Ingénierie des Systèmes d'Information
Vol. 29, No. 5, October, 2024, pp. 1667-1677
Journal homepage: http://iieta.org/journals/isi

Integrating Homomorphic Encryption in IoT Healthcare Blockchain Systems

Habib Aissaoua[1*], Abdelkader Laouid[2], Mostefa Kara[3], Ahcène Bounceur[4], Mohammad Hammoudeh[4], Khaled Chait[2]

[1] LRSD Laboratory, Department of Computer Science, University Ferhat Abbas Setif 1, Setif 19000, Algeria
[2] LIAP Laboratory, University of El Oued, PO Box 789, El Oued 39000, Algeria
[3] ICS Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Kingdom of Saudi Arabia
[4] Information Systems Department, University of Sharjah, University City, Sharjah 27272, United Arab Emirates

Corresponding Author Email: habib.aissaoua@univ-setif.dz

https://doi.org/10.18280/isi.290501

Received: 16 January 2024
Revised: 9 July 2024
Accepted: 2 August 2024
Available online: 24 October 2024

Keywords:
distributed systems, Internet of Things (IoT), blockchain, homomorphic encryption, E-health

ABSTRACT

The Internet of Things (IoT) has recently been implemented in various applications. An IoT network is a group of Internet-connected computing devices embedded in everyday objects. Usually, those devices can interact with each other via the Internet by exchanging data. Because of privacy and security requirements, users of IoT-connected objects need measures to secure corresponding data during storage and transmission. This work presents an architecture that integrates IoT devices, blockchain technology, and embedded homomorphic encryption to ensure high computation speed and security levels in IoT systems. The ultralight linear encryption technique enhances computation speed, making it possible to obtain homomorphic addition over the encrypted data. Using this technique, each administrator gains specific access to the encrypted data. Security, stability, traceability, and anonymity are provided through blockchain technology, which is used to store data. The proposed architecture is demonstrated via a use case from the healthcare sector. The experimental analysis shows our technique's effectiveness in energy consumption reduction and privacy preservation with minimal computation and communication costs. By using five fields per record, name, ID, age, gender, and blood glucose level, we achieved an encryption time equals 0,19 ms and decryption time equals 0,98 ms vs. 15,7 and 1,6 respectively in the best comparison technique. In communication cost, we achieved 1KB vs. 5KB.

1. INTRODUCTION

In recent years, the Internet of Things (IoT) gained significant attention due to its benefits and applications. The goal of IoT is to connect any object at any time, in any location. "Things" in an IoT environment are outfitted with the ability to sense, process, and act. IoT devices frequently work together to deliver intelligent and innovative services autonomously. This technology is utilized across various fields, including home automation, environmental monitoring, and healthcare [1]. One primary objective of modern IoT systems is to bring these various application domains together under a single concept known as smart life. IoT architectures may support numerous heterogeneous devices and integrate various communication technologies that enable the connectivity essential to provide the required services to end-users. Different enabling technologies, e.g., Wireless Sensor Networks (WSNs), Radio Frequency IDentification (RFID), and cloud computing have evolved as essential components for developing IoT applications [2]. Objects in IoT systems inherently have limited resources: they possess restricted memory, low processing capacity, and limited computing power.

Integrating blockchain technology with IoT devices and homomorphic encryption may offer significant advantages in terms of security, privacy, and data integrity [3, 4]. However, scalability and computational demands remain critical challenges [5]. Advanced cryptographic techniques, optimized algorithms, specialized hardware, and hybrid approaches are being developed to address these challenges, making these technologies more practical and efficient for real-world applications. In particular, blockchain technology with IoT devices and homomorphic encryption involves specific scalability and computational demands challenges [6]. Blockchain provides a decentralized framework that secures IoT devices against various cyber threats. Each IoT device acts as a node within the blockchain network, enabling secure and tamper-proof recording of data transactions. Smart Contracts, for instance, are self-executing contracts with the terms directly written into code. They automate processes such as device authentication, data sharing, and other operations without any intervention from a central authority. Scalability allows multiple transactions to occur off the main blockchain, reducing the load and increasing transaction throughput. Integrating homomorphic encryption with blockchain allows computations to be performed on encrypted data without decrypting it, preserving privacy. This is crucial for sensitive data handled by IoT devices. In Smart Contracts, HE can be

integrated into smart contracts to perform operations on encrypted data, ensuring data privacy throughout the computation process [7].

Addressing aforementioned concerns and ensuring security and privacy for IoT products and services overall layers of the IoT architecture is a fundamental priority [8]. IoT devices and related services have to be secure enough so that users can trust them. Additionally, ensuring the safety of the IoT system is imperative to avoid unacceptable risks of injury or physical damage from its components. In this work, we concentrate on the security vulnerabilities of IoT across three critical layers: perception, network, and application layer. Figure 1 represents some of these layer wise security issues.

Levels examine the security issues of IoT across three layers: Perception layer threats, involve attacks on key components of IoT like WSNs and RFID systems. Network layer threats focus on vulnerabilities within communication protocols. Application layer threats, encompass attacks targeting IoT software and end-user devices.

Purposes estimate the impact of security attacks on IoT systems. Common purposes of IoT attacks include gaining unauthorized communication access, capturing or altering data, causing service disruptions, and draining device resources.

Security requirement addresses data, communication, and device security. Securing IoT communications involves implementing authentication, access control, and non-repudiation measures. To safeguard data, essential security requirements such as confidentiality, privacy, and integrity must be met. Additionally, trust and availability of IoT devices are crucial in various environments.

