Tikrit Journal of Engineering Sciences (2025); 32(4): 1823
DOI: http://doi.org/10.25130/tjes.32.4.5

IRAQI Academic Scientific Journals

ISSN: 1813-162X (Print); 2312-7589 (Online)
Tikrit Journal of Engineering Sciences
available online at: http://www.tj-es.com

TJES Tikrit Journal of Engineering Sciences

Implementing a New Lightweight Data Encryption Algorithm for Internet of Things

Hussein A. Mustafa *[a,b], Galip Cansever [b]

[a] Electrical and Computer Engineering Department, Engineering College, Altinbaş University, Istanbul, Türkiye.
[b] Electrical and Electronics Engineering Department, Engineering College, Altinbaş University, Istanbul, Türkiye.

Keywords:
Data Integrity; Internet of Things; Lightweight Encryption algorithm; Mosquitto Broker; Message Queue Telemetry Transport Protocol (MQTT); Raspberry Pi.

Highlights:

- LSID-IoT is a new lightweight encryption algorithm using Raspberry Pi 4 & MQTT Protocol.
- IMPLICIT KEY.
- Provide DATA INTEGRITY.

ARTICLE INFO

Article history:

Received	25 Oct. 2023
Received in revised form	30 Aug. 2024
Accepted	29 Oct. 2024
Final Proofreading	23 Aug. 2025
Available online	29 Aug. 2025

Citation: Hussein A. Mustafa HA, Cansever G. **Implementing a New Lightweight Data Encryption Algorithm for Internet of Things**. *Tikrit Journal of Engineering Sciences* 2025; **32**(4): 1823.
http://doi.org/10.25130/tjes.32.4.5

*Corresponding author:
Hussein A. Mustafa
Electrical and Computer Engineering Department, Engineering College, Altinbaş University, Istanbul, Türkiye.

Abstract: Cyberspace is a complex environment consisting of heterogeneous technologies, i.e., fog computing, Internet of Things, cloud computing, and so forth, that result from interacting with services, software, and people on the Internet. It allows users to interact, share information, swap ideas, engage in social or discussion forums, play games, and conduct business, among many other activities. Cyberspace's biggest challenge is cyber-attacks, which affect security and integrity services. However, many traditional security mechanisms provide protection and security services to solve these issues. Therefore, many researchers have focused on solving security and integrity issues by addressing the need for effective lightweight encryption techniques that incorporate the advantages of lightweight symmetric and asymmetrical algorithms. In this paper, a lightweight encryption technique was created and applied with the following features: Keyless, Encryption & Integrity, Text & Number End-to-End Encryption, Reduce Traffic, and processing overhead. In addition, the proposed system provides data integrity by applying the HASH 256 function to generate a HASH value. The proposed lightweight encryption algorithm focuses on the optimal use of the resources of Internet of Things devices so that it dramatically saves all of (Processor, Memory, Energy, Time, and Bandwidth (no need to distribute the keys)), on the other hand, giving high security, especially against the crypto analyzer. In addition, the proposed lightweight encryption algorithm can manipulate text and numbers in the English and Arabic languages. Also, to achieve data integrity in the proposed system within the Internet of Things environment, 4 hexadecimal digits from the HASH value were used instead of the original 64 hexadecimal digit HASH value to reduce the network bandwidth, processing, and storage.

طريقة جديدة لتشفير بيانات أنترنت الأشياء

حسين علي مصطفى [١]، غالب كانسفر [٢]

[١] قسم هندسة الحاسوب والكهرباء / كلية هندسة / جامعة التن باش / اسطنبول – تركيا.

[٢] قسم هندسة الكهرباء والالكترونيات / كلية هندسة / جامعة التن باش / اسطنبول – تركيا.

الخلاصة

الفضاء الإلكتروني هو بيئة معقدة تتكون من تقنيات غير متجانسة مثل (الحوسبة السحابية، والحوسبة الضبابية، وإنترنت الأشياء، وما إلى ذلك) الناتجة عن تفاعل الخدمات والبرامج والأشخاص على الإنترنت. فهو يتيح للمستخدمين التفاعل ومشاركة المعلومات وتبادل الأفكار والمشاركة في المنتديات الاجتماعية أو منتديات المناقشة وممارسة الألعاب وإجراء الأعمال التجارية، من بين العديد من الأنشطة الأخرى. أكبر التحديات التي تواجه الفضاء الإلكتروني اليوم هي الهجمات السيبرانية، التي تؤثر على خدمات الأمن وتكامل البيانات. ومع ذلك، فإن العديد من آليات الأمن التقليدية توفر خدمات الحماية والأمن لحل هذه القضايا. لذلك، ركز العديد من الباحثين على حل مشكلات الأمان وسلامة تكامل البيانات من خلال زيادة الحاجة إلى تقنيات تشفير فعالة وخفيفة الوزن تتضمن مزايا الخوارزميات المتماثلة وغير المتماثلة خفيفة الوزن. في هذا البحث، سنقوم بتنفيذ وتصميم طريقة تشفير خفيفة الوزن تتميز بالخصائص التالية (مفتاح أقل، التشفير والتكامل، تشفير النص والأرقام من طرف إلى طرف، تقليل حركة المرور وتكاليف المعالجة). بالإضافة إلى ذلك، يوفر النظام المقترح سلامة البيانات من خلال تطبيق دالة (HASH 256) لتوليد قيمة (HASH). تركز خوارزمية التشفير خفيفة الوزن المقترحة على الاستخدام الأمثل لموارد أجهزة إنترنت الأشياء، بحيث توفر بشكل كبير كل من (المعالج، الذاكرة، الطاقة، الوقت، وعرض النطاق الترددي (لا حاجة لتوزيع المفاتيح)) من ناحية أخرى، مما يوفر أمانًا عاليًا، خاصة ضد محلل التشفير. بالإضافة إلى ذلك، تتمتع خوارزمية التشفير خفيفة الوزن المقترحة بالقدرة على معالجة النصوص والأرقام باللغتين الإنجليزية والعربية. أيضًا، لتحقيق تكامل البيانات في النظام المقترح ضمن بيئة إنترنت الأشياء، تم استخدام (٤) أرقام عشرية سداسية من قيمة (HASH) بدلاً من قيمة (HASH) الأصلية المكونة من (٦٤) رقمًا عشريًا سداسيًا من أجل تقليل عرض النطاق الترددي للشبكة ومعالجتها وتخزينها.

الكلمات الدالة: تكامل البيانات، انترنت الأشياء، خوارزمية التشفير خفيفة الوزن، وسيط Mosquitto، بروتوكول MQTT، راسبيري باي.

1.INTRODUCTION

Cyberspace, an electronic and communication device-based virtual and dynamic environment, stores and uses electronic data [1]. The primary purpose of making Cyberspace is to communicate and share information across the Internet. However, it has a significant impact on human daily activities. Yahoo, Google, and Facebook are some examples of Cyberspace. In addition, various emerging technologies are involved in a cyberspace communication network, which includes Artificial Intelligence (AI) [2, 3], Blockchain Technology, Big Data, Mobile Cloud Computing (MCC), Cloud Computing (CC), Internet of Things (IoT), and Big Data Analytics [4-8]. The IoT environment consists of many different devices (objects or things) connected using wired or wireless communication technologies, allowing them to collect, exchange, and process data occasionally. These devices range from smart devices for households, such as intelligent bulbs, temperature sensors, and IP cameras, to more advanced devices, such as accelerometers, Radio Frequency Identification Devices (RFID), heartbeat detectors, tablets, iPads, and other sensors in automobiles [9]. Furthermore, as IoT networks spread globally, more devices will eventually be connected, which is predicted to reach 75 billion by 2025 [10]. These devices manage various tasks in the smart home, companies, smart cities, or government institutions through the Internet. IoT devices interact with their actuators without human interaction, known as Machine-to-Machine (M2M) interaction [11]. Data exchange between network nodes, such as sensors, actuators, services, or other platforms, is conducted via message brokers, a key component of middleware systems. The Broker may be operating locally (on edge) or remotely (in the cloud) [12]. The publish/subscribe pattern is a communication architecture where sensors and actuators function as publishers and subscribers. Applications for users can serve as either publishers or subscribers based on each application's design [13, 14]. The primary attributes of IoT networks include heterogeneity, extensive deployment, proximity-based communication, cost-effectiveness, energy efficiency, dynamic network modifications, minimal delay, highly reliable communication, safety, and intelligence. Also, IoT devices have certain features, such as smaller sizes and lower computation capacity. Moreover, they use particular lightweight communication protocols [15]. On the other hand, the IoT has many restrictions regarding devices and components constrained by processor speed, memory size, communication bandwidth, power consumption, and its heterogeneous nature and widespread use [16]. Traditional protocols, Transmission Control Protocols, and User Datagram Protocols (TCP & UDP) do not work well with resource-constrained devices. Thus, various messaging and communication protocols have been designed to enable the reliable and secure transmission of data across the Internet of Thing networks, such as Bluetooth, Hyper Text Transfer Protocol (HTTP), ZigBee, Long Range Wide Area Network (LoRa WAN), M2M, MQTT, Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), and Constrained Application Protocol (CoAP)) [17]. The MQTT protocol is considered the most superior option in the Internet of Things (IoT) field. For example, MQTT has gained significant popularity in the smart homes domain [18], agricultural Internet of Things [19], and applications in the industrial sector [20], because it has lower requirements

