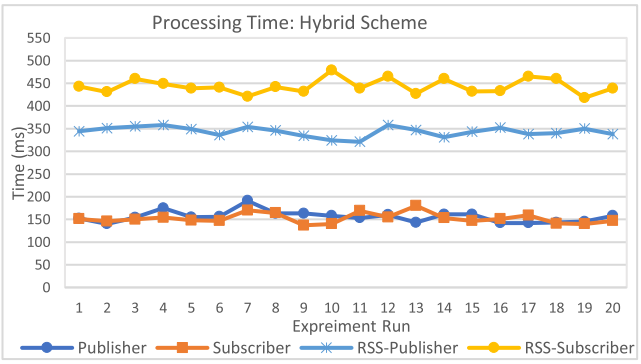

FIGURE 10. Hybrid Scheme: processing time.

B. TRAFFIC OVERHEAD

The traffic overhead will be covered within this section which is associated with the suggested methodology for enhancing

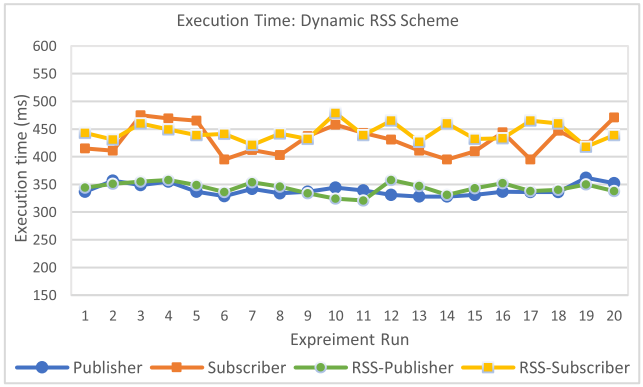

FIGURE 11. Average execution time of the dynamic secure scheme.

the security of MQTT interactions. As indicated earlier, a high density of data packaging of MQTT has little control overhead. The entire message size ware used to determine the traffic overhead of the protocol. The number of interacted messages in the MQTT network depends on employed QoS levels; where the number of messages, as well as the data size, will be increased depending on the QoS level. As demonstrated in Fig. 12, the total data size of the interacted messages in the MQTT network for the proposed scheme is increased linearly relying on the QoS level. However, the proposed approach incurred 229 bytes of traffic overhead for every communication that is sent between the two devices at each QoS level. Nevertheless, there is an obvious tradeoff which is secured transmission. The traffic overhead of MQTT has been measured on both sides of the publisher node as well as the subscriber node to verify the efficiency of the proposed enhancements. Nevertheless, the suggested scheme has introduced an overhead within the MQTT message, the encrypted payload includes three parts and its size is increased linearly reliant on the number of attributes used to generate the ciphertext to prevent modifications and attacks as exhibited in Fig. 13. Furthermore, the basic MQTT has minimal overhead because it is without any security mechanisms; as a result, the data packing density is the greatest. The overhead is shown to decrease with increasing payload size and hence has a minimal impact on the performance.

Regarding the lifetime of the IoT nodes, this research focuses on IoT nodes that do not use batteries as a power source. However, the power consumption is almost inversely related to the processing time [55], [61]. This is expected, given that all ABE cryptosystem processes need almost the same computational power, and so execution time is correlated with power consumption. Furthermore, IoT nodes need to execute encryption and decryption, whereas power-consuming operations that require a lot of power, such as decryption key generation, are handled by the authority, relieving the strain on IoT nodes. Besides, the power consumption of IoT nodes with encrypted payloads will be greater than with unencrypted payloads. This difference is due to the increased CPU consumption required to encrypt the transferred payloads [62]. Consequently, as the time required for encryption and decryption in RSS increases linearly with the number of attributes, the power consumption will also be increased, but the trade-off is secure communication.

