1-1-2020

# Scalable and Secure Big Data IoT System Based on Multifactor Authentication and Lightweight Cryptography

Saleh Atiewi
*Al-Hussein Bin Talal University*

Amer Al-Rahayfeh
*Al-Hussein Bin Talal University*

Muder Almiani
*Al-Hussein Bin Talal University*

Salman Yussof
*Universiti Tenaga Nasional*

Omar Alfandi
*Zayed University*

*See next page for additional authors*

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SCALABLE DEEP LEARNING FOR BIG DATA

Received May 22, 2020, accepted May 29, 2020, date of publication June 16, 2020, date of current version June 30, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3002815*

# Scalable and Secure Big Data IoT System Based on Multifactor Authentication and Lightweight Cryptography

**SALEH ATIEWI[1], AMER AL-RAHAYFEH[1], (Member, IEEE), MUDER ALMIANI[2], (Member, IEEE), SALMAN YUSSOF[3], OMAR ALFANDI[4], (Member, IEEE), AHED ABUGABAH[4], (Member, IEEE), AND YASER JARARWEH[5]**

[1]Department of Computer Science, Al-Hussein Bin Talal University, Ma'an 71111, Jordan
[2]Department of Computer Information Systems, Al-Hussein Bin Talal University, Ma'an 71111, Jordan
[3]Department of System and Networking, Tenaga National University, Kajang 43000, Malaysia
[4]College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates
[5]Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan

Corresponding author: Saleh Atiewi (Saleh@ahu.edu.jo)

**ABSTRACT** Organizations share an evolving interest in adopting a cloud computing approach for Internet of Things (IoT) applications. Integrating IoT devices and cloud computing technology is considered as an effective approach to storing and managing the enormous amount of data generated by various devices. However, big data security of these organizations presents a challenge in the IoT–cloud architecture. To overcome security issues, we propose a cloud-enabled IoT environment supported by multifactor authentication and lightweight cryptography encryption schemes to protect big data system. The proposed hybrid cloud environment is aimed at protecting organizations' data in a highly secure manner. The hybrid cloud environment is a combination of private and public cloud. Our IoT devices are divided into sensitive and nonsensitive devices. Sensitive devices generate sensitive data, such as healthcare data; whereas nonsensitive devices generate nonsensitive data, such as home appliance data. IoT devices send their data to the cloud via a gateway device. Herein, sensitive data are split into two parts: one part of the data is encrypted using RC6, and the other part is encrypted using the Fiestel encryption scheme. Nonsensitive data are encrypted using the Advanced Encryption Standard (AES) encryption scheme. Sensitive and nonsensitive data are respectively stored in private and public cloud to ensure high security. The use of multifactor authentication to access the data stored in the cloud is also proposed. During login, data users send their registered credentials to the Trusted Authority (TA). The TA provides three levels of authentication to access the stored data: first-level authentication - read file, second-level authentication - download file, and third-level authentication - download file from the hybrid cloud. We implement the proposed cloud–IoT architecture in the NS3 network simulator. We evaluated the performance of the proposed architecture using metrics such as computational time, security strength, encryption time, and decryption time.

**INDEX TERMS** Big data, cloud computing, Internet of Things, multilevel authentication, lightweight cryptography.

## I. INTRODUCTION

In accordance with the advancement and wide use of Internet of Things (IoT) applications and with the emergence of wireless communication and mobile technologies, IoT and cloud computing have become important concepts. IoT aims to provide connectivity for anything with minimum storage and computing capabilities [1], [2]. Security is a major issue in cloud-integrated IoT, and the user data stored in the cloud requires secure protection [3]. A lightweight multifactor secured smart card-based user authentication is introduced in cloud–IoT applications [4]. Figure 1 shows the architecture for cloud-integrated IoT, which consists of the hybrid cloud, IoT devices, and users. The hybrid cloud includes public and

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

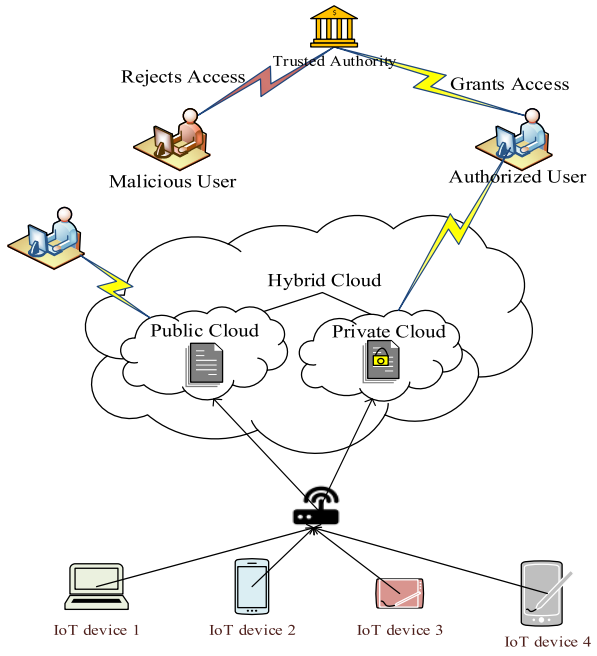


**FIGURE 1.** Architecture for cloud–IoT environment.

private cloud. The public cloud is used to store nonsensitive data, whereas the private cloud is used to store highly sensitive data.

The end-to-end secure communication architecture is proposed for a cloud-connected IoT environment. Herein, a constrained application protocol is proposed for a secure communication between IoT and the cloud [5]. A homomorphic encryption system based on the ring learning with error algorithm is used for cloud user authentication [6]. Role-based access control (RBAC) with the trust evaluation (TE) algorithm is used to provide access control to IoT resources. RBAC involves three TE algorithms, namely, local trust evaluation algorithm, virtual trust evaluation algorithm, and cooperative trust evaluation algorithm [7]. A lightweight IoT-based cryptography authentication scheme is introduced to provide security in a cloud–IoT environment. A proposed lightweight authentication scheme adopts a one-way hash function and exclusive OR operation [8]. An advanced lightweight authentication scheme based on formal and rigorous informal security analysis is proposed for a cloud-assisted IoT environment. Formal security analysis is performed through a random oracle model [9]. A trust-based IoT cloud environment is introduced to provide a secure storage in a cloud environment. The past history of each IoT device is collected using a centralized IoT trust protocol considered for security analysis [10]. A secure and compliant continuous assessment framework (SCCAF) is proposed to protect user data in a cloud-assisted IoT environment. The SCCAF provides guidelines for cloud users in evaluating the security and compliance levels of cloud service providers [11]. Lightweight context-aware IoT services are provided to the user. Moreover, the enacted lightweight context-aware service uses a filter to forward the most relevant data to users on the basis of their context [12]. The fuzzy analytical hierarchical process (FAHP) algorithm is proposed to evaluate the influential factors in IoT. The FAHP provides a satisfactory analysis of tangible factors, namely, security, value, and connectivity [13]. A lightweight bootstrapping mechanism is used for secure IoT services. The Ephemeral Diffie–Hellman Over COSE protocol is used to standardize key agreements in IoT devices [14].

The main aim of the current work is to propose a multilevel authentication scheme that can provide enhanced security in an integrated IoT–cloud environment. The main contributions of this work are summarized as follows:

- It proposes a hybrid cloud consisting of private and public cloud that can improve the security of IoT systems. IoT devices are also divided into sensitive and nonsensitive devices on the basis of the type of data produced.
- The security of sensitive data from sensitive devices is ensured by encrypting them using RC6 and the Fiestel encryption scheme. The encrypted sensitive data are stored in a private cloud via a gateway device to provide high security.
- Nonsensitive data from nonsensitive devices are encrypted through the AES algorithm and then stored in a public cloud via a gateway device.
- To protect cloud-stored data from malicious users, this work proposes a multilevel authentication scheme with trusted authority (TA). The multilevel authentication scheme is subdivided into three levels, however adding (TA) to the proposed Cloud-IoT Environment will result in extra cloud service cost, since the Environment will deal with third party service.
- To prevent malicious users from reading stored files, this work proposes a first-level authentication scheme. At this level, users need to provide their user ID and password to the TA. Then, the TA verifies these credentials against registered credentials. If the verification is successful, then the TA grants the users access to read the files; otherwise, it rejects the request for access.
- To prevent unauthorized users from downloading files, this work provides a second-level authentication scheme in which users need to provide their biometrics, such as fingerprint and retina, to the TA. Then, the TA verifies the given credentials against registered credentials. If the verification is successful, then the TA grants the users access to download files; otherwise, it rejects the request for access.
- The final level of authentication is proposed to protect the data from unauthorized reading and downloading. At this level, users need to provide their user ID, password, and biometrics to the TA. Then, the TA verifies the given credentials against registered credentials. If the verification is successful, then the TA grants the users access to download and read the files from the cloud; otherwise, it rejects the request for access.