for memory and power, less packet loss, and supports communication at lower bandwidths [6, 17]. Nevertheless, the lack of security features in MQTT is its biggest flaw. The devices' processing overhead has increased due to the current methodologies, which are also still exposed to several attacks [21]. IoT technology is being applied in an increasing number of fields. Therefore, they are more vulnerable to privacy and security attacks for the following reasons: increasing popularity; connected to the Internet; lack a standard; controlled remotely; provided with lower security policies, and consume sensitive data. Therefore, they continue to be practical and affordable [22]. Traditional encryption techniques and systems are sometimes unsuitable for IoT networks due to their design for devices with ample resources on high-bandwidth networks. As a result, researchers suggested using lightweight encryption algorithms to address this issue and ensure security and privacy in an IoT context while minimizing resource consumption and reducing encryption overhead. These methods are lightweight, keyless (using symmetric keys), robust, and provide end-to-end Encryption [23]. The greatest challenge of implementing a complete paradigm of IoT is the privacy and security issues. Internet of Things devices are restricted in terms of resources, i.e., limited to a few megahertz of computational power, 5–10 Megabytes of memory, and a small amount of persistent storage available. In addition, they are battery-powered; their power capacity spans from 6 to 24 hours. These limitations prevent IoT devices from implementing resource-intensive security measures, such as security information, event management, energy requirements, big data, heterogeneity, and scalability. Moreover, IoT devices' growth (scale) is another critical challenge. By 2025, it is projected that the number of Internet-connected gadgets will almost triple. The development of IoT by increasing the number of devices makes it a fragile environment that attracts a wide range of attacks. To preserve human life and resources, numerous environmental monitoring applications can be used to monitor air pollution, detect earthquakes early, detect forest fires, etc. These benefits are subject to limitations and problems, including privacy and security. The most significant issues networks have faced throughout the years are security, privacy, complexity, big data, energy consumption, devices/links, and heterogeneity challenges.

2.LITERATURE REVIEW

Several recent studies have successfully developed encryption algorithms and strategies specifically designed for the Internet of Things (IoT) environment. The authors tackle the inherent constraints of IoT devices that pose challenges in implementing conventional encryption methods, which require substantial resources to operate. To provide secure communication and efficiency while utilizing resources sensibly, several researchers have proposed lightweight encryption techniques to address the resource limitation challenge and enhance traditional encryption techniques for IoT devices. The researchers in [24] proposed a scheme to address security and privacy issues in the IoT. A lightweight, pair less Attribute-Based Encryption scheme (ABE) based on the Elliptic Curve Cryptography (ECC) was developed. The safety of the proposed scheme was demonstrated using the trait-based selective group model. It was predicated on the Elliptic Curve Diffie-Hellman Decision (ECDHD) assumption, as opposed to the bilinear Diffie-Hellman assumption. Detailed comparison analyses were conducted with existing ABE schemas by establishing metrics to measure computational overhead, standardization, and communication overhead [24]. The findings indicated that the implementation efficacy was enhanced; in addition to that, the communication costs decreased for the proposed system. The researchers in [25] proposed an algorithm for the Secure Internet of Things (SIT), which is a name for a lightweight encryption algorithm. It was a 64-bit block cipher that encrypts data with a 64-bit key. The algorithm architecture combined a unified switching network with a Feistel network. The simulations showed that the algorithm with only five encryption rounds provided excellent security. The algorithm is a good candidate for IoT applications because the implementation yields encouraging results [25]. The researchers in [26] described a design for an IoT security application-friendly Advanced Encryption Standard–Galois Counter Mode (AES-GCM) Authenticated Encryption (AE) Crypto-Core. The AES-GCM core offers confidentiality using the AES block cipher's Counter Confidential Treatment Request mode (CTR) and authenticity and integrity using GHASH. Two key lengths are supported by AES encryption: 128-bit and 256-bit. The Field-Programmable Gate Array (FPGA) implementation validated the AES-GCM core [26]. The authors in [27] developed and tested a new security method for Voice over Internet Protocol wireless networks (VoIP). New encryption methods were designed to satisfy Quality of Service (QoS) criteria for voice communication and to work with wireless devices. Its goal is to reduce the power consumption and execution time of the encryption process while increasing or maintaining the level of security [27]. This algorithm is made more difficult to hack and secure with the new triple key technology [27]. The researchers in [28] introduced an advanced

approach that ensures data protection using cryptographic methods like Rivest Shamir Adleman (RSA). Additionally, it emphasized the establishment of a random pathway for transmitting messages from the sender to the receiver. This strategy guarantees the safe transmission of packets, even in the presence of attacks on intermediary nodes, by establishing alternative routes between the source and destination [28]. The proposed system performs the following tasks [28]:

1) It is proposed to use a new Identity Path and Location (IPL) privacy algorithm to ensure that the identities and locations of source nodes are not disclosed.
2) It also ensures that packets reach their intended location using only reliable middle nodes.
3) The current intermediate node is kept anonymous.
4) A novel approach is suggested to establish data privacy by offering payload encryption and decryption.
5) They do not realize the intermediate nodes of their previous path.
6) Two critical algorithms were created to ensure the security and integrity of the data transmitted through the sensor network.

The researchers in [23] presented Elastic Encryption Technology (FlexenTech), a new, scalable encryption method used to protect IoT data stored and transmitted. FlexenTech is suitable for networks and devices with limited resources. It protects against replay attacks while providing high-speed Encryption. In addition, it is worth noting that this mode is highly dependable since it can accommodate a wide range of key sizes and cycles. The system provides rapid Encryption, safeguards against typical attacks like replay attacks, and establishes a customizable mode that permits the utilization of various rounds or key sizes. The several tiers of secrecy demonstrate the robustness of FlexenTech in conducting experimental analysis by enabling the implementation of diverse protective measures. The authors of Ref. [29] proposed a scheme for transmitting text files among embedded Internet of Things Devices. An innovative, robust, and efficient symmetric encryption algorithm was presented, together with lightweight encryption methods specifically designed for IoT applications. The researchers evaluated the effectiveness of the lightweight coding method by analyzing its bit error probability and throughput in an Additive White Gaussian Noise (AWGN) channel with Rayleigh vanishing. They assumed that the signal being sent via the channel had a certain bit error probability. To quantify these performance metrics, they formulated analytical formulas that consider modulated signals both with and without coding. Furthermore, they suggested the development of a lightweight encryption algorithm to enhance the security level without substantial alterations in processing speed and size. The experimental data under study showed that the lightweight encryption algorithm outperformed the proposed and literature lightweight encryption algorithms regarding error rate and throughput performance, and outperformed traditional encryption algorithms, such as AES. The researchers of Ref. [30] presented an advanced data encryption method suitable for Internet of Things (IoT) applications. The Catalan object, which acts as a cipher key in the cipher system, provides ciphers based on combinatorial structures with non-overlapping or non-intersecting matching. The experimental section compared the proposed encryption method with the Data Encryption Standard (DES) algorithm and Catalan numbers. This analysis used the machine learning-based definition of the cipher method using only ciphertext [30], as shown in Fig. 1.

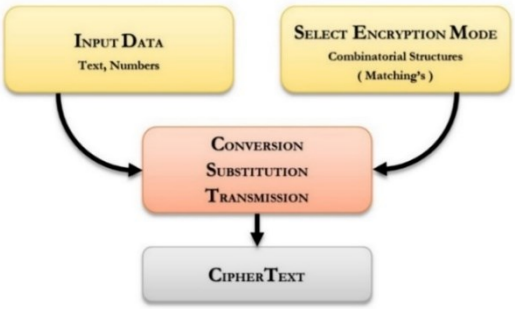

Fig. 1 Proposed Model Framework [30].