This article provides a blockchain-based IoT solution using efficient embedded homomorphic encryption, exploiting a linear asymmetric and additive homomorphic function. We embed data so that multiple data values can be encrypted in one value. This principle may provide many advantages, e.g., an attacker should possess all plain data values of the encrypted message to launch a brute force attack. Another advantage of this technique is that the values are encrypted in the stack, where the first encrypted value cannot be decrypted without decrypting all other values.

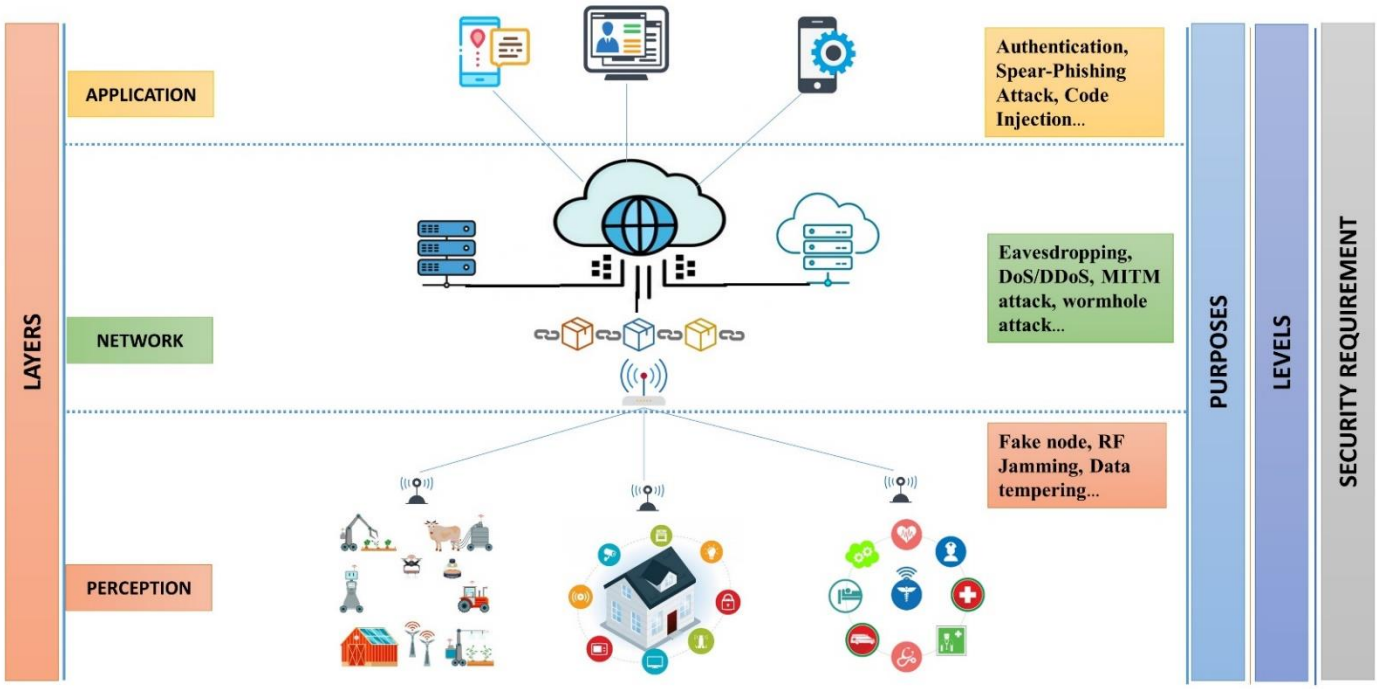

Figure 1. IoT layer wise security issues

2. RELATED WORK

Blockchain is the technology that provides the security and credibility terms for a decentralized system. Cryptocurrency, e.g., Bitcoin, which creates a digital asset transaction market, is one of the early applications of blockchain. Nowadays, blockchain is integrated with many technologies thanks to smart contracts, which are digital contracts triggered by some predefined circumstances. Ethereum was among the first to support their conception using smart contracts and transactions in the blockchain. The current blockchain applications are developed to find more and improve the usability of blockchain. This section discusses and analyzes the recently proposed technology in IoT blockchain solutions.

Zhou et al. [9] presented a novel blockchain based on the threshold IoT service system Bee-Keeper, whose main goal is to provide a secret sharing and secure multi-party computing protocol in IoT environments. The servers blindly perform homomorphic computations on a user's data; the server leader has the ability to reconstruct the data by gathering a threshold number of accurate responses. Moreover, the malicious nodes can be evaluated and examined. As a result, the obtained prototype is constructed for the Ethereum blockchain platform with four nodes, and the server responds to requests with 107 milliseconds. However, this approach exhibits a lightweight computational complexity. BeeKeeper depends on the performance of the underlying blockchain platform. Since BeeKeeper operates within the constraints of the Ethereum