## II. ORGANIZATION OF THE PAPER

The rest of the paper is organized as follows. Section 3 background of previous works related to secure cloud–IoT environments. Section 4 describes the security issues in existing cloud-enabled IoT environments. Section 5 presents the proposed technique and compares its performance with those of existing methods. Section 6 evaluates the performance of the proposed architecture and compare it with existing methods. Finally, Section 7 concludes the paper and highlights the future direction of this research.

## III. RELATED WORK

One of the major issues in cloud-integrated IoT is security. Many researchers have attempted to address security issues, and their contributions are summarized in the subsequent paragraphs.

Kazim *et al.* [15] proposed a novel framework for the secure and dynamic access of IoT services in a multicloud environment. To facilitate IoT multicloud collaboration, they designed and implemented a novel protocol on a cloud platform. The framework involves various stages, including service matchmaking, authentication, and SLA management. SLA management offers service execution in an external cloud according to the agreed SLA, and it monitors the operation to verify whether or not cloud providers comply with the agreed upon SLA. Al-Turjman *et al.* [16] proposed a seamless and secure key agreement-based framework in cloud-assisted IoT services. A seamless key approach satisfies security properties using a bilinear pairing method and elliptic curve cryptography techniques. In this work, a mobile sink strategy was introduced for user authentication over cloud-based environments. The proposed framework consists of seven phases, namely, initialization, system registration, system login, authentication, extraction of sensitive information, and secret key updating. In the system registration phase, IoT devices register their user ID and password to the TA. In the system login phase, a user needs to enter their valid smart card into the terminal to provide their credentials, such as identity and secret keys. In the authentication phase, the communication with the cluster head is authenticated. The works in [15] and [16] described the limitation of stored data in raw form, that is, they can be easily forged by malicious users.

Huang *et al.* [17] proposed a secure and fine-grained data access control for IoT devices. IoT devices send data to a cloud server through a wireless channel. To enhance data security and address the privacy issues in IoT, the authors introduced data protection based on ciphertext policy attribute-based encryption (CP-ABE). They also proposed the outsourced encryption and decryption mechanism in hierarchical ABE. This method employs an appropriate delegation mechanism based on hierarchical ABE and involves a TA and a number of independent domains with authorities. Each domain authority requests secret parameters from the TA, which then generates attribute keys for IoT devices. The method reduces the workload of the TA and achieves scalability in large-scale IoT devices. Herewith, the proposed CP-ABE scheme considers other attributes for verification, and thus, the locations of attributes in an access policy are critical.

Alkeem *et al.* [18] proposed a new system for cloud-enabled IoT devices. It comprises two stages: in the first stage, the user updates their healthcare-related data in the cloud for future use; in the second stage, information is retrieved from the cloud, and the cloud server shares the requested data according to user access privileges. The BSN starts collecting different types of data from the patient's body and then encrypts and sends the data to a mobile phone using Bluetooth or NFC technology. After receiving the encrypted message from the BSN, the mobile phone sends an authentication request to the GSM server. The GSM server sends the authentication request to the mobile phone to ensure that the user is authenticated. The mobile phone receives the triplets from the server for authentication. After authentication, the encrypted data with a secret key are sent to the DHA private cloud with an allocated time stamp. The limitation of this work is that the user authentication is not effective because of its few parameters as credentials for verification that make the forging of stored data easy.

Xu *et al.* [19] proposed a fast and parallel search algorithm over public key cipher texts for cloud-assisted IoT. This scheme is based on the computation bilinear Diffie–Hellman assumption in the random oracle model. Without any search trapdoor, no one knows the semantic information of keywords encrypted by searchable cipher texts. Moreover, no hidden relationship among those searchable cipher texts is leaked. With the keyword search trapdoor, only the corresponding hidden relationship is disclosed, and only matching cipher texts can be found. Herein, user authentication is not performed, hence the loss of stored data in the cloud environment.

Mollah *et al.* [20] proposed a secure data storage in cloud-integrated IoT systems. This scheme utilizes secret key encryption and public key encryption. In the proposed technique, all security operations are offloaded to nearby servers; hence, the processing burden is reduced. A searching scheme was also proposed for secure search using authorized users within encrypted, stored, and shared data in the cloud. After completing the search process, verification is initiated, in which case the shared data are retrieved. This scheme ensures the integrity of shared data and search resultant data. Elmisery *et al.* [21] proposed a technique for privacy protection in a cloud-assisted IoT environment. Cloud–IoT services offer opportunities to users to directly communicate with healthcare professionals. Personal IoT devices integrate collected health data at a central cloud service to extract useful information from users. The proposed cloud service relies on a centralized approach in which user data are collected and stored on the server. It also complies with the data privacy laws applicable to the service. The works in [20], [21] are limited because the security of outsourced data is poorly maintained.

Tao *et al.* [22] proposed a security service framework for IoT that is based on a multilayer cloud architecture model. This model enables effective and seamless interactions on devices provided by IoT services. The model comprises private and public cloud to provide high security to user data in cloud storage. Highly secured data are stored in a private cloud, whereas normal data are stored in a public cloud, which can easily be accessed by cloud users. This work has several limitations, such as the absence of authentication for cloud users that may in turn reduce the security of data.

Hao *et al.* [23] proposed a fine-grained data access control with a attribute hiding policy for cloud-based IoT. A fine-grained access control policy was also put forward to support an excessive access policy with full attributes hidden for cloud-based IoT. Herein, attribute-based information is fully hidden using a randomizable technique. A fuzzy attribute positioning mechanism is used to locate the attributes of authorized users efficiently. A garbled bloom filter is used for this process. However, the study's use of the garbled bloom filter causes a high number of false positives, which indicate that an attribute is a member of an access policy group when it is really not.

Belguith *et al.* [24] proposed a secure outsourcing of multi-authority attribute-based encryption with a hidden policy for cloud-assisted IoT. A scheme called policy hidden outsourced attribute-based encryption was proposed to provide a secure storage for user data in cloud-enabled IoT architecture. The use of a hidden access policy provides enhanced security to user attributes and thus improves efficiency in the data sharing mechanism. However, the security of users' stored data was not considered in this work because it focused on user attribute protection.

Liu *et al.* [25] proposed a privacy-preserving raw data collection in a cloud-assisted IoT environment. In the proposed technique, no TA is added to verify authorized users. Individual data are stored in raw format to enhance the stored data value for users. Herein, a cloud server generates a key pair using parameters. Elgamal-based encryption was proposed to ensure the security of data from IoT devices. However, without a TA, storage in a cloud-enabled IoT environment cannot be secured.

## IV. PROBLEM STATEMENT

This section describes the problems in the current cloud-integrated IoT environment. Guan *et al.* [26] proposed efficient and secure data acquisition from cloud-supported IoT. In this work, a CP-ABE scheme was proposed for efficient and secure data acquisition in a cloud-supported IoT environment. The attributes of user information are encrypted using a threshold secret sharing scheme (TSSS). However, this CP-ABE scheme considers numerous attributes for verification; thus, the locations of attributes in the access policy are critical. The TSSS method divides a secret key into ''n'' shares, which are as large as the secret key itself. Therefore, the secret key is compressed before sharing. However, the same is not possible for high-quality data. Ramu [27] proposed a secure cloud framework that is based on a modified CP-ABE (MCP-ABE) scheme and bloom filter. In this MCP-ABE scheme, all the attributes and their values are hidden. The attribute bloom filter (ABF) method is used in data decryption by evaluating the presence of attributes in the access policy and identifying their positions. However, the ABF is prone to false positives, which indicate that an attribute is a member of the access policy when it really is not. Yang *et al.* [28] put forward a file change with security (FCS) proof stored in a cloud service system. In the data modification process, a user sends a request message to the CSP, including the block signature, data block ID, and versions. On the basis of the modification process, data insertion and data deletion are performed in the CSP. In the data modification process, a user sends a request to the CSP, including a block signature, modified data, and data block ID without encryption. This process may lead to the forging of data block information by malicious users. The cloud client user generates a key and encrypts the data using a bilinear pairing method without any authorization. Dhillo and Kalr [29] proposed a multifactor cloud-enabled IoT service scheme. User biometrics, ID, and password are concatenated and provided to users as smart cards. In the registration phase, a data owner sends a request message, along with their name, to the TA. In the data user registration phase, a user sends their ID, password, and biometrics to the cloud server. After the user registration phase, the CSP concatenates the user ID, password, and biometrics. The concatenated information is then sent to the user via the smart card. This smart card can be forged by malicious users. Multiple Secure Hashing Algorithms (SHA) provide an enhanced security and durability level of the Multifactor Authentication systems [30]. The authors in [31] provide a cloud-based framework for agent cooperation to enable collaboration among fog computing devices and form a cooperative IoT service delivery system. A mobility management architecture for the delivery of 5G-IoT Services is presented in [32]. Ridhawi *et al.* [33] proposed a Mobile Edge Computing (MEC) solution that enables node collaboration among IoT devices to provide reliable and secure communication between devices and the fog layer on one hand, and the fog layer and the cloud layer on the other hand.