The researchers of Ref. [31] examined various new technologies and security procedures and their potential vulnerabilities to equip Industrial IoT infrastructure builders with the necessary knowledge to account for them. The text explored the distinctions and similarities between classical information technology (IT) and operational technology (OT) and their connections to industrial IoT systems. It also examined potential attacks on both the IT and OT layers. The researchers of Ref. [21] introduced the Robust Security Scheme (RSS) as a comprehensive plan to protect MQTT from any vulnerabilities that may result in advanced cyberattacks. The proposed RSS used two encoding schemes:

1) D-AES, or a Dynamic version of the Advanced Encryption Standard.
2) KP-ABE, or Key Policy Attribute-Based Encryption.

RSS introduces D-AES, a novel design architecture that proposes a symmetric AES method to achieve encryption of the MQTT payload. Furthermore, the second component of the hybrid cypher system, KP-ABE, is employed for the private key cypher of D-AES to mitigate the computational constraints

associated with bilinear mappings. The performance of the suggested RSS can be evaluated by measuring processing time and traffic. The empirical findings demonstrate the efficacy of the suggested D-AES algorithm. It is more promising and has advantages over the traditional AES algorithm. In the literature review, it was noticed that most encryption algorithms use keys that need protection, management, and distribution, whether of the Symmetric or Asymmetric keys, which require high processing and consume energy in IOT devices. In addition, these algorithms do not verify the integrity of the text data. Furthermore, most of these algorithms work to encrypt text only, and some of these algorithms are breakable by the crypto analyzer. Therefore, this paper developed a lightweight encryption algorithm that works with an implicit key (Symmetric key) to encrypt numbers and letters, as it depends on the location of the sequence of numbers and letters in plain text while achieving data integrity. Thus, less processing time can be achieved with no need for operations, including protecting, managing, and distributing keys, which provides the user with high security, while saving energy for IOT devices. Moreover, it is robust against the crypto analyzer.

3.PROPOSED ENCRYPTION ALGORITHM

New encryption methods proposed are called Lightweight Security and Integrity Data – Internet of Things *LSID-IoT*. The primary benefits of the proposed methods are their simplicity and efficiency in implementation. The implicit key is derived from the plain text. The algorithm is categorized as a substitution encryption method since it substitutes the characters of the plain text with different ones to generate the ciphertext, encrypts both text (English and Arabic) and numbers, and is robust against a crypto analyzer. In addition, it is End-to-End Encryption.

3.1.Encryption TEXT Methods

Before Encryption can happen, the plaintext must first be turned into a set of symbols. Next, for each character, the encryption alphabet can be found by moving the original alphabet based on the order of the character in the plain text. After that, go from the matching encrypted character to the plain character. Equation (1) expresses how to use the encryption method. The encryption process is shown in Fig. 2 as a diagram.

$$C[I] = (P[I] + (I \bmod 26)) \bmod 26 \quad (1)$$

where C [] is an array containing Cipher text, P [] contains Plain text, and I is the character index in plain or cipher text.

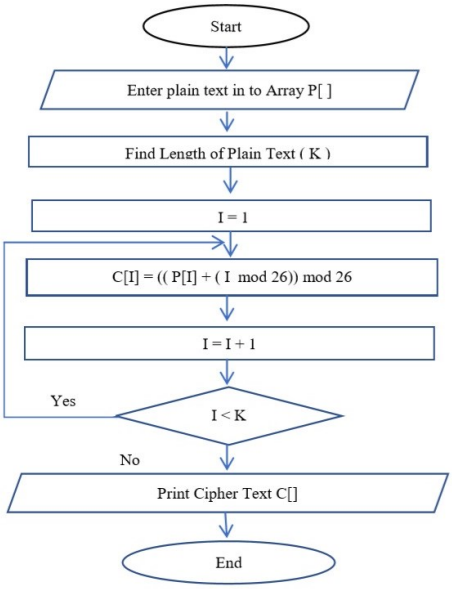

Fig. 2 Flowchart of the Encryption Process.

Figure 3 shows an example of text encryption.

Plain Text = **banana**

1	2	3	4	5	6
b	**a**	**n**	**a**	**n**	**a**

C [0] = ASC (**b**) + (0 mod 26) = **b**
C [1] = ASC (**a**) + (1 mod 26) = **b**
C [2] = ASC (**n**) + (2 mod 26) = **p**
C [3] = ASC (**a**) + (3 mod 26) = **d**
C [4] = ASC (**n**) + (4 mod 26) = **r**
C [5] = ASC (**a**) + (5 mod 26) = **f**

Cipher Text = **bbpdrf**

Plain Text	b	a	n	a	n	a
Cipher Text	b	b	p	d	r	f

Fig. 3 Example of Text Encryption.

3.2.Decryption TEXT Methods

The decryption process is the reverse of the encryption process. It involves taking the ciphertext and transforming it back into a set of characters, then deciphering the characters by determining the decryption alphabetically by shifting the original alphabet according to the sequence of the characters in the ciphertext. The final stage is determining the equivalent plain character for each cipher character. The decryption operation is executed by Eq. (2). Fig. 4 depicts the flowchart illustrating the steps involved in the decryption process.

$$P[I] = (C[I] - (I \bmod 26)) \bmod 26 \quad (2)$$

where C [] is an array containing Cipher text, P [] contains Plain text, and I is the character index in plain or cipher text.

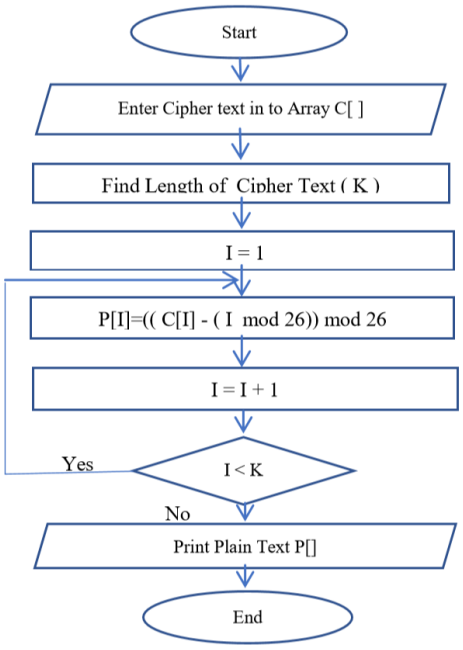

Fig. 4 Flowchart of the Decryption Process.

Figure 5 shows an example of text decryption.

Cipher Text = **bbpdrf**

1	2	3	4	5	6
b	**b**	**p**	**d**	**r**	**f**

P [0] = ASC (**b**) - (0 mod 26) = **b**
P [1] = ASC (**b**) - (1 mod 26) = **a**
P [2] = ASC (**p**) - (2 mod 26) = **n**
P [3] = ASC (**d**) - (3 mod 26) = **a**
P [4] = ASC (**r**) - (4 mod 26) = **n**
P [5] = ASC (**f**) - (5 mod 26) = **a**

Plain Text = **banana**

Cipher Text	b	b	p	d	r	f
Plain Text	b	a	n	a	n	a

Fig. 5 Example of Text Decryption.

3.3.Encryption and Decryption NUMBERS Methods

Furthermore, the identical encryption process was employed to encrypt the numbers by making straightforward modifications, explicitly converting the numerical digits into characters with adding 17 to the American Standard Code for Information Interchange (ASCII) value of the number and subsequently encrypting the resulting character. For instance, to encrypt any given number, Eq. (3) is used.

$$\mathbf{C[I] = ((P[I] + 17) + (I \bmod 26)) \bmod 26} \quad \mathbf{(3)}$$

where C [] is an array containing Cipher text, P [] contains Plain text, and I is the character index in plain or cipher text. Figure 6 shows an example of how to encrypt a number.

Plain Number = **357621**

1	2	3	4	5	6
3	**5**	**7**	**6**	**2**	**1**

C [0] = ASC (**3**) + 17 + (0 mod 26) = **D**
C [1] = ASC (**5**) + 17 + (1 mod 26) = **G**
C [2] = ASC (**7**) + 17 + (2 mod 26) = **J**
C [3] = ASC (**6**) + 17 + (3 mod 26) = **J**
C [4] = ASC (**2**) + 17 + (4 mod 26) = **G**
C [5] = ASC (**1**) + 17 + (5 mod 26) = **G**

Cipher Number = **DGJJGG**

Plain Num.	3	5	7	6	2	1
Cipher Num.	D	G	J	J	G	G

Fig. 6 Example of a Number Encryption.

Equation (4) is used to decrypt any number.

$$\mathbf{P[I] = ((C[I] - 17) - (I \bmod 26)) \bmod 26} \quad \mathbf{(4)}$$

where C [] is an array containing Cipher text, P [] contains Plain text, and I is the character index in plain or cipher text. The proposed methods could also deal with the Real numbers. Figure 7 shows an example of decrypting a Number.