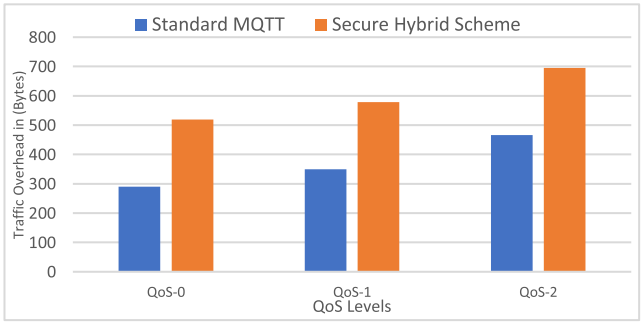

FIGURE 12. The data size of the proposed RSS in the MQTT network.

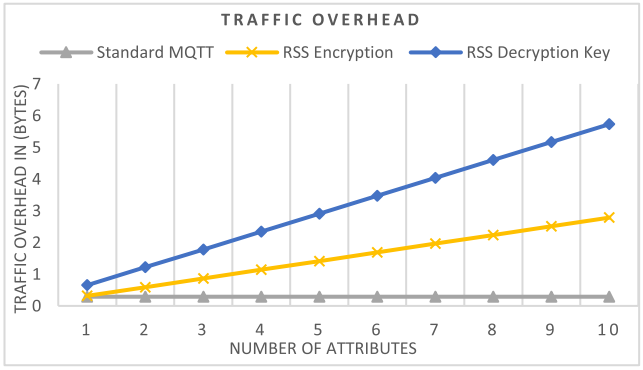

FIGURE 13. Traffic overhead of proposed RSS based on the number of attributes.

XI. COMPARATIVE ANALYSIS

The comparative analysis of the proposed RSS along with the proposed D-AES algorithm to secure MQTT data in IoT networks was done in terms of processing time and traffic overhead as discussed in the next.

A. PROCESSING TIME

The experiments and the obtained results were presented in Section VIII. Based on the results, standard MQTT consumes less processing time compared to the existing KP-ABE scheme [27], [55], and proposed an RSS during the process of the publish-subscribe process in the MQTT network. The reason for this result is logical since the standard MQTT does not use any authentication and encryption mechanisms, thus resulting in lower total processing time compared to the existing KP-ABE scheme [27], [55], and proposed RSS.

Moreover, the proposed RSS consumes a slight increase in the execution time of encryption and decryption compared to the existing KP-ABE scheme [55] during the process of publishing and subscribing of MQTT messages. It is found that the existing KP-ABE scheme [55] increases the average

total processing time by 54.20% and 64.58% milliseconds overhead for MQTT publishers and subscribers respectively. In contrast, the proposed secure mechanism increases the average total processing time overhead by only 54.65% and 65.65%. The proposed RSS has a slight increase in the execution time overhead different by 0.99% and 3.01% than the existing KP-ABE scheme [55] for MQTT publishers and subscribers respectively, that is due to the RSS utilizing the enhanced AES symmetric algorithm instead of a standard algorithm but the proposed scheme provides better security in term of confusion as well as diffusion. The use of a hybrid cryptosystem in the proposed scheme significantly reduces the processing time because the hybrid cryptosystem encrypts the MQTT payload utilizing the proposed symmetric algorithm and the key of the symmetric algorithm will be encrypted by the KP-ABE algorithm to avoid the computation processing overhead resulting by bilinear maps of KP-ABE algorithm, which reduces energy consumption, requires fewer hardware resources and increases the throughput of the mechanism. Therefore, the proposed RSS is still viable for resource-constrained nodes compared to the existing KP-ABE scheme [55] and the scheme of [27].

Furthermore, the existing KP-ABE scheme is more vulnerable to attacks such as differential and linear cryptanalysis as well as interpolation, related key, and square attacks, because the existing KP-ABE scheme employs the standard AES with KP-ABE and according to [52] and [53], the key schedule process of the standard AES algorithm suffers vulnerabilities due of the direct relationships between sub-key bytes that are created as a consequence of the key schedule operation, an adversary can be utilized in the round of the algorithm to crack the key.