Existing methods clearly suffer from limitations when used in cloud–IT integration. In the current work, we address these problems using our proposed highly secure authentication and encryption schemes.

## V. PROPOSED WORK

This section explains the proposed multifactor authentication and lightweight cryptography encryption schemes used for the secure integration of the IoT and cloud environment.

### A. SYSTEM OVERVIEW

Our proposed work enhances security through multifactor authentication and cryptography encryption schemes.

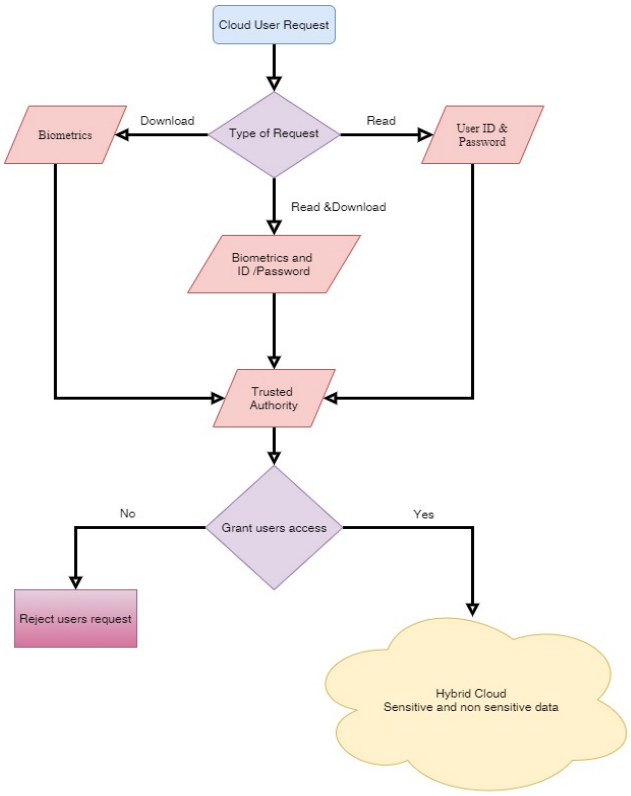


**FIGURE 2.** Procedure of multifactor authentication and lightweight cryptography method

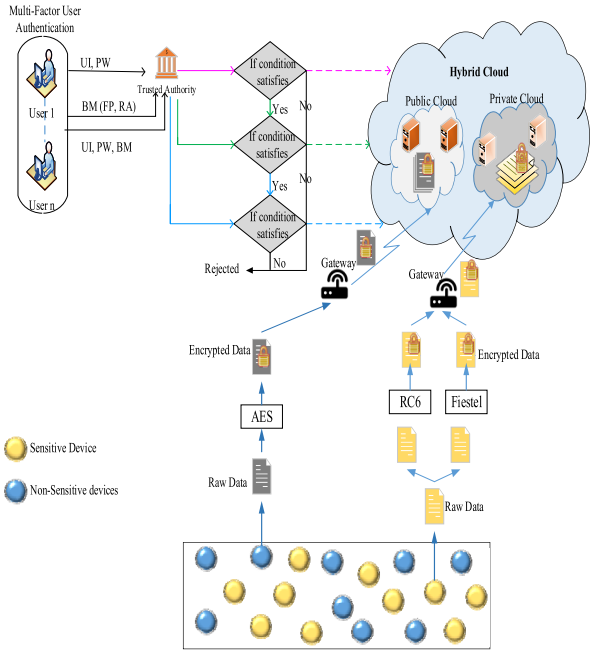


**FIGURE 3.** Architecture for proposed cloud–IoT environment.

As shown in Figure 2, the types of user's requests are categorized into three categories downloading, reading, and both. In case the request is made only for reading contents from the cloud, the user is given permission through the password and user Id. In the case that the user requests to download content from the cloud, the cloud user will be asked for his biometrics, if the user's request is both cases (reading and downloading content) the password and the user name are used in addition to the biometrics. In all scenarios of the request's success, the trusted authority gives the user the permission to access the hybrid cloud otherwise, his request is rejected

The proposed cloud-enabled IoT architecture comprises IoT devices (sensitive devices ($S_1, S_2, \ldots S_n$) and nonsensitive devices ($NS_1, NS_2, \ldots NS_n$)), cloud (private and public cloud), TA, users, and gateway (Figure 3). To protect cloud-stored data from unauthorized users, we provide multifactor authentication to users. Furthermore, we protect data from IoT devices by encrypting the data using RC6 and Fiestel encryption schemes. Sensitive data from sensitive IoT devices are encrypted using RC6 and Fiestel encryption. The encrypted data are stored in a private cloud. We store highly sensitive data in a private cloud to provide high security to stored data. Sensitive data are also encrypted using the two aforementioned schemes to avoid forging. Nonsensitive data from nonsensitive IoT devices are encrypted using the AES algorithm because they contain nonsensitive information that is stored in a public cloud. Sensitive and nonsensitive data are respectively stored in private cloud and public cloud via a gateway device. To provide high security to the stored information, we implement user authentication to access stored files. The TA performs user authentication through registered credentials, such as user ID, password, and biometrics (e.g., fingerprint or retina). The TA provides three levels of authentication when a user reads or downloads a file from the private and public cloud. In the first level of authentication, the TA verifies the username and password to provide read access to the files in the public cloud. The second level of authentication is performed when the user wants to download a file from the public cloud. The user is authenticated via biometrics, such as fingerprint or retina. Lastly, the third level of authentication is performed. The TA receives the user ID, password, and biometrics from the user and then provides them with access to read and download files in the private cloud. Figure 3 shows the proposed architecture for the cloud–IoT environment. The proposed architecture comprises four entities, namely, hybrid cloud, IoT devices, gateway, and TA.

### B. IoT DEVICES

IoT devices are capable of sensing, communicating, and processing data with built-in sensors, through which things can be connected to the Internet. It generates and distributes the data to the gateway via a wireless communication channel.

### C. GATEWAY

The gateway device serves as a relay node between the cloud server and IoT devices. It simultaneously receives data from IoT devices and outsources the sensed data into the cloud.

### D. TA

The TA is a trusted third party that is responsible for protecting stored data from malicious users and authenticating cloud users. The TA also provides authentication to authorized users.

### E. CLOUD SERVER

Our work uses a hybrid cloud, which is a cloud computing environment that consists of a mix of a public cloud and a private cloud, to perform distinct functions within enterprises. Such a hybrid cloud organization can improve efficiencies by employing public cloud services for nonsensitive information. A private cloud environment provides not only security but also applicable requirements for data handling and storage.

### F. DATA ENCRYPTION

In the proposed architecture, the IoT devices include sensitive and nonsensitive devices. The encryption schemes used vary with the device type.

i. Sensitive Data: Encryption using RC6 and Fiestel
ii. Nonsensitive Data: Encryption using AES

### G. SENSITIVE DATA ENCRYPTION (RC6 AND FIESTEL)

Sensitive device data are divided into two parts to protect them from malicious users. One part of the data is encrypted using the RC6 encryption algorithm, and the other part is encrypted using the Fiestel encryption algorithm. Even if an unauthorized user manages to obtain an encryption key, this user would still be unable to fully read the data. Let us consider a sensitive device $S_i$ data that are split into $S_{ia}$ and $S_{ib}$.

#### 1) RC6 ENCRYPTION

$S_{ia}$ is encrypted using the RC6 encryption algorithm. RC6 is a symmetric key block cipher derived from RC5. It contains a normal block size of 128 bits and supports key sizes of 128, 192, and 256 bits up to 2040 bits. RC6 encrypts data at a high speed with low code memory. The RC6 algorithm has six basic operations to encrypt and decrypt a given text:

- Integer addition $(a + b)$,
- Integer subtraction $(a - b)$,
- Bitwise exclusive OR $(a \wedge b)$,
- Integer multiplication $(a * b)$,
- Rotate left $(a \ll b)$,
- Rotate right $(a \gg b)$.

The RC6 encryption is shown in Figure 4(a). $S_{ia}$ is stored in four w-bit RC6 registers, that is, A, B, C, and D. The process of RC6 encryption and decryption consists of three stages, namely, prewhitening, postwhitening, and inner loop rounds. The prewhitening and postwhitening steps remove the possibility of the plaintext revealing part of the input in the first round of encryption and the ciphertext revealing part of the input in the last round of encryption.