blockchain, which imposes a maximum limit of 1014 bytes, the efficiency of this approach is inherently bounded by the capabilities of the blockchain platform. BeeKeeper 2.0 [10], an enhancement of the study by Zhou et al. [9], is among the first blockchain-enabled IoT systems to use fully homomorphic computations. In this system, the servers cannot receive any plaintext data; if a server behaves dishonestly, it will be discovered by the validators. The outcomes of the experiments carried out by the authors indicate the effectiveness of the proposed system using resource-constrained devices. They also prove that the average access delay to the blockchain is proportional to the rate of shared transactions. Furthermore, the access delay is highly affected by low throughput. Blockchain with homomorphic encryption is also addressed by Loukil et al. [11], in which a Privacy-preserving IoT Data Aggregation (PrivDA) technique is proposed. PrivDA ensures confidentiality, data integrity, and the sender's identity verification, which are security properties [12]. Also, it guarantees two privacy properties, namely anonymity and pseudonymity. The authors of PrivDA employed the Paillier homomorphic encryption scheme to allow IoT data aggregators to process encrypted data without disclosing the raw data produced by smart devices. The Ethereum blockchain stores data, ensures tamperproof communication, and controls data aggregation using a smart contract. In PrivDA, the network is split into two areas, namely regional area and blockchain networks. The first area contains several groups, each consisting of one data aggregator and several smart devices. Each group is connected to the blockchain network. The latter may contain smart devices, aggregators, consumers, and a key generation node. When a consumer wants to receive aggregated data from any group, it creates a smart contract. The smart contract places the potential data producers that can respond to the consumer into one group. Then, it chooses one aggregator to compute the group-requested result using homomorphic computations. The challenge with the Paillier cryptosystem is that it needs a larger storage space because it uses modulo n^2.

The link between blockchain and homomorphic encryption is not limited to IoT. In an attempt to combine blockchain and homomorphic encryption, Qu et al. [13] improved the voting efficiency through a blockchain-based protocol that avoids interacting with a non-trusted third party using a smart contract. The authors exploited the homomorphic encryption technique and the homomorphic signature encryption algorithm to encrypt and sign the ballot. Each registered voter can be regarded as a peer in the blockchain network and can vote on the identity of his peers in the blockchain. The number of voters does not change during the voting process. The smart contract is responsible for ensuring the verifiability and non-repeatability of the protocol during the verification phase and preventing a voter from voting twice, i.e., it finds the corresponding status of the voters as 'voted' and immediately discards the duplicate votes. However, the authors did not mathematically prove the method of calculating the number of votes.

To resolve the security problems in data sharing and model sharing, Jia et al. [14] have developed an application model of blockchain-enabled Federated Learning to produce a data protection aggregation scheme, aiming to be used in an Industrial IoT (IIoT) scenario. They proposed three models: distributed K-means clustering with differential privacy and homomorphic encryption, distributed Random Forest with differential privacy, and distributed AdaBoost with homomorphic encryption. Integrating these models with blockchain and Federated Learning gives multiple data protection in complex IIoT environments. This solution is, therefore, used in a specific environment. Singh et al. [15] proposed a privacy-preserving data aggregation model for smart grids based on deep learning and homomorphic encryption. The data aggregation process has a multi-tiered architecture and is recorded in the cloud using a blockchain; it is also shown to be more effective in detecting smart meter manipulation while having low computational overhead. The authors used a new symmetric homomorphic encryption scheme and applied the SHA-256 hash function. Similar to Singh et al., Yan et al. [16] distributed privacy protection architecture based on blockchain and leveraging fully-homomorphic Pailler cryptosystem [17] on edge computing. The task execution side encrypts all data, while the edge node processes the data's received ciphertext and returns the final result's ciphertext to the client. Through experiment results, the authors verified that their approach achieves security protection and integrity check of cloud data and realizes more extensive secure multiparty computation. However, this work does not clarify why it needs additive homomorphic encryption.

In this article, we attempt to address some of the gaps in the aforementioned literature. We present an architecture for healthcare systems that integrates IoT devices and blockchain technology with embedded homomorphic encryption. The designed architecture is applied to a healthcare data system as a use case. The embedded homomorphic encryption makes it possible to obtain homomorphic addition over the encrypted data. Using this technique, the administrator can gain a different level of access to the encrypted data. In terms of security level and to ensure data stability, traceability, and anonymity, we use blockchain to store digital health information.

2.1 Contributions and innovations

Healthcare systems impose stringent requirements for data security, privacy, and accessibility, challenges that our architecture effectively addresses. Our research integrates IoT devices, blockchain technology, and embedded homomorphic encryption to enhance computation speed and security levels within IoT systems. Specifically, our proposed ultralight linear encryption technique accelerates computations, enabling homomorphic addition over encrypted data. This technique also provides customized access controls, ensuring that each individual gains specific access only to essential encrypted data, thereby minimizing exposure of sensitive information.

2.2 Differences from existing studies

Enhanced computation speed: Unlike traditional homomorphic encryption methods, our ultralight linear encryption technique significantly improves computation speed while maintaining robust security. This capability is crucial for processing sensitive patient information in real-time without compromising privacy. Existing studies often focus on the theoretical aspects of homomorphic encryption but do not address the practical challenges of implementing it in resource-constrained IoT devices.

Customized access control: Our solution enables specific access to encrypted data by different individuals. Patient datasets are exclusively accessible to healthcare professionals

directly involved in their care. We enforce strict, least-privileged controls over access to IoT data, ensuring that individuals can only retrieve essential information necessary for their roles, thereby minimizing unnecessary exposure. Existing studies often focus on general access control mechanisms that do not provide this level of granularity.

Integration with blockchain and fog computing: Our architecture leverages blockchain technology to provide security, stability, traceability, and anonymity. This integration ensures that all data transactions are recorded immutably, enhancing accountability and transparency. In addition, our model enables efficient data processing at the fog or cloud level, leveraging the increased computational resources available at these higher levels. Existing studies often focus on the use of blockchain for data storage but do not explore its potential for enhancing the security and integrity of data processing.