Cipher Number = **DGJJGG**

1	2	3	4	5	6
D	**G**	**J**	**J**	**G**	**G**

P [0] = ASC (**D**) - 17 - (0 mod 26) = **3**
P [1] = ASC (**G**) - 17 - (1 mod 26) = **5**
P [2] = ASC (**J**) - 17 - (2 mod 26) = **7**
P [3] = ASC (**J**) - 17 - (3 mod 26) = **6**
P [4] = ASC (**G**) - 17 - (4 mod 26) = **2**
P [5] = ASC (**G**) - 17 - (5 mod 26) = **1**

Plain Number = **3 5 7 6 2 1**

Cipher Num.	D	G	J	J	G	G
Plain Num.	3	5	7	6	2	1

Fig. 7 Example of a Number Decryption.

3.4.Integrity

To achieve integrity service in the proposed system, the Secure Hash Algorithm SHA-256 (256 bit) algorithm was used, as shown in Fig. 8. The first and last two hexadecimal numbers of the output of the above algorithm were used to reduce the size of the sent message and achieve lightweight, in addition to using low network bandwidth.

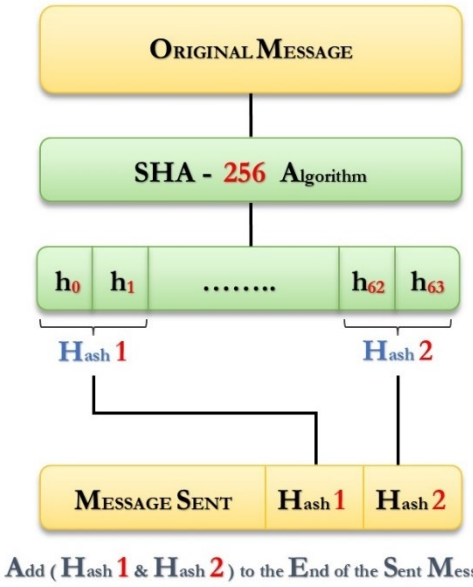

Fig. 8 Flowchart of the Integrity Service for the Proposed System.

4.DESIGN SUGGESTED SYSTEM

The proposed system was designed within the IoT environment in two stages:

4.1.Software Design Stage

At this stage, the proposed system was designed into two main parts:

- **The First Part** includes Encryption and integrity of the sent message, as illustrated in the flowchart in Fig. 9.

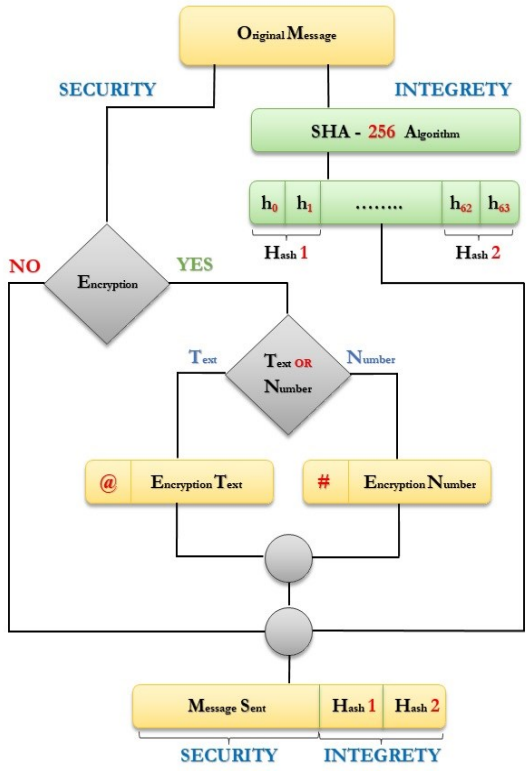

Fig. 9 Encryption Flowchart.

- **The Second Part** includes decrypting and integrity of the received message, as illustrated in the flowchart in Fig. 10.

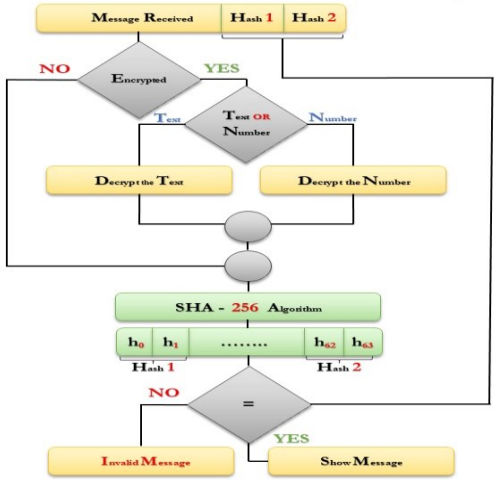

Fig. 10 Decryption Flowchart.

4.2.Hardware Design Stage

In addition to using two computers, a Raspberry Pi4 device containing Mosquitto Broker that acts as a coordinator for MQTT messages was used. The first computer, PC1, is used to send MQTT messages (Publisher Device), and the second computer, PC2, is used to receive MQTT messages (Subscriber device), as shown in Fig. 11.

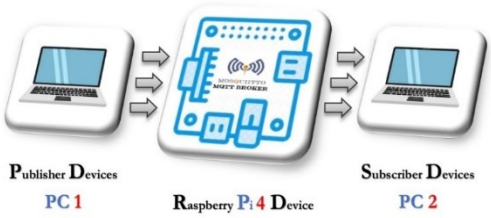

Fig. 11 General IoT Network for the Proposed System.

5.TESTING PROPOSED SYSTEM ON PCS

5.1.Testing the Proposed System On Raspberry Pi

The testing phase of the proposed system was done after finishing the entire configuration by implementing the tenser flow (64-bit) environment on Linux OS (32-bit) and completing the appropriate environment for executing the proposed system on Raspberry Pi devices. The proposed system consists of the following steps:

1) Define the used libraries, such as PYSHARK, asyncio, and numpy.
2) Determine if the proposed system shows the MQTT Message in the command windows.
3) Capture operation starts on the Wireless Local Area Network (WLAN) interface and captures only the MQTT messages using the PYSHARK library.
4) Extract the message filed from Publish MQTT messages.
5) Show the Extract message depending on step 2.

5.2.Testing Proposed System On PC's

At this stage, two applications were developed using the Python language within the Anaconda environment.

- **The First Application**
  It sends MQTT messages to the Broker inside the Raspberry device using the MQTT protocol. The development includes the following steps:
  1) Create the program interface using the standard Graphical User Interface (GUI) library for Python (Tkinter).
  2) Develop an Integrity function to compute the HASH value.
  3) Giving the user the ability to specify whether he/she wants to encrypt the message or not. If the user chooses to encrypt the message, the proposed encryption algorithm is called to encrypt carry out the process of encrypting the message.
  4) Connect to the Broker by providing the username and password.
  5) Send the message, whether encrypted or unencrypted, to the Broker using the Publish function of the Paho library, specifying the Topic within the Broker that the message is stored in.
- **The Second Application**
  It receives MQTT messages from the Broker inside the Raspberry device using the MQTT protocol. The development includes the following steps:
  1) Create the program interface using the standard GUI library for Python (Tkinter).
  2) Develop a particular function to provide integrity.
  3) Connect to the Broker by providing the username and password.
  4) Listen to the port no. (1883) and determine the Topic to receive the MQTT message.
  5) Receive the MQTT message from the Broker and check whether it was encrypted or not; if encrypted, the application calls the decrypt function in the proposed algorithm to decrypt the message.
  6) Compute the HASH value of the MQTT message (after decrypting the message) and compare it with the received HASH value. If equal, the application will display the MQTT message in the list box. Otherwise, the application will display an invalid string message in the list box.

6.RESULTS

Many different experiments were conducted to verify the proposed system's performance. This verification was done by performing a set of scenarios that included encrypting and decrypting texts in Arabic and English, as well as numbers, and merging them. To realize the efficiency of the proposed system, the time difference between processing the received original message (without Encryption) and decrypting the encrypted message was calculated.

6.1.Scenario 1

This Scenario includes sending a message (English Text only) once without Encryption and with Encryption, in addition to calculating the received time as follows:

1) Send the message (**without Encryption**), as shown in Fig. 12.
2) Send the message (**with Encryption**), as shown in Fig. 13.