Comparing the computing time incurred in encryption and decryption, as these two parts have a direct impact on the user experience. Fig. 14 and 15, respectively show the comparison of encryption and decryption time with a different number of attributes. From Fig. 14 and 15, it can see that the proposed RSS has a slight increase in the execution time of encryption and decryption, owing to the security features included to prevent cyber-attacks. Besides, the attribute authority helps the data user doing part of the decryption, leaving only a small amount of computation overhead to the data user.

In a word, the proposed scheme effectively improves efficiency and is more suitable for practical application in IoT systems.

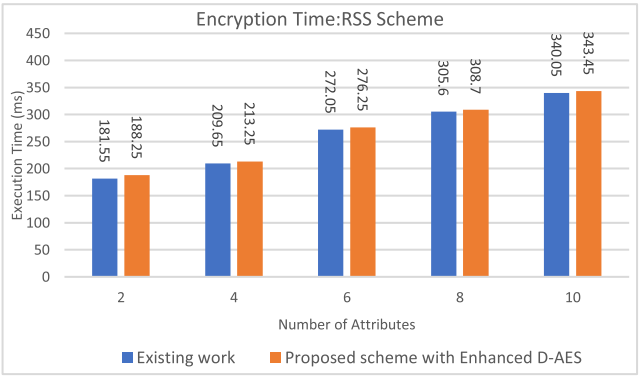

FIGURE 14. The encryption time with a different number of attributes.

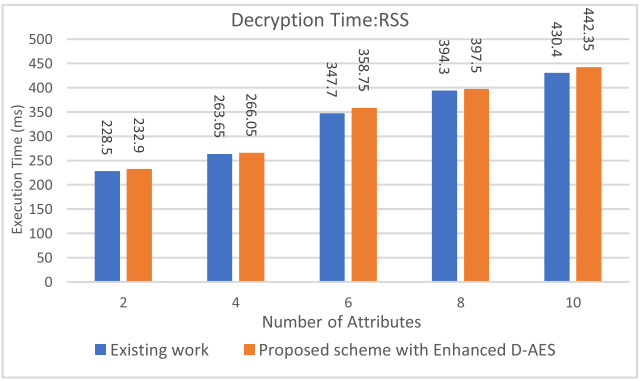

FIGURE 15. The decryption time with a different number of attributes.

B. AVALANCHE EFFECT

In order to ensure that the enhanced symmetric algorithm has a better security level, the obtained experimental result was compared with existing work such as; [27], [42], [44], [45], [47], and [63] in term of avalanche effect during the encryption process. Few researchers have employed the avalanche effect to measure the performance of their algorithms. Therefore, it is recommended that more researchers should use it, as it is a desirable property of encryption algorithms. However, the comparative results have shown that the enhanced symmetric algorithm is achieved a significant improvement in terms of the avalanche effect. Fig. 16 depicts the comparative results on all existing work i.e., [27], [42], [44], [45], [47], [63] in terms of avalanche effect measurement in different modifications of the AES algorithm.

The comparison results signify that the proposed enhancements of the symmetric AES algorithm are working efficiently in terms of the avalanche property compared with the existing work. In contrast, the proposed enhancements of the symmetric algorithm recorded an average of 54.24, compared with the standard algorithm with an average of 49.11 which is higher than standard AES by 10.45% improvements. Moreover, the proposed enhancements of the symmetric algorithm achieved a higher degree of avalanche effect compared [27], [42], [44], [45], [47], [63]. The proposed enhancements display a higher avalanche which illustrates a good degree of confusion being obtained by the techniques.

It can be observed that the proposed enhancements of the symmetric algorithm have a 10.45% increase in avalanche effect for the entire ten rounds when compared to the standard algorithm as well as existing works, there is 0.20%, 0.13%, 0.58%, 4.79%, 5.88%, 5.10%, and 2.63% difference increase [27], [42], [63], [45], [44], and [47], respectively, which are all achieved less degree of avalanche effect compared

to the proposed enhancements, that indicates the proposed changes will make the algorithm very difficult to decrypt the ciphertext without proper decryption key and control bit information.