To summarize, our architecture offers a unique combination of high computation speed, specific access control, and robust security through blockchain integration. By addressing the practical challenges of implementing homomorphic encryption in IoT systems, our work provides a significant innovation in the field of IoT security and data processing. This response clarifies the unique contributions and innovations of our paper, highlighting the differences from existing studies and the usefulness of our suggested architecture.

3. BLOCKCHAIN AND HOMOMORPHIC ENCRYPTION INTEGRATION APPROACH

In healthcare systems, the patient's information and medical history are collected whenever a patient visits a doctor, hospital, or pharmacy. Only the patient and the healthcare professionals directly involved in his care and follow-up should have access to his complete profile.

Certain medical information in a patient's file may also be useful for other purposes beyond individual follow-up, such as improving healthcare and public services. The patient must be able to know who is allowed access and how his data is protected to respect his privacy. In this section, we present the details of an approach that integrates blockchain and homomorphic encryption to protect patients' data effectively.

3.1 Proposed approach specifications

IoT is an emerging paradigm recognized as a revolutionary technology of this century. It allows devices to communicate with one another seamlessly, providing services without the need for human intervention [18]. The main goal of IoT is to improve human life by leveraging its intelligent and smart functionalities. IoT devices collect and exchange data over the network, which increases the attack vector. Hence, it is essential to implement mechanisms to preserve user privacy in IoT systems. Cryptography is a widely used technique for safeguarding data transmitted over wireless channels [19]. It encompasses both encryption and decryption processes, which can be categorized into two primary types: symmetric and asymmetric methods. Symmetric techniques utilize a single key for both encryption and decryption, whereas asymmetric methods involve a pair of keys: a public key for encryption and a private key for decryption. Traditional cryptographic methods are often impractical for IoT devices with limited resources, as they require substantial processing power and memory capacity. As a result, attaining robust security using streamlined methods presents a challenging task [20]. Lightweight cryptography has recently gained significant attention as a means to optimize traditional cryptographic algorithms and provide security solutions tailored for resource-constrained devices [21, 22].

To accomplish this, we employed an asymmetric lightweight scheme to showcase privacy preservation in our healthcare scenario.

The used encryption technique consists of three processes:

Keys generation KeyGen(): that returns a set of secret keys Sk_i and public keys Pk_i, $(Sk_i, Pk_i) = (k_i, k_i + \alpha \times p)$

Encryption Enc(m): that encrypts a plaintext m using the public keys Pk_i where:

$$c = (m_1 \times pk_1 + m_2 \times pk_2 + \ldots + m_i \times pk_i)\ mod\ n$$

To encrypt a message, the sender decomposes it into several parts so $m=m_1\ O\ m_2\ O\ m_3\ O.\ m_i$ where O is an operation, for example, Addition, Multiplication, Concatenation, &, etc. The encryption consists of multiplying each part mi by a public key pk_i. If we choose "O" to be "&", this technique will be very flexible, because in this case, parts have no relationship between them, so m_i is independent of m_j. In our model, each mi will be a piece of information for the patient.

Decryption Dec(c): decrypts a ciphered text c using the secret keys $Sk_i, m_i = \frac{c}{sk_i}, i = l, \ldots 0$, with $c \leftarrow c - c \times sk_i$ after each computed m_i; where, $\left(\frac{c}{sk_i}\right)$ is the quotient of $c \div sk_i$ and l denotes the length of c. To ensure decryption and getting a valid plaintext, Eq. (1) must be verified.

$$p > \sum_{j=0}^{i} m_j \times k_j \quad (1)$$

Algorithm 1 shows the *KeyGen* function. This function returns i secret keys where i is the number of fields to be encrypted; for each secret key, the function also returns the corresponding public key, which is easily computed using the private key p.

Algorithm 1: Keys Generation

(1): **Require Private**: p, q two large prime numbers, α: large random number, secret keys k_i

(2): **Require Public**: $i = len(m), M = max\ (m)$

(3): **Ensure**: $\alpha > q, p > M \times \sum_{j=0}^{i} k_j$

(4): $Sk_i \leftarrow k_i$

(5): $Pk_i \leftarrow k_i + \alpha \times p$

(6): $n \leftarrow p \times q$

(7): **Return** $(Sk_i, (Pk_i, n))$

After the key generation process, the devices use the Pk_i to encrypt their data using Algorithm 2.

Algorithm 2: Encryption

(1): **Require Public**: n, Pk_i

(2): **Ensure**: c

(3): $c \leftarrow (m_1 \times pk_1 + m_2 \times pk_2 + \ldots + m_i \times pk_i)\ mod\ n$.

(4): **Return** c

The final user, patient, or administrator (e.g., doctor) can use the data after decrypting it according to Algorithm 3.

Algorithm 3: Decryption

(1): **Required Private**: p, Sk_i

(2): **Ensure**: $m_1, m_2, \ldots, m_i$

(3): $i \leftarrow l$, where l is the length of c

(4): $m_i = \frac{c}{sk_i}$

(5): $c \leftarrow c - c \times sk_i, i \leftarrow i - 1$

(6): repeat 4 and 5 until $i = 0$

(7): $m = \sum_{j=0}^{i} m_j$; (effectively, m=(m_1, m_2, … m_i))

(8): **Return** m

The computational complexity of the proposed system equals *O(i)* where *i* is the number of fields or information (Algorithms 1 and 2).