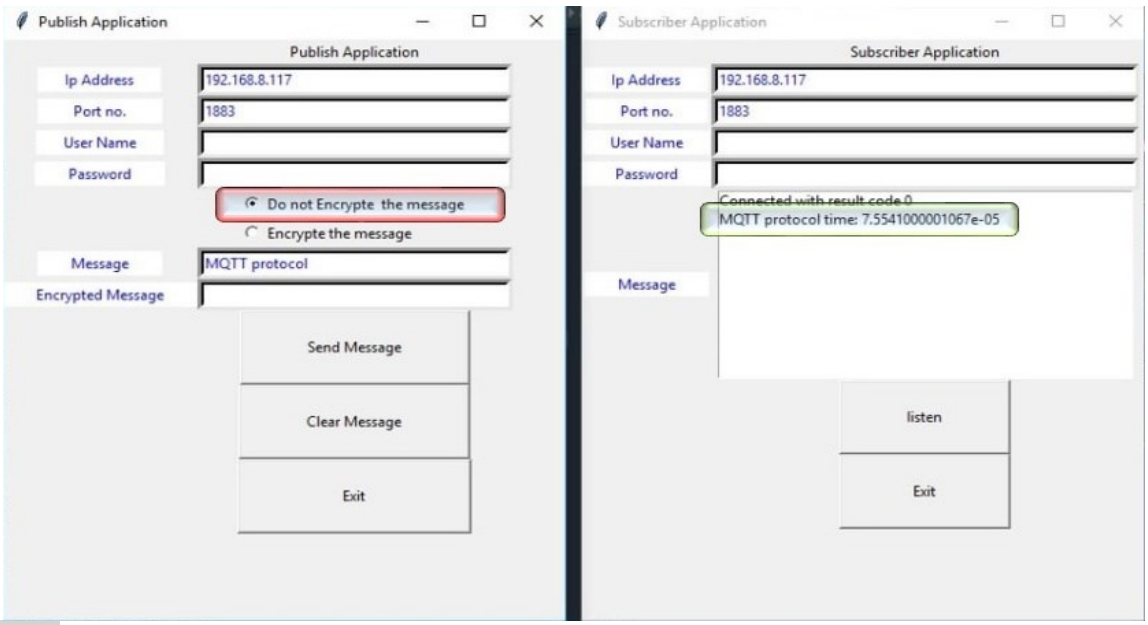

Fig. 12 The Interface of the Proposed System to Send messages (without Encryption), with calculating the received Time in Scenario (1).

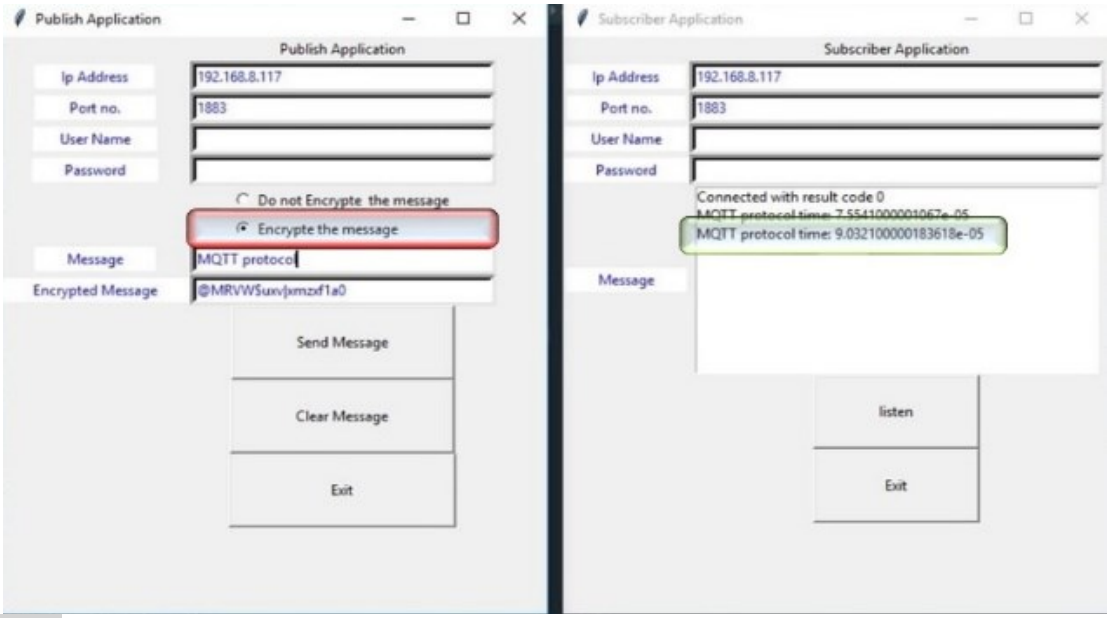

Fig. 13 The Interface of the Proposed System to Send messages (with Encryption), with calculating the received Time in Scenario (1).

Then, the difference between the two times of the two cases above is calculated.

6.2.Scenario 2

This Scenario includes sending a message (Number only) once without Encryption and with Encryption, in addition to calculating the received time as follows:

1) Send the message (**without Encryption**), as shown in Fig. 14.
2) **2.** Send the message (**with Encryption**), as shown in Fig. 15.

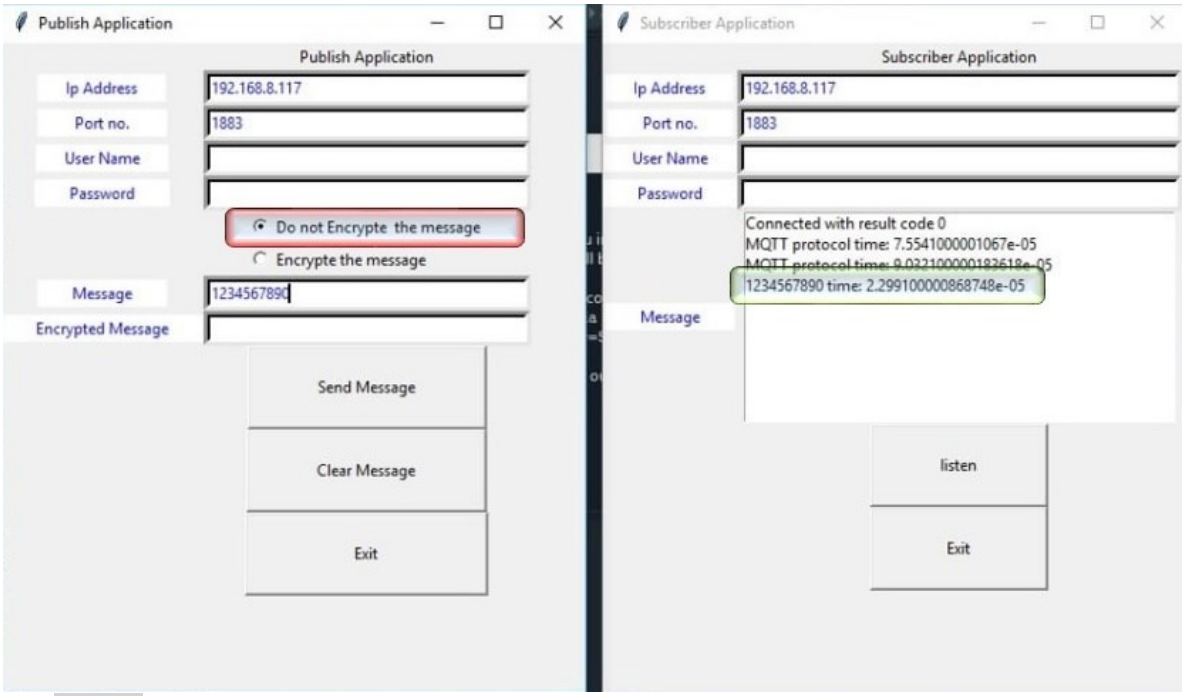

Fig. 14 The Interface of the Proposed System to Send a message (without Encryption), with Calculating the Received Time in Scenario (2).