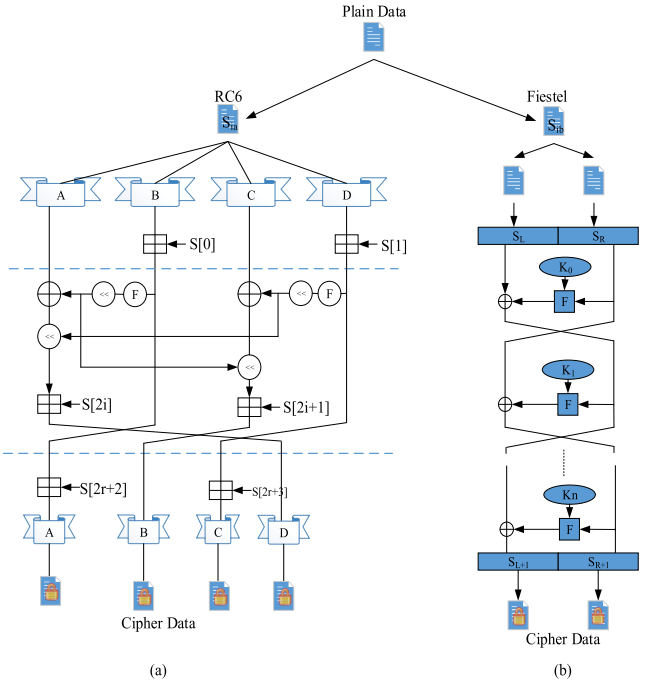


**FIGURE 4.** (a) RC6 encryption 2, (b) Fiestel encryption system.

**TABLE 1.** Pseudocode for sensitive data encryption.

| **Pseudo Code for Sensitive Data Encryption** |
|---|
| **RC6 Encryption** |
| **Input :** Raw data $S_{ia}$ stored in four w-bit input registers A, B, C, D |
| **Number of Rounds:** K |
| W-bit round keys S [0, …., 2R+3] |
| **Output:** Ciphertext of given data stored in A, B, C, D |
| **Procedure** |
| //Prewhitening |
| B = B + S[0] |
| D = D + S[1] |
| //Inner Loop Rounds |
| for i = 1 to K do |
| { |
| t = (B x (2B + 1)) <<< $\log_2$ w |
| u = (D x (2D + 1)) <<< $\log_2$ w |
| A = ((A t) <<< u) + S[2i] |
| C = ((C u) <<< t) + S[2i+ 1] |
| (A,B,C,D) = (B,C,D,A) |
| } |
| //Postwhitening |
| A = A + S[2r + 2] |
| C = C + S[2r + 3] |
| **Fiestel Encryption** |
| **Input:** Raw data $S_{ib}$ given as input to the Fiestel |
| **Round Function :** 𝔽 and "r" rounds |
| **Round Keys:** $K_0$ .. $K_n$ |
| **Output:** Ciphertext of given input |
| **Procedure** |
| //Data Splitting |
| $S_{ib} = S_{ib_L} + S_{ib_R}$ |
| Compute $S_{ib_L}(r+1)$ and $S_{ib_R}(r+1)$ functions → equation 1 and 2 |
| Ciphertext → $S_{ib_{R+1}}$ & $S_{ib_{L+1}}$ |

The pseudocode in Table 1 shows the RC6 and Fiestel encryption algorithms used to encrypt sensitive data. log2 w represents the rotation of the w-bit data. Variables t and u are

used to store the computed values. Initially, registers B and D undergo prewhitening. Thereafter, the inner loop rounds are executed, that is, the four registers undergo the rotate left, rotate right, and addition operations. Using these processes, RC6 converts the given text into ciphertext. This encryption process is reversed in the decryption process.

#### 2) FIESTEL ENCRYPTION

The Fiestel algorithm is used to encrypt the other part of the divided data, i.e., $S_{ib}$. In cryptography, the Fiestel cipher is a symmetric technique used to construct block ciphers. In this encryption scheme, the encryption and decryption operations are similar and are thus easy to implement. The proposed Fiestel algorithm splits the data into two parts to improve the security of the encrypted data. The pseudocode for the Fiestel encryption scheme is illustrated in Table 1. The Fiestel encryption scheme encrypts the data at a rapid rate, which is important in encryption operations. The encryption process of the Fiestel algorithm involves multiple rounds of handling raw data. Each round has a substitution process monitored by the permutation process. Virtually all rounds in the Fiestel encryption process are similar.

Let $\mathbb{F}$ be the round function of the Fiestel cipher, and let $K_0, K_1, \ldots K_n$ be the subkeys for rounds 0,1,..n, respectively. Data $S_{ib}$ are split into two equal pieces $S_{ib_L}$ and $S_{ib_R}$. For each round r = 0, 1….n, we compute

$$S_{ib_L}(r+1) = S_{ib_R}(r), \tag{1}$$

$$S_{ib_R}(r+1) = S_{ibL}(r) \oplus \mathbb{F}(S_{ib_R}, K_i), \tag{2}$$

where $\oplus$ represents the XOR operator and $K_i$ represents the key value. Then, the ciphertext is attained as $S_{ib_{R+1}}$ and $S_{ib_{L+1}}$. One of the advantages of the Fiestel encryption scheme is that the round function does not need to be invertible.

Figure 4(b) illustrates the Fiestel encryption and decryption operations. The major advantage can be seen in the diagram, that is, the given raw data are split and then subjected to the encryption process. This method of encryption and decryption increases the efficiency of both operations. The decryption process is the reverse of the encryption process, that is, key values are provided from the last value to the first value.

By using the two lightweight cryptography algorithms, namely, RC6 and Fiestel, we provide security to the IoT data. The encrypted data are stored in the hybrid cloud via the gateway device, which acts as a communication link between the IoT device and the hybrid cloud.

### H. NONSENSITIVE DATA ENCRYPTION (AES)

Non-sensitive data are produced by non-sensitive IoT devices. Hence, we use the AES algorithm to encrypt non-sensitive data.

The proposed AES algorithm is a variant of the Rijndael algorithm, which has a fixed block size of 128 bits and key sizes of 128, 192, and 256 bits. AES basically repeats four major operations to encrypt data. It then takes a 128-bit block of data and key and generates a ciphertext as the output. The four operations are

- ➢ Byte substitution
- ➢ Shift rows
- ➢ Mix Columns
- ➢ Add round key

The AES algorithm is based on the substitution and permutation network.

### I. BYTE SUBSTITUTION

Byte substitution is the first process in the AES-based encryption scheme. Herein, 16 bytes of input data are substituted by looking up a fixed table (S-box). The result of this process is provided in a matrix of four rows and four columns.

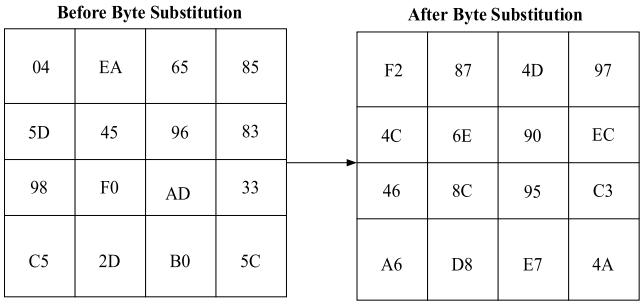


**FIGURE 5.** Example of byte substitution.

Figure 5 depicts the byte substitution example of 4 × 4 matrixes, where the bytes on the left are converted to bytes on the right after substitution.

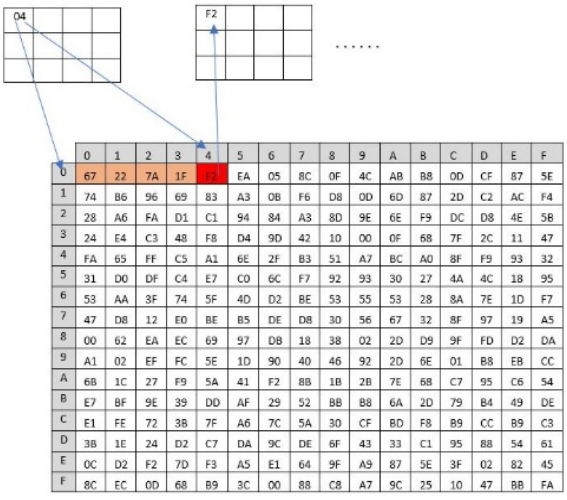


**FIGURE 6.** Byte substitution process.

As shown in Figure 6, Given a 2-byte input, the 2-byte output is found by selecting the row using the outer byte (0), and the column using the inner byte (4). For example, an input "04" has outer byte "0" and inner byte "4"; the corresponding output would be "F2".