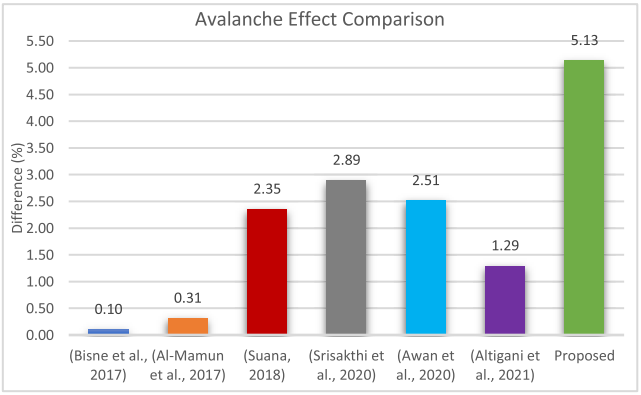

FIGURE 16. Avalanche effect differences comparison in different modifications.

XII. LIMITATIONS AND FUTURE RESEARCH

A number of future works outside the scope of this research have been identified. These directions for future research are recommended in order to overcome this research limitation for enhancing the performance of the proposed secure hybrid scheme. In future work, similar manipulations can be suggested to the AES in order to attain better performance and security without increasing the implementation cost.

Moreover, the secure hybrid cryptosystem scheme is designed for securing MQTT devices that are employed in IoT networks. However, an internal/external publisher and a subscriber can also exploit to cause various cyber-attacks on the MQTT broker. Since this research work is not focusing on the cyber-attacks that exhaust the resources of the MQTT broker. Therefore, the future direction presents a mechanism for MQTT Broker.

Besides, a DDoS attack is one of the most common network attacks that cause a DoS attack on any server such as an MQTT broker. The secure hybrid scheme utilized external authority in this implementation; this authority can be implemented on the same MQTT broker by modifying the broker and then could be used to propose a mechanism to detect the abnormal behavior in the DDoS attack.

Additionally, there is potential for the KP-ABE to be improved such that it is both more secure and a hierarchical scheme that can allow role delegation in the Internet of Things applications, in particular IoT-connected healthcare systems.

Furthermore, Blockchain and other emerging technologies provide intriguing possibilities. However, blockchain technology and symmetric algorithms such as D-AES could be employed to ensure the integrity of MQTT data and the availability of the broker against the most frequent attacks against IoT systems, such as linking attacks, a man in the middle, and Distributed.

Lastly, this study focuses on IoT nodes that do not rely on batteries for the power source. Therefore, future research might investigate the effects of the proposed scheme on battery-powered node-based IoT devices.

XIII. CONCLUSION

In order to address the key expansion limitation of the AES, the key expansion unit is structured to utilize four values of the RCON in each round in order to compensate for this vulnerability. Utilizing these round constants within the key expansion component effectively improved AES' confusion feature. Moreover, strengthening the unit of key expansion has improved the strength of the D-AES. Further, redesign the SubBytes transformation of the AES algorithm in order to ensure that a key change could well be recognized within encrypted messages by making this transformation work dynamically by introducing a new design of this transformation, where all the bytes in the matrix of 128-bits round key has been utilized.

Moreover, a new design of the shift row transformation of the AES algorithm has been introduced in order to make this transformation work dynamically and for improving the algorithm security level in terms of diffusion. the transformation process in this enhancement did not rely on the static offset but became a dynamic transformation process that relied on a dynamic value. The confusion rate has increased when the enhanced method is executed ten times for the 128-bits block to generate the encrypted data as well as increasing its resistance to cryptanalysis.

A new security solution called RSS has been proposed to be adopted on IoT devices in which security feature is augmented to the existing MQTT protocol. The proposed scheme uses two separate cryptosystems i.e., enhanced D-AES and KP-ABE cryptosystems in order to distribute the secret key of the publisher to the subscriber as well as to provide confidentiality of MQTT payload, broadcast encryption, fine-grained access control, and collision resistance.