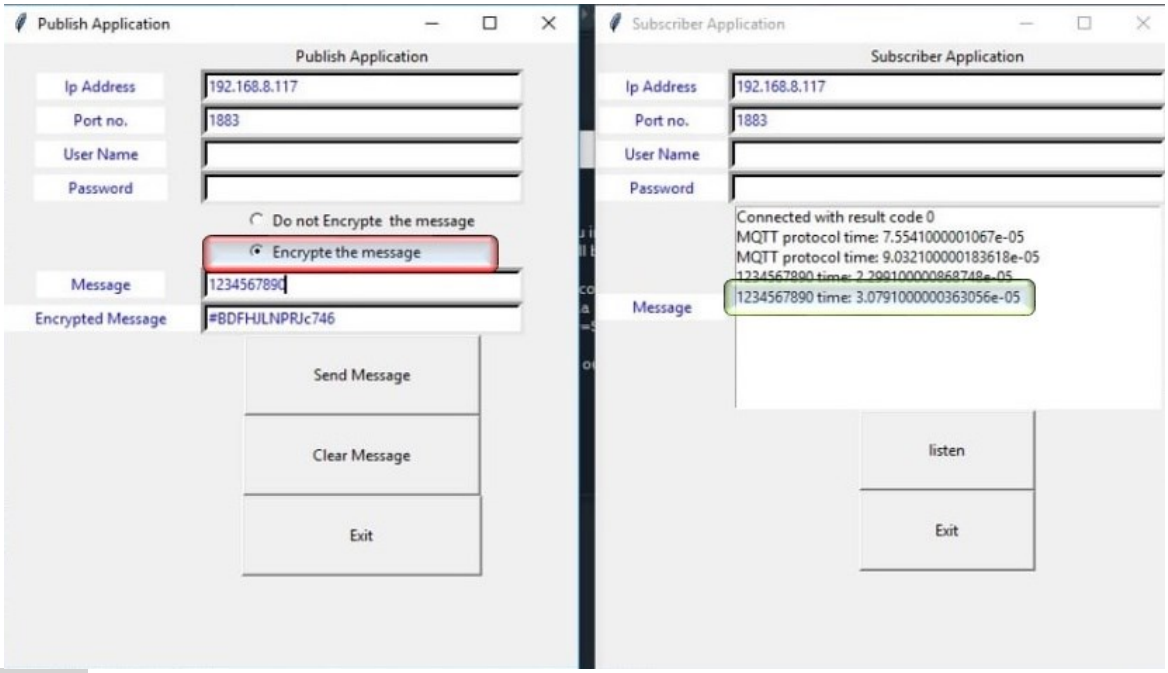

Fig. 15 The Interface of the Proposed System to Send messages (with Encryption), with Calculating the Received Time in Scenario (2).

6.3.Scenario 3

This Scenario includes sending a message **(English Text with a Number)** once without Encryption and with Encryption, in addition to calculating the received time as follows:

1) Send the message (**without Encryption**), as shown in Fig. 16.
2) Send the message (**with Encryption**), as shown in Fig. 17.

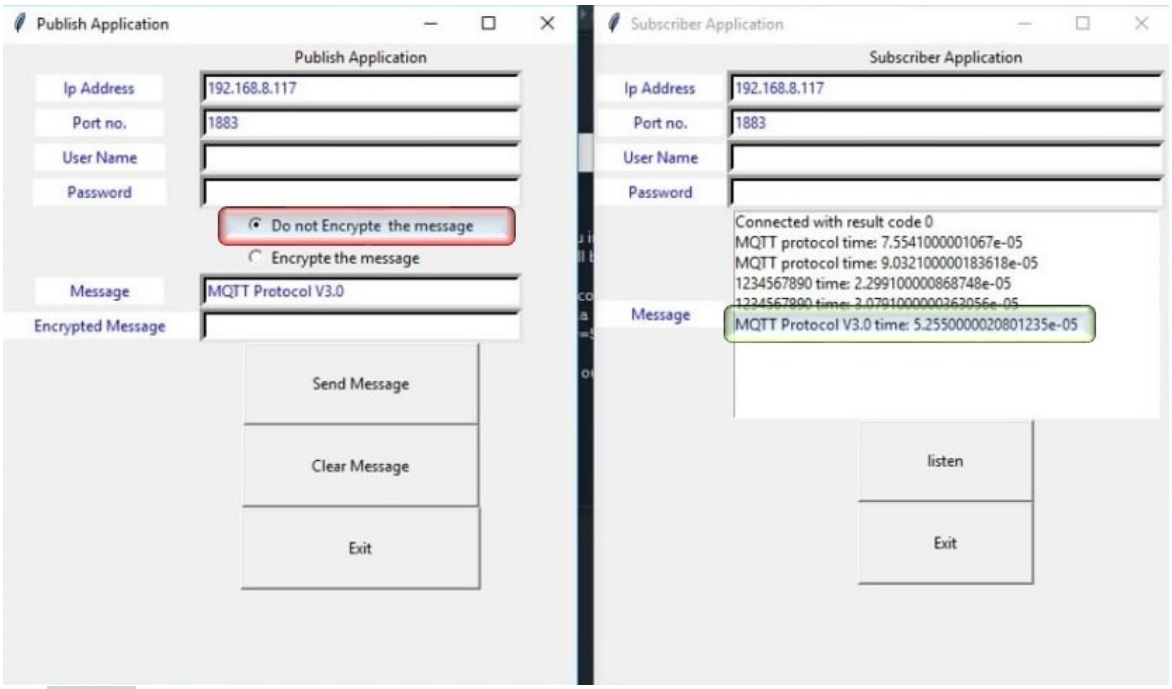

Fig. 16 The Interface of the Proposed System to Send a message (without Encryption), with Calculating the Received Time in Scenario (3).

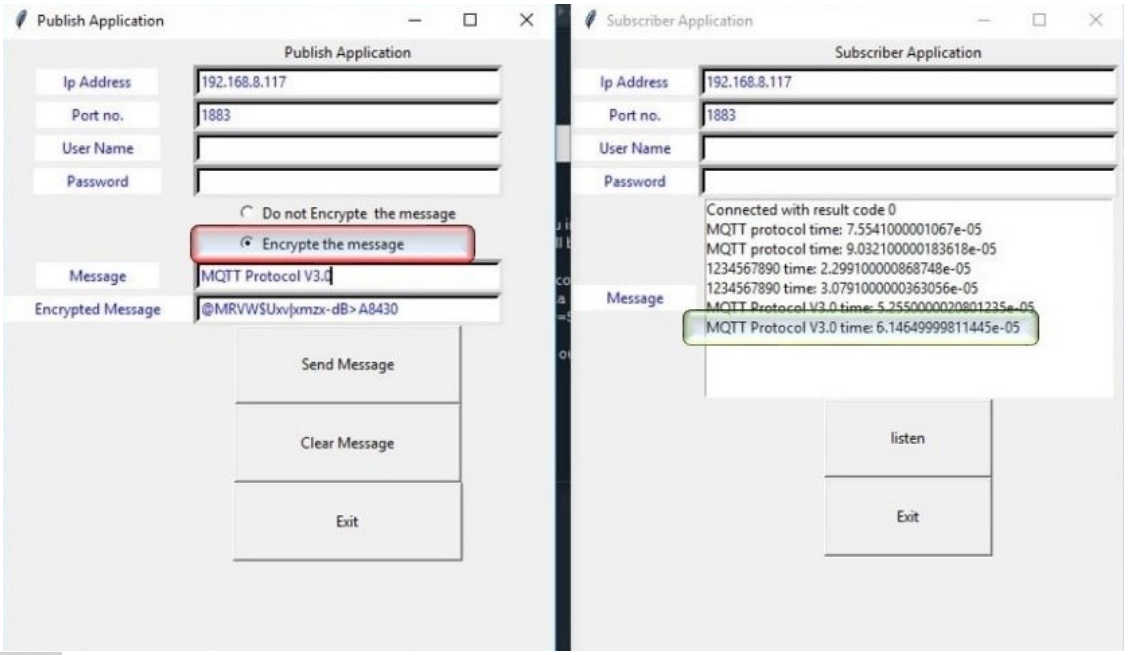

Fig. 17 The Interface of the Proposed System to Send a message (with Encryption), with Calculating the Received Time in Scenario (3).

6.4.Scenario 4

This Scenario includes sending a message (**Arabic and English Texts and Numbers**) once without Encryption and once with Encryption, in addition to calculating the received time as follows:

1) Send the message (**without Encryption**), as shown in Fig. 18.
2) Send the message (**with Encryption**), as shown in Fig. 19.

All results of the scenarios of this paper are summarized in Table 1 and Chart 1.