### J. SHIFT ROWS

Shift row is the second process in the AES encryption scheme. Here, each of the four rows of the matrix is shifted to the left. Any entry that falls off are reinserted on the right side of the row. Shifts are carried out using the following steps:

- The first row is not shifted.
- The second row is shifted one (byte) position to the left.
- The third row is shifted two positions to the left.
- The fourth row is shifted three positions to the left.
- The resultant is a new matrix consisting of the same 16 bytes but shifted with respect to one another.

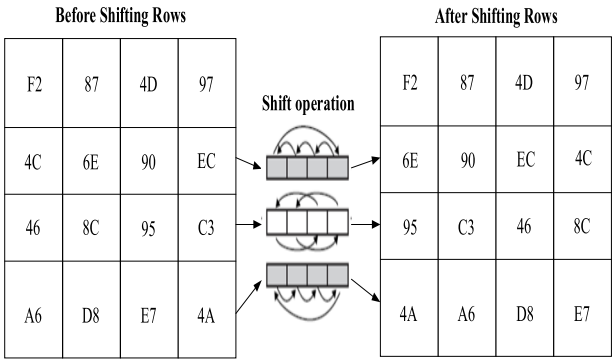


**FIGURE 7.** Example of shift rows.

Figure 7 depicts the shifting rows of a $4 \times 4$ matrix, in which the first row is not shifted and the other three rows are shifted to the left. The shift row operation permutes bytes between columns by shifting the column value of each byte to the left side. This shifting operation is performed throughout the matrix, except for the first row.

### K. MIX COLUMNS

The mix column is the third process in the AES encryption scheme. Each column is processed separately. Moreover, each column of four bytes is converted using a special mathematical function. This function takes a four-byte input of one column and produces a four-byte output. The result is another new matrix that consists of 16 new bytes. This process is not performed in the final round.

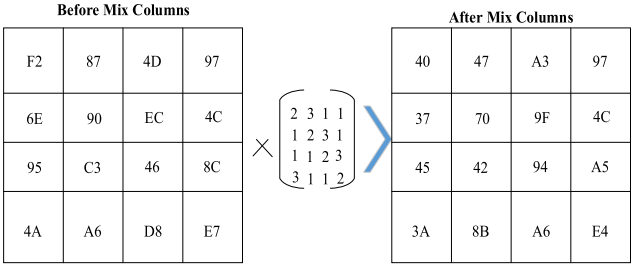


**FIGURE 8.** Example of mix column.

Figure 8 demonstrates the mix column operation, in which the output from the previous process is a matrix multiplied by the matrix depicted in Figure 7. Upon multiplying this matrix with the previous output values, we obtain the mix column output. This output is given to the next process to obtain the cipher data of the given plain data.

### L. ADD ROUND KEY

The add round key process is the final process in the AES encryption scheme. The result from the previous step is considered as a 128-bit input, and the operation involves XORing the 128-bit data with a 128-bit key. The result from this round is described as the cipher data of the given plain data.

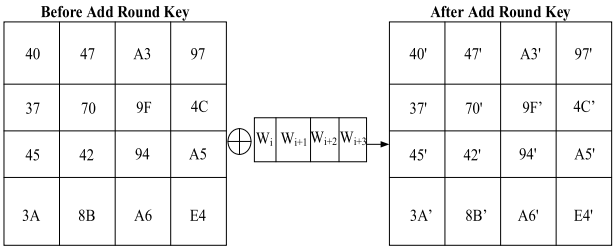


**FIGURE 9.** Example of add round key.

Figure 9 demonstrates the example for the add round key, in which the output from the mix column operation is converted to cipher data. The output from this process is the cipher data of the given input. By utilizing the AES encryption method, we encrypt our nonsensitive data effectually. The decryption process is similar to the encryption process, that is, the four operations described previously are performed in reverse order.

### M. USER REGISTRATION

In the user registration phase, users register their credentials with the TA so that they can be authenticated to access data in the cloud. In our work, users need to provide their user ID, password, and biometrics, such as retina and fingerprint, to the TA. After receiving credentials from users, the TA provides a pseudo ID for each user. After a successful registration, each user can request the TA for authentication to access data in the cloud. A user needs to provide their credentials in each level of data to access the stored data securely. The TA authentication process is explained in the subsequent sections.

### N. MULTILEVEL AUTHENTICATION

Access to data in the cloud is provided by the TA through a multilevel authentication scheme. The level of authentication required is based on the type of file access that the user wants to perform, as explained below.

I. First-level authentication - read files (public cloud)
II. Second-level authentication - download files (public cloud)
III. Third-level authentication - read or download files (private cloud)

Figure 10 illustrates the multilevel authentication operation.

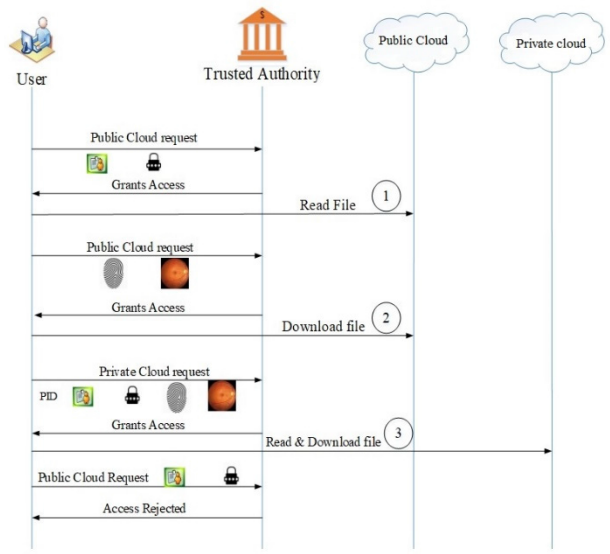


**FIGURE 10.** Multilevel authentication.

### O. FIRST-LEVEL AUTHENTICATION

The first-level authentication is performed to read the files that are stored in the public cloud. Figure 10 shows the multilevel authentication. In this figure, symbol 1 read file represents the first level of authentication. In this level, the user needs to send a read request together with their user ID and password to the TA. The TA verifies the user-provided credentials against the registered credentials. If they match, then the access to read the file in the public cloud is granted, and the key used to decrypt the data is given to the user.

### P. SECOND-LEVEL AUTHENTICATION

The second level of authentication is required to download files from the public cloud. In Figure 10, the second level of authentication is represented by symbol 2. A user sends a request to download a file from the public cloud together with their biometrics, such as fingerprint or retina, to the TA. After obtaining the credentials from the user, the TA verifies the received credentials against the registered credentials. If the registered and received credentials are similar, then the TA allows the requested file from the public cloud to be downloaded and sends the key required to decrypt the file.

### Q. THIRD-LEVEL AUTHENTICATION

Third-level authentication is the final step in the multilevel authentication approach. After a successful completion of the first and second levels of authentication, the user can enter the third level of authentication. Herein, the user needs to send a private cloud request along with their credentials, such as user ID, password, and biometrics (e.g., retina and fingerprint, pseudo ID). The TA receives the user ID, password, pseudo ID, and biometrics from the user. If these credentials match the registered credentials, then the TA provides access to read and download files from the private cloud; otherwise, it rejects the request.

## VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed architecture and compare it with existing methods, namely, FCS, MCP-ABE, and CP-ABE. The performance metrics used are computational time, security strength, encryption time with respect to key size and message size, and decryption time with respect to key size and message size. This section is divided into three subsections, namely, experimental setup, performance metrics, and comparative analysis. The experimental setup section explains the simulation environment and simulation parameters.

### A. EXPERIMENTAL SETUP

In this section, we describe our experimental setup. We implement our proposed architecture in the Network Simulator 3.26 (NS3.26) tool installed in an Ubuntu operating system. The NS3.26 simulator is a discrete event simulator that supports the simulation of different types of networks. The NS3 simulator is developed to provide an open, extensible network simulation platform for networking research and education. Hence, we use this tool in our cloud–IoT environment. Table 2 presents the important simulation parameters.

**TABLE 2.** Simulation parameters.

| Parameters | Value |
|---|---|
| Simulation area | 1000 m×1000 m |
| Number of users | 50 |
| Number of IoT devices | 50 |
| Number of sensitive devices | 25 |
| Number of nonsensitive devices | 25 |
| Communication range of IoT devices | 100 m |
| Number of cloud servers | 2 |
| Number of trusted authorities | 1 |
| Number of gateways | 2 |
| Block size | 128 bits |
| Maximum key size | 1024 bits |
| Number of rounds | 20 |
| Maximum message size | 500 MB |
| Simulation time | 100 s |

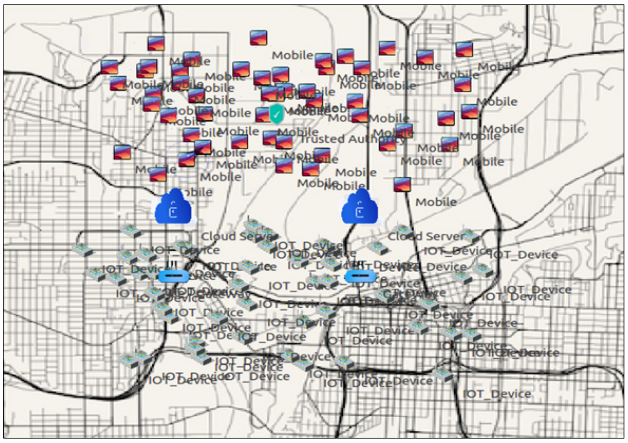


**FIGURE 11.** Simulation environment of cloud–IoT.

Figure 11 shows the simulated cloud–IoT environment with IoT devices, data users, gateway devices, cloud servers,

and the TA. In this simulation, the IoT devices are sensing their surroundings in the range of 100 m. The sensor data produced by the IoT devices are then encrypted according to the specified scheme. The encrypted data are sent and stored in cloud servers via the gateway devices. The TA provides security to the stored data by providing access control through the multilevel authentication scheme.