The experimentation results were examined to evaluate the performance of proposed enhancements of a symmetric algorithm as well as the proposed RSS in order to achieve the research objectives of this thesis. The analysis results showed that the proposed D-AES is more promising with improvements than the standard algorithm, there exists an 8.75%, 10.45%, and 6.81% difference in terms of balance, avalanche effect, and hamming distance, respectively. Additionally, D-AES recorded a slight increase in the average total processing time difference by only 2.16 and 3.21 milliseconds for encryption and decryption, respectively. Besides, the proposed RSS has a slight increase in the execution time overhead difference by 0.99% and 3.01% than the existing scheme, respectively. Furthermore, the basic MQTT protocol has minimal overhead because it is without any security mechanisms, while the proposed scheme increases the traffic overhead by 90.57% for ten attributes, there is an obvious tradeoff which is secured transmission. It could be concluded that the proposed security enhancements effectively improve

the efficiency of the secure scheme compared to the existing works and are more suitable for practical application in IoT systems.

REFERENCES

[1] D. Sehrawat and N. S. Gill, "Smart sensors: Analysis of different types of IoT sensors," in *Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2019, pp. 523–528.

[2] Z. Bi, L. D. Xu, and C. Wang, "Internet of Things for enterprise systems of modern manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1537–1546, May 2014.

[3] B. V. Philip, T. Alpcan, J. Jin, and M. Palaniswami, "Distributed real-time IoT for autonomous vehicles," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1131–1140, Feb. 2019.

[4] G. B. Nils Andersson, "Ericsson mobility report," Ericsson, Stockholm, Sweden, Tech. Rep. EAB-22:010742 Uen Rev D, Nov. 2022.

[5] L. S. Vailshery, *IoT and Non-IoT Connections Worldwide 2010–2025*, vol. 1101442. Hamburg, Germany: Statista, Sep. 2022.

[6] C. Shouqi, L. Wanrong, C. Liling, H. Xin, and J. Zhiyong, "An improved authentication protocol using smart cards for the Internet of Things," *IEEE Access*, vol. 7, pp. 157284–157292, 2019.

[7] J. Soldatos, *OpenIoT: Open Source Internet-of-Things in the Cloud*, vol. 9001. Cham, Switzerland: Springer, 2015.

[8] 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.

[9] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "MQTT vulnerabilities, attack vectors and solutions in the Internet of Things (IoT)," *IETE J. Res.*, pp. 1–30, May 2021.

[10] M. Agrawal, J. Zhou, and D. Chang, "A survey on lightweight authenticated encryption and challenges for securing industrial IoT," in *Advanced Sciences and Technologies for Security Applications*, Cham, Switzerland: Springer, 2019, pp. 71–94.

[11] Z. Tbatou, A. Asimi, Y. Asimi, and Y. Sadqi, "Kerberos V5: Vulnerabilities and perspectives," in *Proc. 3rd World Conf. Complex Syst. (WCCS)*, Nov. 2015, pp. 1–5.

[12] S. Nafie, J. Robert, and A. Heuberger, "SCRAM: A novel approach for reliable ultra-low latency M2M applications," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.

[13] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol—Version 1.2*, document RFC 5246, 2008.

[14] A. J. Hintaw, S. Manickam, S. Karuppayah, and M. F. Aboalmaaly, "A brief review on MQTT's security issues within the Internet of Things (IoT)," *J. Commun.*, vol. 14, no. 6, pp. 463–469, 2019.

[15] L. Malina, G. Srivastava, P. Dzurenda, J. Hajny, and R. Fujdiak, "A secure publish/subscribe protocol for Internet of Things," in *Proc. 14th Interfaces Conf. Availability, Rel. Secur.*, Aug. 2019, pp. 1–10.

[16] S. P. Mathews and R. R. Gondkar, "Protocol recommendation for message encryption in MQTT," in *Proc. Int. Conf. Data Sci. Commun. (IconDSC)*, Mar. 2019, pp. 1–5.

[17] E. Elemam, A. M. Bahaa-Eldin, N. H. Shaker, and M. A. Sobh, "A secure MQTT protocol, telemedicine IoT case study," in *Proc. 14th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2019, pp. 99–105.

[18] A. Kaminsky, M. Kurdziel, and S. Radziszowski, "An overview of cryptanalysis research for the advanced encryption standard," in *Proc. MILCOM Mil. Commun. Conf.*, Oct. 2010, pp. 1310–1316.