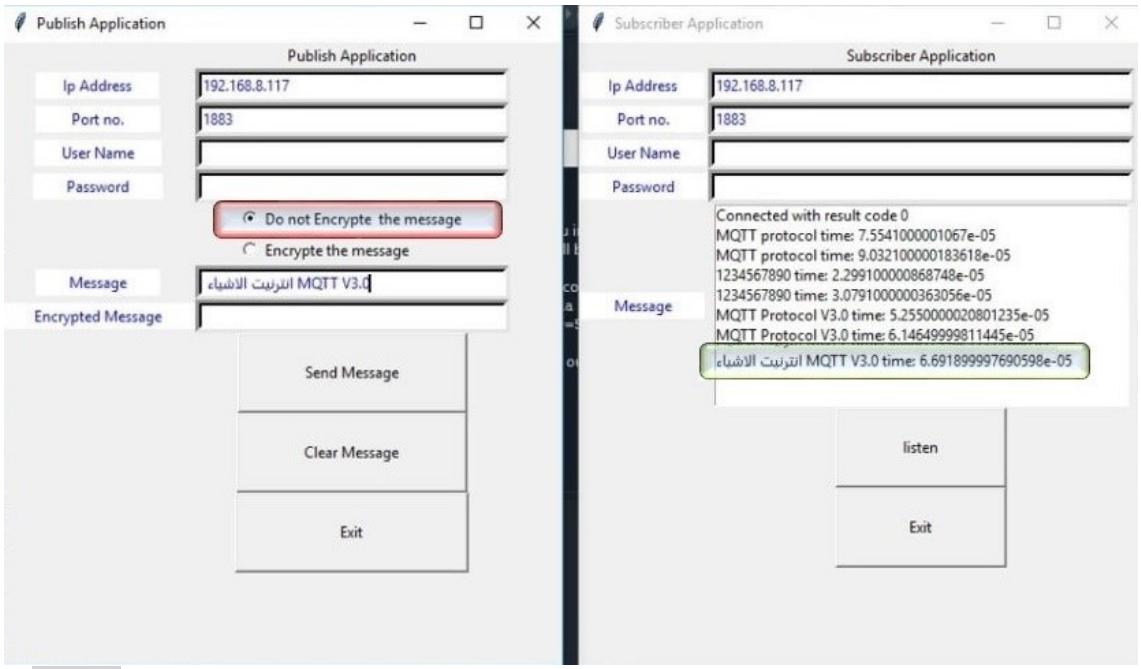

Fig. 18 The Interface of the Proposed System to Send a message (without Encryption), with Calculating the Received Time in Scenario (4).

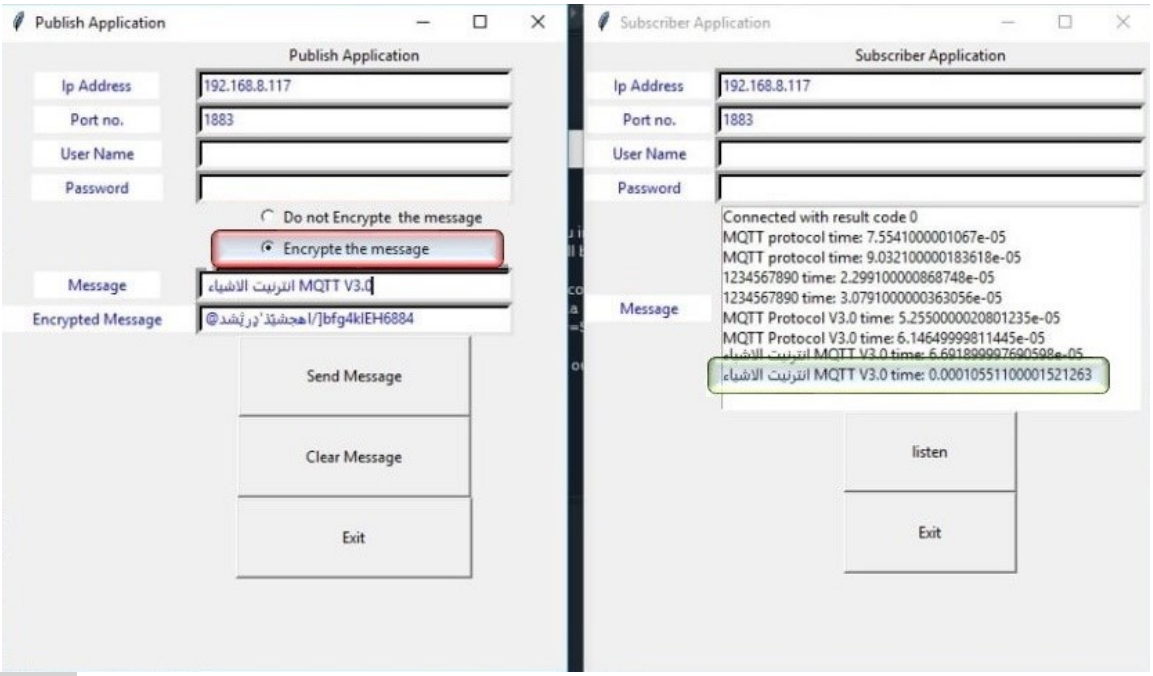

Fig. 19 The Interface of the Proposed System to Send a message (with Encryption), with Calculating the Received Time in Scenario (4).

Table 1 The Results of the Proposed System in the Present Scenarios.

Sc. No.	Message Type	Message	Time without Encryption in second	Time with Encryption in second	Different Time in the second
1	English Text Only	MQTT Protocol	0.000075541	0.000090321	0.0000148000
2	Number Only	1234567890	0.000022991	0.000030791	0.0000078000
3	English Text with Number	MQTT Protocol V3.0	0.000052550	0.000061464	0.000008914
4	Arabic with English Text and Number	انترنيت الاشياء MQTT V3.0	0.000066918	0.000105511	0.000038593

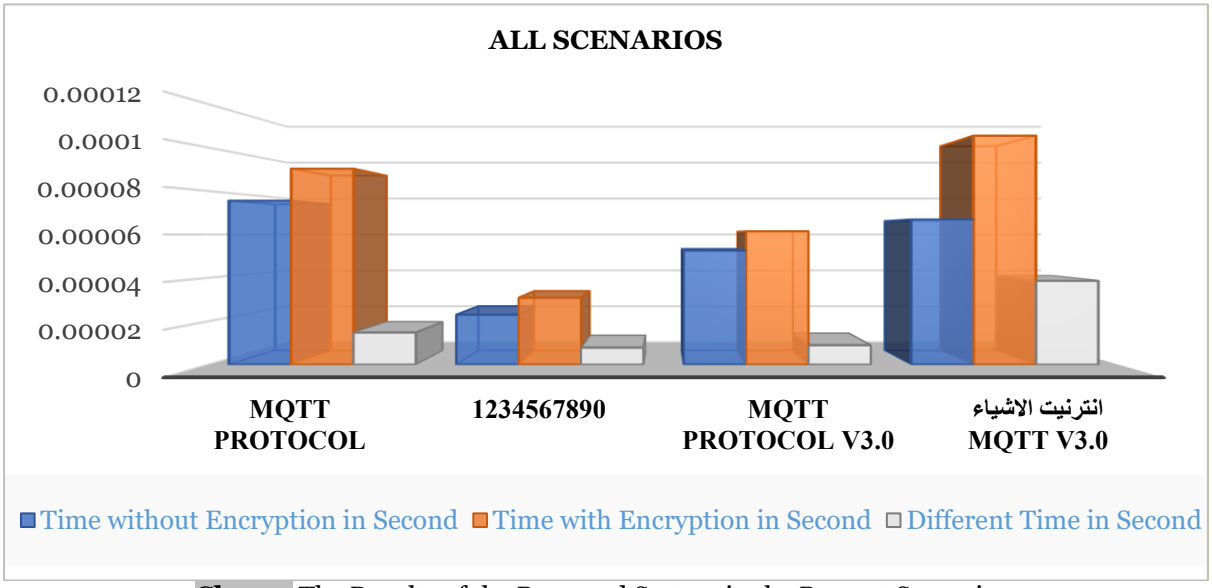

Chart 1 The Results of the Proposed System in the Present Scenarios.

6.DISCUSSIONS

As demonstrated in the present scenarios, the proposed system insignificantly affects the performance of the Raspberry device operating in the Internet of Things environment. It does not need to distribute keys used in encryption and decryption operations because it relies on the implicit key principle, which adds strength to its security, considering the optimal use of Internet of Things resources, i.e., Bandwidth, Memory, Central Processing Unit (CPU), and Energy. In addition, the proposed system is robust against attempts to Crypto Analyzer attack, so that repeated letters in the plaintext do not take the same letters encrypted in the ciphertext, such as the Encryption of the word (**banana**), which, after Encryption, will become (**bbpdrf**). Moreover, the ability of the

proposed system to provide end-to-end Encryption, i.e., the process of Encryption and integrity, will protect data from the sending side to the receiving side, which is also essential. Furthermore, the results in Table 1 show that the proposed system in the present scenarios proved that the time difference is very small between the message in its two states, i.e., with Encryption and without Encryption, This result indicates the possibility of applying the proposed system within the Internet of Things environment, which provides security and data integrity services while maintaining accuracy and speed, in addition to the possibility of dealing with all types of texts and numbers. It should be noted that the time calculation will vary depending on several factors, including:

- The length of the message.
- Specifications of the sending and receiving devices and the Raspberry Pi.
- Network speed.