### B. PERFORMANCE METRICS

This section describes the four performance metrics used in performance evaluation.

### C. COMPUTATIONAL TIME

Computational time is defined as the time taken to complete a user request until the data retrieval process is completed successfully. Computational time is also referred to as the running time. The computational time metric is expressed as follows:

$$Computational\ time = \frac{Time\ required\ for\ i^{th}\ user}{Total\ Number\ of\ user}. \quad (3)$$

### D. SECURITY STRENGTH

The security strength metric is an important metric to analyze the proposed multilevel authentication and lightweight encryption methods. Herein, we propose three lightweight cryptography mechanisms to provide high security to sensed data and allow multilevel authentication to protect data against malicious users. The security strength metric measured in our work using key size.

### E. ENCRYPTION TIME

Encryption time can be described as the time required to convert plaintext into ciphertext. It can be computed using the following expression:

$$Encryption\ time = \frac{Time\ required\ to\ encrypt\ i^{th}\ data}{Total\ number\ of\ data}. \quad (4)$$

### F. DECRYPTION TIME

Decryption time can be defined as the time required to convert a ciphertext into plain text. It can be computed using the following expression:

$$Decryption\ time = \frac{Time\ required\ to\ decrypt\ i^{th}\ data}{Total\ number\ of\ data}, \quad (5)$$

### G. COMPARATIVE ANALYSIS

In this section, we compare our proposed architecture with CP-ABE, MCP-ABE, and FCS in terms of encryption time, decryption time, computation time, and security strength.

### H. IMPACT ON COMPUTATIONAL TIME

Computational time measures how much time is required to complete the process of one user. A high computational time indicates an inefficient method. We measure computational time with respect to the number of users.

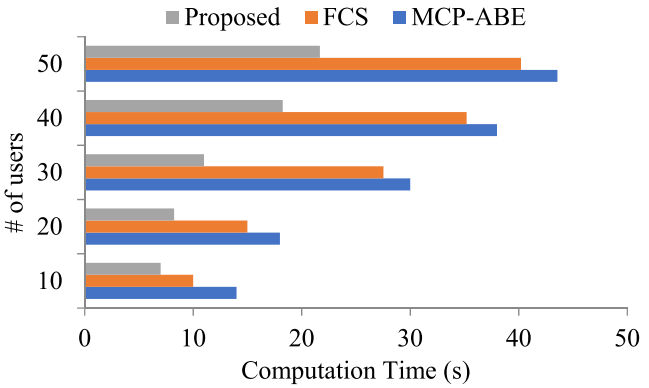


**FIGURE 12.** Comparison according to computational time.

Figure 12 depicts the comparison of computational times. The proposed scheme has a lower computational time than existing methods, such as FCS and MCP-ABE. We use lightweight cryptography encryption schemes with a low encryption and decryption time. We run all three encryption methods in parallel, hence the low computational time. Our proposed scheme achieves a maximum computational time of 22 s for 50 users. The MCP-ABE method achieves a maximum computational time of 43 s for 50 users. This method is slower than ours because it uses the ABF-based attribute selection policy and thus takes long to find the correct attribute of the requested user. In addition, ABF is greatly prone to false positives, which indicate that attributes are members of an access policy group when they really are not. FCS takes a maximum of 40 s to complete the process for 50 users. As it maintains the signatures of users in a table, its computational time is high. From these comparisons, we conclude that our proposed scheme achieves a lower computational time than the existing methods of FCS and MCP-ABE.

### I. IMPACT ON SECURITY STRENGTH

The security strength metric is evaluated to measure the security level of the proposed cloud-assisted IoT environment. This metric evaluates the security of the stored data in the cloud-assisted IoT. We measure security strength using the key sizes of the proposed encryption schemes.

Figure 13 depicts the comparison of security strengths. The proposed method achieves better security strength than the existing methods of MCP-ABE and FCS. This superiority of the proposed method can be attributed to the proposed highly secure lightweight cryptography-based encryption for sensed data of IoT devices. We use two algorithms, namely, RC6 and Fiestel, for sensitive data encryption. The proposed use of RC6 and Fiestel results in better security than other encryption algorithms. As the proposed method uses symmetric keys for encryption, it consumes less time in comparison with other asymmetric encryption schemes. Hence, the proposed environment achieves a maximum security strength of 92% for a key size of 1024 bits.

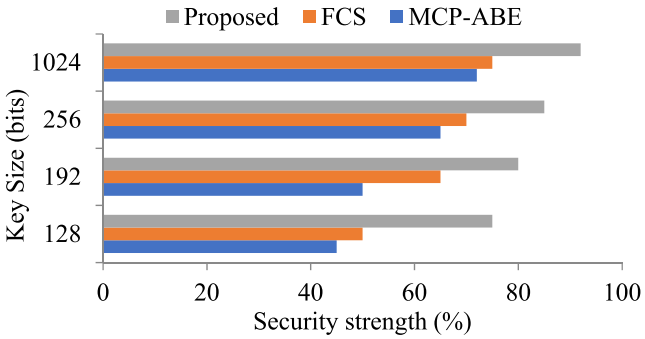


**FIGURE 13.** Comparison according to security strength.

The MCP-ABE and FCS schemes attain minimal security strength due to their poor encryption schemes. The FCS method achieves a low security strength for a key size of 1024 bits key size because users only need to select such key for encryption. This condition leads to reduced data security. The MCP-ABE scheme achieves low security strength because it encrypts data using bilinear pairing keys with a low security of 75%. From these comparisons, we conclude that our proposed method achieves high security that exceeds that of existing methods by more than 17%.

### J. IMPACT ON ENCRYPTION TIME

The encryption time metric is evaluated to determine the proposed method's efficiency in terms of time. Encryption time refers to the time taken to complete the encryption of plain data. Our work measures the encryption time with respect to key size and message size.

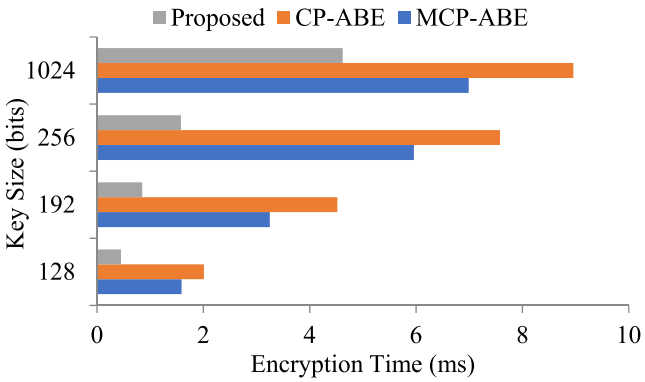


**FIGURE 14.** Comparison according to encryption time vs. key size.

Figure 14 demonstrates the comparison of encryption times with respect to key size. From the comparison results, we conclude that our proposed encryption scheme has a lower encryption time than CP-ABE and MCP-ABE. This difference is due to our parallel use of the RC6, Fiestel, and AES encryption methods. Moreover, the proposed method uses a highly secure lightweight encryption scheme that computes keys at a rapid rate and encrypts plain data effectually with low computation time. The proposed method achieves a maximum encryption time of 4 ms for a key size of 1024 bits, whereas CP-ABE and MCP-ABE respectively require 8.96 and 6.99 ms for the encryption of plain data using the same key size. From the comparison, we conclude that our proposed method is faster than MCP-ABE and CP-ABE in terms of encryption time.