3.2 Proposed model design

In our proposed model design, IoMT (Internet of Medical Things) devices (Level 1 in Figure 2) receive patient information in fields such as name, surname, information relating to their state of health, genetic data, and biometric data (which are physical characteristics that are measurable and machine verifiable). To encrypt this data, the device uses an encryption scheme that multiplies each field value by a public key, then gathers the obtained results in a single record (transaction) by calculating the sum of these encrypted values, forming embedded homomorphic encryption.

$$c \leftarrow (m_1 \times pk_1 + m_2 \times pk_2 + \ldots + m_i \times pk_i) mod\ n \quad (2)$$

To harness the computational capabilities, storage resources, intelligence, and processing power of devices, we employ Fog Computing to facilitate seamless data collection through device-generated transactions (Level 2 in Figure 2). The fog generates the blocks and participates in a consensus to integrate them into the blockchain. Using fogs within the local network enables real-time data processing, provides ample storage space, and optimizes data performance by distributing workloads.

After a consensus iteration, a new block is added to the blockchain (Level 3 in Figure 2); this block contains the sensitive patient information, and the blockchain version is found at the fog level or at the cloud level (for more security). In these levels, homomorphic operations (such as addition) are performed to guarantee patient privacy.

The end-user is at the highest level of the presented model (Level 4 in Figure 2). An end-user can be a patient himself, a doctor, a healthcare provider, a commissioner (who assesses how care is provided), an academic researcher (to understand the sources of disease better or develop new remedies), or a charity (to evaluate services and identify ways to improve care). The entire patients' dataset will only be viewed by healthcare professionals directly involved in their care. Strict, least-privileged controls are implemented to govern access to IoT data, ensuring that individuals can only access the essential information required, minimizing unnecessary exposure. Our approach guarantees this by attributing a specific set of secret keys to everyone; whoever has more keys can read more embedded fields, using Eq. (3).

$$m_j \leftarrow \frac{c}{sk_j}\ with\ c \leftarrow c - c \times sk_j \quad (3)$$

3.3 Safe arrangement of fields

The encrypted fields (embedded fields) order is not random in the proposed model. While combined to create a record, the first information, encrypted by the first key, cannot be interchanged with the second, and vice versa.

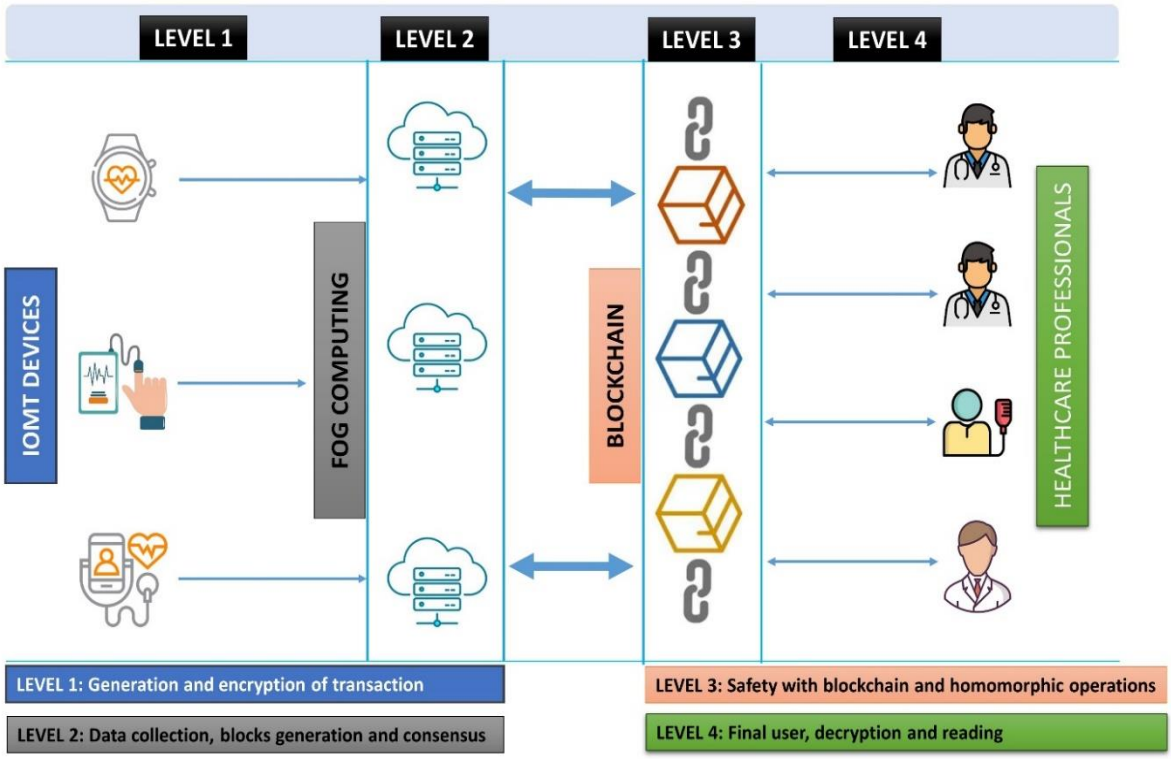

Figure 2. The proposed architecture illustration in the healthcare sector

In Eq. (3), it is evident that possession of the key sk_j allows access to the most recent information m_j. Likewise, having both keys sk_j and sk_{j-1} grants access to both the latest information m_j and the preceding information m_{j-1}, and so forth. We arrange the fields in an incremental fashion. For example:

$$c = m_1 \times pk_1 + m_2 \times pk_2 + m_3 \times pk_3$$

A user who captures information m_2 must be authorized to access information m_3 because m_2 can only be decrypted by possessing the keys sk_2 and sk_3. The owner of key sk_3 will only be able to access the information m_3. As for a user who needs to access information m_1 must have the keys sk_1, sk_2, and sk_3; anyone who is qualified should have access to all the information. Table 1 summarizes the information encoding hierarchy based on the following formulate.