[19] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Appl. Sci.*, vol. 9, no. 5, p. 848, Feb. 2019.

[20] 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.

[21] L. Nastase, "Security in the Internet of Things: A survey on application layer protocols," in *Proc. 21st Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2017, pp. 659–666.

[22] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4132–4156, Mar. 2021.

[23] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I. H. Hasbullah, A. A. Bahashwan, and S. Al-Sarawi, "Renyi joint entropy-based dynamic threshold approach to detect DDoS attacks against SDN controller with various traffic rates," *Appl. Sci.*, vol. 12, no. 12, p. 6127, Jun. 2022.

[24] A. Bashir and A. H. Mir, "Securing publish-subscribe services with dynamic security protocol in MQTT enabled Internet of Things," *Int. J. Secur. Appl.*, vol. 11, no. 11, pp. 53–66, Nov. 2017.

[25] O. Khomlyak, "An investigation of lightweight cryptography and using the key derivation function for a hybrid scheme for security in IoT," M.S. thesis, Dept. Comput. Sci., Blekinge Inst. Technol., Karlskrona Sweden, 2017.

[26] M. Usman, I. Ahmed, M. Imran, S. Khan, and U. Ali, "SIT: A lightweight encryption algorithm for secure Internet of Things," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 1, pp. 1–10, 2017.

[27] L. Bisne and M. Parmar, "Composite secure MQTT for Internet of Things using ABE and dynamic S-box AES," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, Apr. 2017, pp. 1–5.

[28] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Oct. 2006, pp. 89–98.

[29] B. S. Adiga, P. Balamuralidhar, M. A. Rajan, R. Shastry, and V. L. Shivraj, "An identity based encryption using elliptic curve cryptography for secure M2M communication," in *Proc. 1st Int. Conf. Secur. Internet Things*, Aug. 2012, pp. 68–74.

[30] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 746–751.

[31] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Secur. Symp.*, no. 3, 2011, pp. 1–16.

[32] M. S. Taha, M. S. M. Rahim, S. A. Lafta, M. M. Hashim, and H. M. Alzuabidi, "Combination of steganography and cryptography: A short survey," *IOP Conf. Mater. Sci. Eng.*, vol. 518, no. 5, May 2019, Art. no. 052003.

[33] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security," *Optik*, vol. 127, no. 4, pp. 2341–2345, Feb. 2016.

[34] J. Saleem and M. Hammoudeh, "Defense methods against social engineering attacks," in *Computer and Network Security Essentials*, K. Daimi, Ed. Detroit, MI, USA: Springer, 2018, pp. 603–618.

[35] A. Abdulgader, M. Ismail, N. Zainal, and T. Idbeaa, "Enhancement of AES algorithm based on chaotic maps and shift operation for image encryption," *J. Theor. Appl. Inf. Technol.*, vol. 71, no. 1, pp. 1–12, 2015.

[36] A. Altigani and B. Barry, "A hybrid approach to secure transmitted messages using advanced encryption standard (AES) and word shift coding protocol," in *Proc. Int. Conf. Comput., Electr. Electron. Eng. (ICCEEE)*, Aug. 2013, pp. 134–139.

[37] S. Suri and R. Vijay, "An AES-CHAOS-based hybrid approach to encrypt multiple images," in *Recent Developments in Intelligent Computing Communication and Devices*, S. Patnaik, Ed. Bhubaneswar, Odisha, India: SOA University, 2017, pp. 37–43.

[38] P. Singhai and A. Shrivastava, "An efficient image security mechanism based on advanced encryption standard," *Int. J. Adv. Technol. Eng. Explor.*, vol. 2, no. 13, p. 175, 2015.

[39] R. Hosseinkhani and H. H. S. Javadi, "Using cipher key to generate dynamic S-box in AES cipher system," *Int. J. Comput. Sci. Secur.*, vol. 6, no. 1, pp. 19–28, 2012.