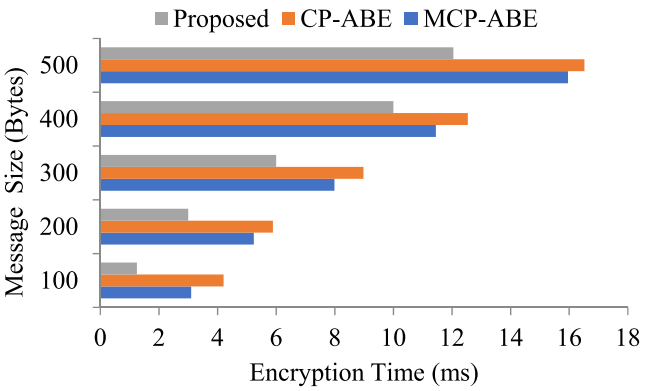


**FIGURE 15.** Comparison according to encryption time vs. message size.

Figure 15 illustrates that the proposed method requires less encryption time than MCP-ABE and CP-ABE. Here, the encryption time is measured with respect to message size (bytes).

The proposed method achieves an encryption time of 12 ms for a message size of 500 MB. By contrast, CP-ABE and MCP-ABE respectively achieve maximum encryption times of 16.52 and 15.96 ms for the same message size. The CP-ABE scheme has a high encryption time due to its poor encryption mechanism. It uses bilinear pair mapping-based key generation, which is time consuming. The proposed TSSS method divides the secret key into "n" shares, which are large as the secret key itself. Thus, the secret key is compressed before sharing. However, the same is not possible for high-quality data and is time consuming. MCP-ABE also requires a high encryption time because it involves a large number of attributes for verification. From the comparison results, we conclude that the proposed scheme achieves the lowest encryption time among all methods compared herein.

### K. IMPACT ON DECRYPTION TIME

Decryption time is evaluated to measure the decryption time of the proposed method. We measure decryption time with respect to key size and message size.

Figure 16 illustrates that our proposed decryption mechanism achieves better results than CP-ABE and MCP-ABE. This result is due to our proposed cryptography encryption-based security schemes that perform decryption rapidly. Herein, the cryptography encryption scheme is a symmetric key-based encryption in which the encryption and decryption functions are similar and thus reduce the decryption time for converting cipher data into normal data. Hence, we achieve a maximum decryption time of 14 ms for 500 MB of data. By contrast, CP-ABE and MCP-ABE respectively achieve maximum decryption times of 17.34 and 16.32 ms for the same data size due to their poor decryption mechanism.

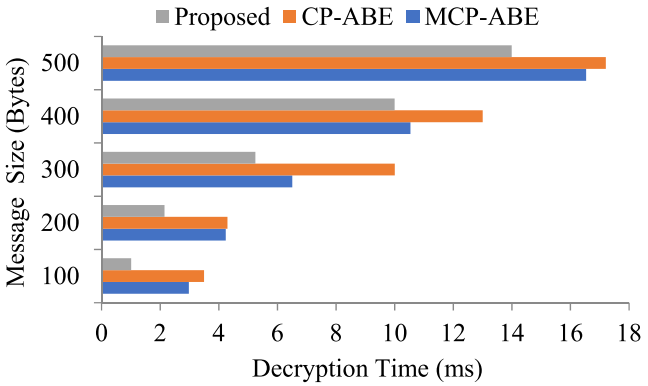


**FIGURE 16.** Comparison according to decryption time vs. message size.

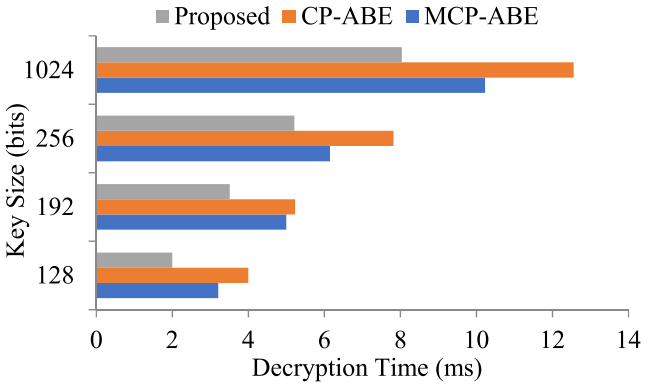


**FIGURE 17.** Comparison according to decryption time vs. key size.

Figure 17 indicates that our proposed method achieves a lower decryption time than CP-ABE and MCP-ABE. The CP-ABE and MCP-ABE schemes respectively achieve maximum decryption times of 12.52 and 10.01 ms. The CP-ABE method involves a large number of attributes to verify authorized users to provide security to stored data. Such high volume data are encrypted via bilinear pairing, which involves numerous complex multiplication operations that lead to a high computational time in data decryption. Similarly, MCP-ABE involves multi-attribute data and performs encryption/decryption operations through a weak algorithm that tends to increase the decryption time. From the comparison, we conclude that our proposed method achieves better performance than the FCS, CP-ABE, and MCP-ABE methods.

## VII. CONCLUSION

In recent years, cloud-integrated IoT applications have become popular among researchers due to their vital applications in organizations, private sectors, domestic appliances, etc. This work proposes a secure cloud–IoT environment using multifactor authentication and lightweight cryptography schemes. The proposed method splits IoT devices into sensitive and nonsensitive devices. We propose the use of a hybrid cloud that contains public cloud and private cloud. Sensitive device data are divided into two and encrypted using the RC6 and Fiestel encryption algorithms. These data are stored in a private cloud to provide high security via a gateway device. By contrast, nonsensitive device data are encrypted using AES and stored in a public cloud via a gateway device. Multifactor authentication is provided by the TA. In this process, the user undergoes three levels of authentication by providing their credentials, such as user ID, password, and biometrics (e.g., retina and fingerprint). We evaluate the performance of the proposed method using metrics that include computational time, security strength, encryption time, and decryption time. From the comparison results, we prove that the proposed method performs better than FCS, CP-ABE, and MCP-ABE.

In the future, we intend to propose mutual authentication between gateway devices and IoT devices. In addition, we aim to propose DDoS attack detection in cloud servers.

## REFERENCES

[1] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 43, pp. 619–636, Oct. 2018.
[2] I. Al Ridhawi, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "A profitable and energy-efficient cooperative fog solution for IoT services," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3578–3586, May 2020.
[3] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.
[4] G. Sharma and S. Kalra, "A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications," *J. Inf. Secur. Appl.*, vol. 42, pp. 95–106, Oct. 2018.
[5] S. Raza, T. Helgason, P. Papadimitratos, and T. Voigt, "SecureSense: End-to-end secure communication architecture for the cloud-connected Internet of Things," *Future Gener. Comput. Syst.*, vol. 77, pp. 40–51, Dec. 2017.
[6] B. Jin, J. Park, and H. Mun, "A design of secure communication protocol using RLWE-based homomorphic encryption in IoT convergence cloud environment," *Wireless Pers. Commun.*, vol. 105, no. 2, pp. 599–618, 2019.
[7] H. C. Chen, "Collaboration IoT-based RBAC with trust evaluation algorithm model for massive IoT integrated application," *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 839–852, 2019.
[8] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Gener. Comput. Syst.*, vol. 91, pp. 244–251, Feb. 2019.
[9] G. Sharma and S. Kalra, "Advanced lightweight multi-factor remote user authentication scheme for cloud-IoT applications," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 1771–1794, Feb. 2019.
[10] J. Guo, I.-R. Chen, D.-C. Wang, J. P. Tsai, and H. Al-Hamadi, "Trust-based IoT cloud participatory sensing of air quality," *Wireless Pers. Commun.*, vol. 105, pp. 1461–1474, Feb. 2019.
[11] X. Li, X. Jin, Q. Wang, M. Cao, and X. Chen, "SCCAF: A secure and compliant continuous assessment framework in cloud-based IoT context," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–18, Oct. 2018.
[12] S. P. Gochhayat, P. Kaliyar, M. Conti, P. Tiwari, V. B. S. Prasath, D. Gupta, and A. Khanna, "LISA: Lightweight context-aware IoT service architecture," *J. Cleaner Prod.*, vol. 212, pp. 1345–1356, Mar. 2019.
[13] P. T. M. Ly, W.-H. Lai, C.-W. Hsu, and F.-Y. Shih, "Fuzzy AHP analysis of Internet of Things (IoT) in enterprises," *Technol. Forecasting Social Change*, vol. 136, pp. 1–14, Nov. 2018.
[14] S. Pérez, D. Garcia-Carrillo, R. Marín-López, J. L Hernández-Ramos, R. Marín-Pérez, and A. F. Skarmetaa, "Architecture of security association establishment based on bootstrapping technologies for enabling secure IoT infrastructures," *Future Gener. Comput. Syst.*, vol. 95, pp. 270–285, Jun. 2019.
[15] M. Kazim, L. Liu, and S. Y. Zhu, "A framework for orchestrating secure and dynamic access of IoT services in multi-cloud environments," *IEEE Access*, vol. 6, pp. 58619–58633, 2018.
[16] F. Al-Turjman, Y. K. Ever, E. Ever, H. X. Nguyen, and D. B. David, "Seamless key agreement framework for mobile-sink in IoT based cloud-centric secured public safety sensor networks," *IEEE Access*, vol. 5, pp. 24617–24631, 2017.