$$record = field_1 \times pk_1 + field_2 \times pk_2 + \cdots + field_j \times pk_j$$

Table 1. Field record levels

User Level	Retrieved Fields	Required Keys
l_j	f_j	sk_j
l_{j-1}	f_j, f_{j-1}	sk_j, sk_{j-1}
l_{j-2}	f_j, f_{j-1}, f_{j-2}	sk_j, sk_{j-1}, sk_{j-2}
…	…	…
l_1	$f_j, \dots, f_1$	$sk_j, \dots, sk_1$

3.4 Additive technique

The proposed embedded homomorphic encryption verifies the property of homomorphic addition, which we express in the following equation:

$$Enc(m) \oplus Enc(m') = Enc(m + m') \quad (4)$$

This homomorphic property is used in the healthcare sector [11] because it provides statistics on the patient's condition while respecting his privacy. We will subsequently prove that the proposed scheme satisfies the homomorphic addition.

$$Enc(record) = m_1 \times pk_1 + m_2 \times pk_2 + \cdots + m_j \times pk_j$$
$$Enc(record') = m'_1 \times pk_1 + m'_2 \times pk_2 + \cdots + m'_j \times pk_j$$
$$Enc(record) + Enc(record') = m_1 \times pk_1 + m_2 \times pk_2 + \cdots + m_j \times pk_j + m'_1 \times pk_1 + m'_2 \times pk_2 + \cdots + m'_j \times pk_j = (m_1 + m'_1) \times pk_1 + (m_2 + m'_2) \times pk_2 + \cdots + (m_j + m'_j) \times pk_j = Enc(record + record')$$

Using our technique, a user at a certain level can ask the cloud server to perform patient data operations without decrypting it. For example, a user of level *j* wants to calculate the sum of t samples $\sum_{i=1}^{t} m_{j_i}$,where m_j are blood glucose levels. The server calculates $s_1 = \sum_{i=1}^{t} record_i$. Using private key sk_j, the user decrypts *s* as follows:

$$s_2 = s_1 - (s_1 mod\ sk_j) = \sum_{i=1}^{t} m_{j_i} \times pk_j = pk_j \times \sum_{i=1}^{t} m_{j_i}$$

Therefore, $s_2 mod\ (sk_j - 1) = \sum_{i=1}^{t} m_{j_i}$

4. PERFORMANCE EVALUATION

Experimentation was conducted using an HP Laptop with the following specifications: Processor Intel(R) Core (TM) i3-3110M CPU @ 2.40GHz, 2 Core(s), 4 Logical Processor(s), 4 Go RAM. All the experiments were performed using Python programming language. We have employed encryption keys with sizes of both 128 bits and 256 bits. Table 2 summarizes our evaluation results. The proposed technique simulates a data compression process and therefore must be studied in several models in order to show its effectiveness compared to other techniques. There are two main modes: changing the key size and changing the number of merged or compressed fields.

To conduct our experiments, we created a random dataset simulating a real database of records, each one consisting of five fields: name, ID, age, gender, and blood glucose level. We do not need correct values for each of these fields because we are interested in the size to get computation cost and the number of fields to get communication cost.

Table 2. Comparative study as a function of running time (ms)

Scheme	Enc_{128}	Dec_{128}	Enc_{256}	Dec_{256}
[23]	0.14	3.4	93.9	279
[24]	74.5	0.4	403	1.6
[25]	2.3	5.8	15.7	37.3
Ours	0.13	0.79	0.19	0.87

4.1 Computation cost

In the experiment presented in Table 2, we performed two tests, the first with a key of length 128 bits and the second with a key of length 256 bits. Each test is performed ten times; these results are the average of the obtained values.

A record consists of five fields: name, ID, age, gender, and blood glucose level. In some studies [23-25], each field is coded separately c_i=Enc(m_i). The output consists of five encrypted fields (c_1, c_2, c_3, c_4, and c_5). Therefore, the total encoding time is the sum of the time required to encrypt each field.

$$T = \sum_{j=1}^{i} t_j \quad (5)$$

where t_j denotes the time required to encrypt a piece of information, and *i* is the amount of information.

Our technique takes less time because the fields are coded together (see Eq. (2)) so *i=1* in Eq. (5) and the output consists of only one field. Despite this, the execution time is also linked to the details of the techniques. By that, we mean the elemental operations represented in each technique separately, such as addition, multiplication, exponentiation, etc. While encoding each field requires multiplication and addition in the proposal, needs multiplication and two exponentiation operations [24].