7.CONCLUSIONS

Nowadays, developing and designing practical and lightweight encryption algorithms and techniques, while considering the resource constraints of connected devices, is the primary security challenge in the IoT environment. Utilizing traditional security mechanisms in an IoT network, such as conventional Intrusion Detection System (IDS) and authentication, will be time-consuming and could cause problems for an extensive system with many connected heterogeneous devices. They are not appropriate for restricted IoT devices. As a result, there is an increasing need for effective lightweight encryption techniques that integrate both lightweight symmetric and asymmetric algorithm features. This work designed and implemented a lightweight encryption method with the following characteristics: Keyless, Encryption & Integrity, Text & Number End-to-End Encryption, Reduce Traffic, and processing overhead. The Time taken for the encryption and decryption processes for the proposed encryption algorithm is very short, and it does not require key distribution as they operate with an implicit key, which makes them ideal for working within the Internet of Things environment. To provide the best Network Bandwidth, four hexadecimal digits from the HASH value were used instead of the original 64 hexadecimal digit HASH value, which provides the data integration service.

ACKNOWLEDGEMENTS

The authors would like to thank the Electrical and Computer Engineering Department, College of Engineering, Altinbas University, for the facilities that have helped enhance the quality of this work.
The research was fully funded by the university, the research advisor, and the researcher.

REFERENCES

[1] Clark D. **Characterizing Cyberspace: Past, Present, and Future**. *Explorations in Cyber International Relations* 2010; **1**(2): 1-18.

[2] Shetty N. **Vulnerability Assessment for Cybersecurity Using Machine Learning**. *Social Science Research Network Electronic Journal* 2021; **1**(1): 1-16.

[3] Smith DL, Anandavalli T, Belonging S. **Ioneliness in Cyberspace: Impacts of Social Media on Adolescents' Well-Being**. *Australian Journal of Psychology* 2021; **73**(1): 12-23.

[4] Zwitter A, Hazenberg J. **Cyberspace, Blockchain, Governance: How Technology Implies Normative Power and Regulation**. In: Cappiello B, Carullo G, editors. *Blockchain, Law and Governance*. Cham: Springer International Publishing; 2021. p. 45-62.

[5] Venish Raja C, Chitra K, Jonafark M. **A Survey on Mobile Cloud Computing**. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 2018; **3**(3): 2096–2100.

[6] Ammar M, Russello G, Crispo B. **Internet of Things: A Survey on the Security of IoT Frameworks**. *Journal of Information Security and Applications* 2018; **38**(1): 8-27.

[7] Zhang D. **Big Data Security and Privacy Protection**. *International Conference on Virtual Reality and Intelligent Systems*; IEEE; 2019. p. 87–89.

[8] Saeed MSM. **Lightweight Cyber Attack Intelligent Detection Model Based on Blockchain in IoT**. Ph.D. Thesis. Mosul University, Mosul, Iraq; 2023.

[9] Hussain F, Hussain R, Hassan SA, Hossain E. **Machine Learning in IoT Security: Current Solutions and Future Challenges**. *IEEE Communications Surveys & Tutorials* 2019; **22**(3): 1686-1721.

[10] Cvitić I, Peraković D, Periša M, Botica M. **Smart Home IoT Traffic Characteristics as a Basis for DDoS Traffic Detection**. *Proceedings of the 3rd European Alliance for Innovation International Conference on*

Management of Manufacturing Systems; University of Zagreb; 2018. p. 1-10.

[11] Vaccari I, Aiello M, Cambiaso E. **SlowTT: A Slow Denial of Service Against IoT Networks**. *MDPI Electronics* 2020; **11**(9): 1-18.

[12] Khan MA, Khan MA, Jan SU, Ahmad J, Jamal SS, Shah AA, Pitropakis N, Buchanan WJ. **A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT**. *MDPI Sensors* 2021; **21**(21): 1-25.

[13] Dizdarević J, Carpio F, Jukan A, Masip-Bruin X. **A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration**. *ACM Computing Surveys* 2019; **52**(4): 1–30.

[14] Kraijak S, Tuwanut P. **A Survey on IoT Architectures, Protocols, Applications, Security, Privacy, Real-World Implementation and Future Trends**. *11th International Conference on Wireless Communications, Networking and Mobile Computing*; IEEE; 2015. p. 1-6.

[15] Dankan Gowda V, Sridhara SB, Naveen KB, Ramesha M, Naveena Pai G. **Internet of Things: Internet Revolution, Impact, Technology Road Map and Features**. *Advances in Mathematics: Scientific Journal* 2020; **9**(7): 4405-4414.

[16] Fenanir S, Semchedine F, Baadache A. **A Machine Learning-Based Lightweight Intrusion Detection System for the Internet of Things**. *Revue d'Intelligence Artificielle* 2019; **33**(3): 203–211.

[17] Al-Masri E, Kalyanam KR, Batts J, Kim J, Singh S, Vo T, Yan C. **Investigating Messaging Protocols for the Internet of Things (IoT)**. *IEEE Access* 2020; **8**(1): 94880-94911.

[18] Cornel-Cristian A, Gabriel T, Arhip-Calin M, Zamfirescu A. **Smart Home Automation with MQTT**. *54th International Universities Power Engineering Conference*; IEEE; 2019. p. 1–5.

[19] Mukherji SV, Sinha R, Basak S, Kar SP. **Smart Agriculture Using Internet of Things and MQTT Protocol**. *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing*; IEEE; 2019. p. 14–16.

[20] Atmoko RA, Yang D. **Online Monitoring & Controlling Industrial Arm Robot Using MQTT Protocol**. *International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics)*; IEEE; 2018. p. 12–16.

[21] Hintaw AJ, Manickam S, Karuppayah S, Aladaileh MA, Aboalmaaly MF, Laghari SUA. **A Robust Security Scheme Based on Enhanced Symmetric Algorithm for MQTT in the Internet of Things**. *IEEE Access* 2023; **11**(1): 43019–43040.

[22] Ghannadrad A. **Machine Learning-Based DoS Attacks Detection for MQTT Sensor Networks**. M.Sc. Thesis. Politecnico di Milano University, Italy; 2021.

[23] Medileh S, Laouid A, Nagoudi EMB, Euler R, Bounceur A, Hammoudeh M, AlShaikh M, Eleyan A, Khashan OA. **A Flexible Encryption Technique for the Internet of Things Environment**. *Ad Hoc Networks* 2020; **106**(1): 1-16.

[24] Yao X, Chen Z, Tian Y. **A Lightweight Attribute-Based Encryption Scheme for the Internet of Things**. *Future Generation Computer Systems* 2015; **49**(1): 104–112.

[25] Usman M, Ahmed I, Aslam MI, Khan S, Shah UA. **SIT: A Lightweight Encryption Algorithm for Secure Internet of Things**. *International Journal of Advanced Computer Science and Applications* 2017; **8**(1): 1–10.

[26] Sung BY, Kim KB, Shin KW. **An AES-GCM Authenticated Encryption Crypto-Core for IoT Security**. *IEEE International Conference on Electronics Information and Communication* 2018; p. 1–3.

[27] Hazzaa FI. **A New Security Scheme and Lightweight Encryption Algorithm for Voice Over Wireless Networks Connectivity to Internet**. Ph.D. Thesis. Anglia Ruskin University, Chelmsford, United Kingdom; 2019.

[28] Bhaskar AV, Baingane A, Jahnige R, Zhang Q, Zhu T. **A Secured Protocol for IoT Networks**. *International Conference on Virtual Reality and Intelligent Systems*; Cornell University; 2020. p. 1–5.

[29] Khattabi YM, Matalgah MM, Olama MM. **Revisiting Lightweight Encryption for IoT Applications: Error Performance and Throughput in Wireless Fading Channels with and without Coding**. *IEEE Access* 2020; **8**(1): 13429-13443.

[30] Sara˘cevi´c MH, Adamovi´c SZ, Miškovic VA, Elhoseny M, Ma˘cek ND, Selim MM, Shankar K. **Data Encryption for Internet of Things Applications Based on Catalan Objects and Two Combinatorial Structures**. *IEEE Transactions on Reliability* 2021; **70**(2): 819–830.

[31] Abdul-Jabbar MM, Jassim J. **Securing Industrial Internet of Things (Industrial IoT) - A Review of Challenges and Solutions**. *Al-Rafidain Engineering Journal* 2023; **28**(1): 312–320.