[40] K. Kazlauskas and J. Kazlauskas, "Key-dependent S-box generation in AES block cipher system," *Informatica*, vol. 20, no. 1, pp. 23–34, Jan. 2009.

[41] G. N. Krishnamurthy and V. Ramaswamy, "Making AES stronger: AES with key dependent S-box," *Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 9, pp. 388–398, 2008.

[42] A. Al- Mamun, S. S. M. Rahman, T. Ahmed Shaon, and M. Hossain, "Security analysis of AES and enhancing its security by modifying S-box with an additional byte," *Int. J. Comput. Netw. Commun.*, vol. 9, no. 2, pp. 69–88, Mar. 2017.

[43] J. Daemen and V. Rijmen, "AES proposal: Rijndael," NIST AES Proposal, Gaithersburg, MD, USA, Tech. Rep. NIST-FIPS-197, 1998.

[44] I. A. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, "Secure framework enhancing AES algorithm in cloud computing," *Secur. Commun. Netw.*, vol. 2020, pp. 1–16, Sep. 2020.

[45] S. Srisakthi and A. P. Shanthi, "Towards the design of a stronger AES: AES with key dependent shift rows (KDSR)," *Wireless Pers. Commun.*, vol. 114, no. 4, pp. 3003–3015, Oct. 2020.

[46] H. N. Noura, A. Chehab, and R. Couturier, "Efficient & secure cipher scheme with dynamic key-dependent mode of operation," *Signal Process., Image Commun.*, vol. 78, pp. 448–464, Oct. 2019.
[47] A. Altigani, S. Hasan, B. Barry, S. Naserelden, M. A. Elsadig, and H. T. Elshoush, "A polymorphic advanced encryption standard—A novel approach," *IEEE Access*, vol. 9, pp. 20191–20207, 2021.
[48] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1948.
[49] S. Khan, P. Iqra University Karachi, M. S. Ibrahim, M. Ebrahim, and H. Amjad, "FPGA implementation of secure force (64-bit) low complexity encryption algorithm," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 12, pp. 60–69, Nov. 2015.
[50] S. Khan, M. S. Ibrahim, H. Amjad, K. A. Khan, and M. Ebrahim, "FPGA implementation of 64 bit secure force algorithm using full loop-unroll architecture," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2015, pp. 1–6.
[51] M. Ebrahim and C. Wai Chong, "Secure force: A low-complexity cryptographic algorithm for wireless sensor network (WSN)," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng.*, Nov. 2013, pp. 557–562.
[52] C. Bouillaguet, P. Derbez, O. Dunkelman, P. Fouque, N. Keller, and V. Rijmen, "Low-data complexity attacks on AES," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 7002–7017, Nov. 2012.
[53] O. Dunkelman and N. Keller, "The effects of the omission of last round's MixColumns on AES," *Inf. Process. Lett.*, vol. 110, nos. 8–9, pp. 304–308, 2010.
[54] B. S. Adiga, M. A. Rajan, R. Shastry, V. L. Shivraj, and P. Balamuralidhar, "Lightweight IBE scheme for wireless sensor nodes," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2013, pp. 1–6.
[55] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the IoT," in *Proc. IEEE Interfaces Conf. Commun. (ICC)*, Jun. 2014, pp. 725–730.
[56] M. Ambrosin, M. Conti, and T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices," in *Proc. Workshop IoT Challenges Mobile Ind. Syst.*, May 2015, pp. 49–54.
[57] O-Standard. (2014). *MQTT Version 3.1. 1*. Accessed: Apr. 3, 2023. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html
[58] P-Client. (2015). *Eclipse Paho Client*. Accessed: Feb. 20, 2022. [Online]. Available: https://www.eclipse.org/paho/index.php?page=downloads.php
[59] (2019). *Eclipse Mosquitto Broker*. Accessed: Apr. 2, 2023. [Online]. Available: https://mosquitto.org/
[60] R. Forrié, "The strict avalanche criterion: Spectral properties of Boolean functions and an extended definition," in *Proc. Conf. Theory Appl. Cryptogr*, 1990, pp. 450–468.
[61] V. Seoane, C. Garcia-Rubio, F. Almenares, and C. Campo, "Performance evaluation of CoAP and MQTT with security support for IoT environments," *Comput. Netw.*, vol. 197, Oct. 2021, Art. no. 108338.
[62] F. De Rango, G. Potrino, M. Tropea, and P. Fazio, "Energy-aware dynamic Internet of Things security system based on elliptic curve cryptography and message queue telemetry transport protocol for mitigating replay attacks," *Pervas. Mobile Comput.*, vol. 61, Jan. 2020, Art. no. 101105.
[63] M. V. C. Suana, "Enhancement of advanced encryption standard (AES) cryptographic strength via generation of cipher key-dependent S-box," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 6, no. 4, pp. 1420–1428, Apr. 2018.