The decryption process of our scheme is slightly slower compared to the encryption process because all the fields are interleaved in a single record, and we perform six operations (four modulo operations and two subtraction operations) to extract information (see Eq. (3)); therefore, thirty total operations to recover the five fields. On the contrary, Ren et al. [24] required two exponentiation operations, two subtraction operations, and four modulo operations. As for the other study [23], it was the slowest in terms of decoding speed, and this is

due to the large number of operations required to extract the information; there are two exponentiation operations, two multiplications, divisions, additions, and subtractions for each iteration, knowing that there are a number of iterations equals to *length of ciphertext*. In the study by Rivest et al. [25], there is an exponentiation operation in both the encryption and the decryption operations, but $time_{Dec}>time_{Enc}$ because *length(ciphertext)>length(plaintext)*.

Figure 3 shows the percentage increase in the encryption time when the key or message size increases [23-25]. Scheme 3 achieved the largest increase in execution time because it contains more exponential operations than other schemes, as the exponential operation is characterized by a high execution time compared to other operations such as addition and multiplication.

The scheme proposed by Kara et al. [23] is characterized by a significant increase in ciphertext size because it depends on the message length. Figure 4 shows the proposed technique exhibits a slight increase in time when the message size increases. This is because the multiplication operation is not much affected by the increase in size compared to the exponential operation.

Table 3 shows the size of the records encrypted by the patient's device (embedded encryption). Our proposed model demonstrates significant scalability improvements, as evidenced by the reduced record size of just 1kb compared to other methods.

$$S = \sum_{j=1}^{i} s_j \tag{6}$$

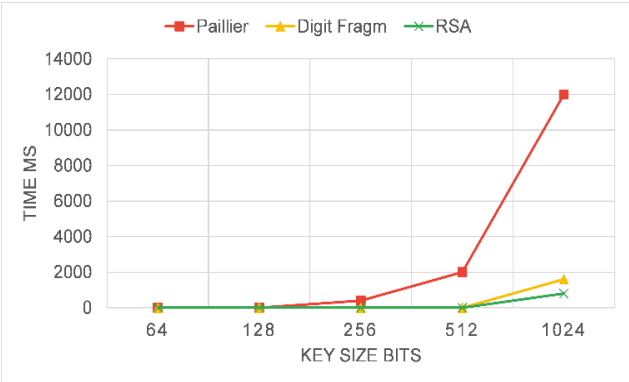

Figure 3. Encryption time of Pailler [24], Digit Fragm [23], and RSA [25]

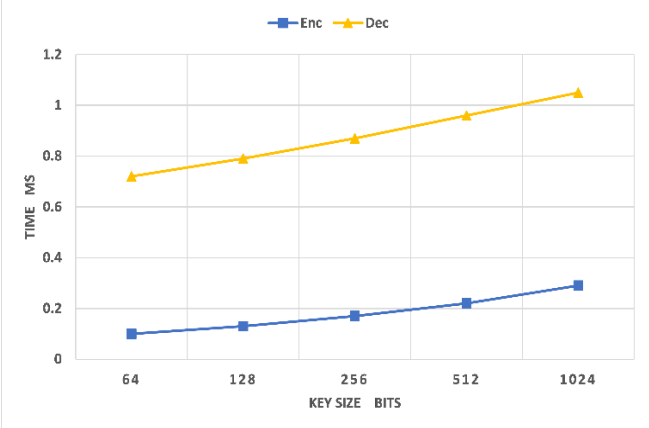

Figure 4. Encryption time of the proposed model

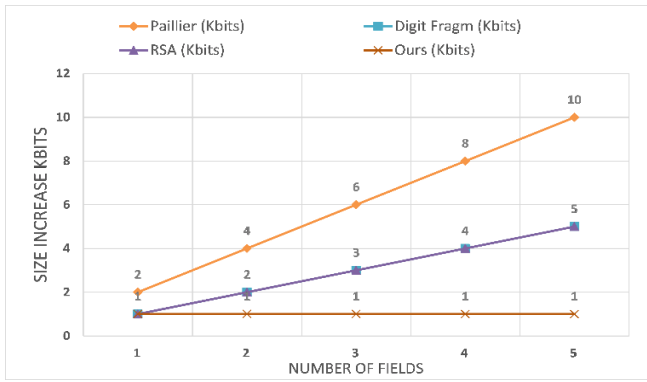

Figure 5. Field size increase

Eq. (6) summarizes how to calculate the size resulting from the encryption process, which is represented in the original text, s_j denotes the size of a single output. The output is usually in order of the modulus n, and therefore when encrypting each piece of information separately for example Name or ID. In other words, the total size will be the number of information or fields. In the proposed technique $i=1$ which gives $S=1kb$ vs. $i=5$ and $S=5kb$.

Because there are i information fields ($i > 2$), the size of a record is related to the public key. In some studies [23-25], data will be presented in i independent fields, so the size in the study by Kara et al. [23, 25] will be i kb (kilobit) if the size of the public key is 1 kb. In the study of Ren et al. [24], the size is doubled because the authors used the Pailler public key encryption technique [17]. On the contrary, our proposed model combines the encoding of the i fields in a single record using natural addition, which increases the scalability of the system and the rigidity of the technique and decreases the storage size; this appears when the number of information fields is larger, as shown in Figure 5.

Table 3. Field record size

	[23]	[24]	[25]	Ours
Record size	5kb	10kb	5kb	1kb

4.2 Scalability

Scalability is a critical consideration when applying blockchain and homomorphic encryption in the domain of IoT and connected objects. Both technologies offer unique advantages and challenges in handling large volumes of users and devices.