[17] Q. Huang, L. Wang, and Y. Yang, "DECENT: Secure and fine-grained data access control with policy updating for constrained IoT devices," *World Wide Web*, vol. 21, no. 1, pp. 151–167, Jan. 2018.
[18] E. A. Alkeem, D. Shehada, C. Y. Yeun, M. J. Zemerly, and J. Hu, "New secure healthcare system using cloud of things," *Cluster Comput.*, vol. 20, no. 3, pp. 2211–2229, Sep. 2017.
[19] P. Xu, X. Tang, W. Wang, H. Jin, and L. T. Yang, "Fast and parallel keyword search over public-key ciphertexts for cloud-assisted IoT," *IEEE Access*, vol. 5, pp. 24775–24784, 2017.
[20] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Secure data sharing and searching at the edge of cloud-assisted Internet of Things," *IEEE Cloud Comput.*, vol. 4, no. 1, pp. 34–42, Jan. 2017.
[21] A. M. Elmisery, S. Rho, and M. Aborizka, "A new computing environment for collective privacy protection from constrained healthcare devices to Iot cloud services," *Cluster Comput.*, vol. 22, no.1, pp. 1611–1638, 2019.
[22] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes," *Future Gener. Comput. Syst.*, vol. 78, pp. 1040–1051, Jan. 2018.
[23] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, and X. Shen, "Fine-grained data access control with attribute-hiding policy for cloud-based IoT," *Comput. Netw.*, vol. 153, pp. 1–10, Apr. 2019.
[24] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT," *Comput. Netw.*, vol. 133, pp. 141–156, Mar. 2018.
[25] Y.-N. Liu, Y.-P. Wang, X.-F. Wang, Z. Xia, and J.-F. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Comput. Netw.*, vol. 148, pp. 340–348, Jan. 2019.
[26] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported Internet of Things in smart grid," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1934–1944, Dec. 2017.
[27] G. Ramu, "A secure cloud framework to share EHRs using modified CP-ABE and the attribute Bloom filter," *Edu. Inf. Technol.*, vol. 23, no. 5, pp. 2213–2233, Sep. 2018.
[28] C.-Y. Yang, C.-T. Huang, Y.-P. Wang, Y.-W. Chen, and S.-J. Wang, "File changes with security proof stored in cloud service systems," *Pers. Ubiquitous Comput.*, vol. 22, no. 1, pp. 45–53, Feb. 2018.
[29] P. K. Dhillon and S. Kalra, "Multi-factor user authentication scheme for IoT-based healthcare services," *J. Reliable Intell. Environ.*, vol. 4, no. 3, pp. 141–160, Sep. 2018.
[30] Y. Jararweh, L. Tawalbeh, H. Tawalbeh, and A. Moh'd, "Hardware performance evaluation of SHA-3 candidate algorithms," *J. Inf. Secur.*, vol. 3, no. 2, pp. 69–76, 2012.
[31] Y. Kotb, I. Al Ridhawi, M. Aloqaily, T. Baker, Y. Jararweh, and H. Tawfik, "Cloud-based multi-agent cooperation for IoT devices using workflow-nets," *J. Grid Comput.*, vol. 17, no. 4, pp. 625–650, Dec. 2019.
[32] V. Balasubramanian, F. Zaman, M. Aloqaily, I. A. Ridhawi, Y. Jararweh, and H. B. Salameh, "A mobility management architecture for seamless delivery of 5G-IoT services," in *Proc. ICC-IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.
[33] I. A. Ridhawi, S. Otoum, M. Aloqaily, Y. Jararweh, and T. Baker, "Providing secure and reliable communication for next generation networks in smart cities," *Sustain. Cities Soc.*, vol. 56, May 2020, Art. no. 102080.

**SALEH ATIEWI** received the B.Sc. degree in computer science from Al-Isra University, Amman, Jordan, in 1999, the master's degree in Internet technology from Wollongong University, Wollongong, Australia, in 2004, and the Ph.D. degree in computer science from Tenaga Nasional University, Putrajaya, Malaysia, in 2017. Since 2004, he has been with Al Hussein Bin Talal University, Maan, Jordan, where he is currently an Assistant Professor with the Department of Computer Science. He is currently the Head of Computer Science Department. His research interests include network security, cloud computing, and the Internet of Things.

**AMER AL-RAHAYFEH** (Member, IEEE) received the B.S. degree in computer science from Mutah University in 2000, the M.S. degree in computer information systems from The Arab Academy for Banking and Financial Sciences in 2004, and the Ph.D. degree in computer science and engineering from the University of Bridgeport, USA, in 2014. He is an Assistant Professor with the College of Information Technology, Al-Hussein Bin Talal University, Ma'an, Jordan. He served as the Chair of Computer Science Department from 2015 to 2018. His research interests include image processing, computer security, sensor networks, and the Internet of Things.

**MUDER ALMIANI** (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Bridgeport, USA, in 2013. He is an Associate Professor with the College of Information Technology, in 2014. He was the Chair of the Computer Information Systems Department, College of Information Technology. He is a member of the Information Technology, College Council, AHU, the Business Administration and Economics College Council, AHU, the University Senate Executive Committee, AHU, from 2007 to 2008, the University Planning Council, AHU, from 2007 to 2008, and the IEEE Computer Society since 2011. He has been the Chair of the Computer/Computational Intelligence Chapter, IEEE Jordan Section, since 2015. His research interests include security, sensor networks, cloud computing, the Internet of Things, and image processing.

**SALMAN YUSSOF** received the B.S. and M.S. degrees in electrical and computer engineering from the University of Carnegie Mellon, USA, in 1999, and the Ph.D. degree in engineering from University Tenaga National, Malaysia, in 2010. From 1998 to 1999, he was a Research Programmer with the Institute for Complex Engineered System, Carnegie Mellon University (CMU), USA. Since 1999, he has been a Lecturer with the System and Networking Department, UNITEN University. He is currently an Associate Professor with UNITEN University. He is the author of more than 72 publications. His research interests include computer networks, network security, distributed system, image processing, robotic, and evolutionary computing. In recent years, he has been involved with many professional bodies, such as the Association for Computing Machinery (ACM) since 2003, the Malaysian Invention and Design Society (MINDS) since 2004, the Boards of Engineers Malaysia (BEM) since 2005, the Malaysian National Computer Confederation (MNCC) since 2008, and the Internet Society (ISOC) since 2011.

**OMAR ALFANDI** (Member, IEEE) received the M.Sc. degree in telecommunication engineering from the University of Technology Kaiserslautern, Germany, in 2005, and the Dr. rer.nat. degree in computer science and telematics from the Georg-August-University of Goettingen, Germany, in 2009. He is an Assistant Dean for Students Affairs (AUH) and an Associate Professor with the College of Technological Innovation, Zayed University. From 2009 to 2011, he enjoyed a Postdoctoral Fellowship with the Telematics Research Group and he founded the Research and Education Sensor Laboratory, where he is currently a Lab Advisor. Before that, he carried his doctoral research as part of an industry, academia and research centers collaboration European Union (EU) project. He was a Working Package Leader of EU DAIDALOS II in the 6th framework project. He published numerous articles on authentication framework for 4G communication Systems, future Internet and trust and reputation systems in mobile ad hoc and sensor networks. He is the Co-Founder and the Co-Director of the Sensors and Mobile Applications Research and Education (SMART) Laboratory, CTI. In August 2015, he was appointed as the Assistant Dean for Abu Dhabi Campus. His current research interests include the Internet of Things (IoT), security in next generation networks, smart technologies, security engineering, and mobile and wireless communications.

**AHED ABUGABAH** (Member, IEEE) received the degree in information systems in Australia. He worked in the airlines industry in the aircraft engineering and supply chain management. Before joining Zayed University in 2016, he was involved in administration as the Associate Dean and a University Council Member at American University, United Arab Emirates. He is an Assistant Professor with the College of Technological Innovation, Zayed University. His research interests include information systems, enterprise applications and development, healthcare information systems, and RFID in healthcare.

**YASER JARARWEH** received the Ph.D. degree in computer engineering from the University of Arizona in 2010. He is currently an Associate Professor of computer science with the Jordan University of Science and Technology, Jordan. He has coauthored about 70 technical papers in established journals and conferences in fields related to cloud computing, HPC, SDN, and big data. He is also chairing many IEEE events, such as ICICS, SNAMS, BDSN, IoTSMS, and many others. He served as a guest editor for many special issues in different established journals. Also, he is the Steering Committee Chair of the IBM Cloud Academy Conference. He is an Associate Editor in *Cluster Computing* (Springer) journal, *Information Processing and Management* (Elsevier), and others.