AHMED J. HINTAW was born in Karbala, Iraq, in 1986. He received the B.S. degree in computer science in Hefei, in 2009, the M.S. degree in computer science from Jamia Hamdard University (JHU), New Delhi, India, in 2012, and the Ph.D. degree in internet infrastructures security from the National Advanced IPv6 Research Center (NAv6), Universiti Sains Malaysia (USM), in 2022. From 2012 to 2020, he was a Lecturer with the Applied Medical Science College (AMSC), University of Karbala, Iraq, where he has been a Senior Lecturer, since 2020. His research interests include cybersecurity, the Internet of Things, cryptography, and network security.

SELVAKUMAR MANICKAM received the bachelor's and master's degrees in computer science, in 1999 and 2002, respectively, and the Ph.D. degree from Universiti Sains Malaysia (USM), in 2013. He is an Associate Professor and a Researcher with the National Advanced IPv6 Research Centre (NAv6), USM. His research interests include internet security, cloud computing, the IoT, and android and open source technology. He is an Executive Council Member of the Internet Society (ISOC), Malaysian Chapter; and also the Head of the Internet Security Working Group Under Malaysian Research and Education Network (MyREN).

SHANKAR KARUPPAYAH (Member, IEEE) received the B.Sc. degree in computer science from Universiti Sains Malaysia (USM), Malaysia, the M.Sc. degree in software systems engineering from KMUTNB, Thailand, and the Ph.D. degree in cyber security from Technische Universität Darmstadt, in 2016. He is currently a Senior Lecturer and a Researcher with the National Advanced IPv6 Research Centre (NAv6), USM. His main research interests include P2P botnets, distributed systems, and cyber security. He has authored and coauthored many articles in journals, workshops, and conference proceedings. He is a reviewer of many esteemed networks and security journals.

MOHAMMAD ADNAN ALADAILEH received the B.S. degree in computer science from Mutah University, the M.Sc. degree in information technology from Universiti Utara Malaysia (UUM), in 2016, and the Ph.D. degree in internet infrastructures security from University Sains Malaysia (USM). He was a Postdoctoral Fellow with the National Advanced IPv6 Centre (NAv6), USM. He is currently a Lecturer with the Cybersecurity Department, School of Information Technology, American University of Madaba (AUM), Amman, Jordan. His current research interests include computer networks, network security, the Internet of Things (IoT), intrusion detection systems (IDS), intrusion prevention systems (IPS), and software defined networking (SDN).

MOHAMMED FAIZ ABOALMAALY received the bachelor's degree in software engineering from the Mansour University College and the master's and Ph.D. degrees in computer sciences from Universiti Sains Malaysia, Penang, Malaysia. He is currently the Head of the Computer Center and E-Learning, Department of Mathematics, Al-Zahraa University for Women, Iraq. His research interests include parallel computing, cloud computing, computer networks, and the IoT.

SHAMS UL ARFEEN LAGHARI received the B.Sc. (Hons.) and M.Sc. degrees in computer science from the University of Sindh, Jamshoro, Pakistan, and the M.S. degree in computer science from PAF-KIET, Karachi, Pakistan. He is currently pursuing the Ph.D. degree in network security with the National Advanced IPv6 Centre, Universiti Sains Malaysia. His research interests include cybersecurity, industry 4.0, distributed systems, cloud computing, and mobile cloud computing.