As the number of transactions and participants increases, traditional blockchains encounter delays and increased costs per transaction. This inefficiency poses a significant barrier to deploying blockchain in IoT applications where real-time data processing and responsiveness are crucial. To address blockchain's scalability limitations, we can introduce one of the proposed approaches such as sharding [26, 27], sidechains [28], and off-chain [29]. Sharding divides the blockchain network into smaller partitions, allowing for parallel transaction processing and reducing congestion. Sidechains enable specific transactions to be processed separately from the main chain, alleviating network congestion while maintaining interoperability. Off-chain solutions involve conducting transactions outside the main blockchain network, thereby reducing the burden on the main chain and improving overall scalability.

Encryption systems can be computationally intensive, potentially limiting their scalability when applied to large-scale IoT deployments with numerous devices generating continuous streams of data. To enhance scalability, we used linear encryption that does not require much computational complexity. By embedding many pieces of information in only one record, the sizes of the outputs decrease very significantly (s_{ours}=s/i), especially when the number of merged fields *i* increases, and thus the blockchain's size is less, which opens the possibility for a larger number of participants to join the system, i.e. greater scalability.

5. SYSTEM ANALYSIS AND EFFICIENCY

Our proposed model incorporates several mechanisms to ensure data privacy and confidentiality during data sharing and transmission. Here are the key aspects:

Homomorphic Encryption: We employ a homomorphic encryption technique to ensure that data is encrypted at all times, including during transmission and computation. This means that patient data remains encrypted throughout its lifecycle, maintaining privacy even when it is being processed. Only authorized personnel can access the data, and even then, only for the specific purposes for which they are authorized. This ensures that data privacy and confidentiality are maintained during data sharing and transmission.

Hierarchical Access Control: Data is encrypted with multiple keys that are combined to generate a transaction. Only users with the appropriate number of private keys can access the specific data records, ensuring that only authorized personnel can view sensitive information.

Blockchain Technology: Integrating blockchain provides decentralization, transparency, and anonymity. Each transaction is securely recorded on the blockchain, ensuring that data sharing is traceable and tamper-proof. The use of the CW-PoW algorithm further enhances security while preserving energy.

Fog Computing: To further enhance data privacy and confidentiality, we utilize fog computing. By processing data closer to its source, fog computing reduces latency and minimizes the risk of data breaches during transmission. This additional layer of security ensures that data sharing and transmission are secure and efficient.

In fact, preserving privacy in IoT systems, especially during the data aggregation process, is still a challenging task. Usually, aggregated data in IoT is collected, stored, and processed using a centralized server. Such an approach may be practical solely under trusted servers. Unfortunately, centralized structures suffer from issues such as the single point of trust problem. The collected data may be deleted or modified by an untrusted server. Some distributed solutions have been suggested to cope with IoT data aggregation issues encountered in centralized schemes, in which data aggregator nodes are selected to collect the received data from the collaborating users. However, as the received data is encrypted, aggregators must decrypt it to aggregate them, which may lead to data disclosure. In Section 3, we have introduced a blockchain-based IoT solution with the help of a homomorphic encryption scheme.

5.1 Blockchain limitations to consider

The key motivation to introduce blockchain technology in mission-critical IoT systems, e.g., healthcare, is its decentralized property. Decentralization allows our model to operate without a trusted third party that controls and manages the patient data, thus solving the single point of trust problem. Another significant blockchain feature that we leveraged in our approach is transparency, which provides users with visibility into the legitimacy of every transaction within our system, allowing for collective verification by all users. Additionally, blockchain's security helps guarantee that all healthcare transactions are recorded, organized, and maintained in immutable and secure blocks. The blockchain makes sure that once a transaction is successfully committed, it cannot be altered. This ensures patient data integrity within each block, preventing tampering. Finally, blockchain's anonymity feature may hide patients' identities, a crucial privacy feature for any healthcare system. For these reasons, among others, blockchain technology may emerge as a prudent choice for designing privacy-preserving patient data systems in the healthcare sector.

The mining process, i.e., block creation and validation, must be considered in IoT environments. Typically, in the context of IoT, various challenges are often encountered, including constraints such as limited power and memory, issues related to scalability and complexity, as well as latency overheads. Some studies in the literature address blockchain implementations, the most popular of which are Proof of Work (PoW) [30] and Proof of Stake (PoS) [31]. Nevertheless, the majority of existing implementations are ill-suited for direct application within the IoT context. For instance, PoW demands immense computational resources, while PoS requires both memory and computational resources to achieve consensus. Also, in both schemes, the blocks are broadcasted and verified by each node in the network. This leads to scalability issues because most IoT devices have limited bandwidth and processing power. In addition, the mining process usually suffers from latency issues (e.g., in Bitcoin, a wait of up to 30 minutes to confirm a transaction is needed), while most IoT, especially healthcare applications, have stringent delay requirements. On the other hand, due to the limited throughput and the total number of committed transactions per second that can be stored within a block, current blockchain implementations are unsuitable for an IoT environment as the number of interactions between IoT devices may exceed such limits. Hence, it is imperative to select an IoT-compatible consensus scheme that effectively tackles the aforementioned issues. Furthermore, an ill-advised choice of consensus algorithm could render the entire system inoperable.

5.2 Selected consensus algorithm

In the literature, there are many efficient proposed algorithms. PoW is a powerful consensus mechanism used in blockchain, and it is one of the most cited and referred by researchers. Nonetheless, it exhibits vulnerabilities, including high power consumption and susceptibility to the 51% attack. One approach to harness PoW's advantages is to utilize its derivative known as Compute and Wait PoW (CW-PoW) [32]. CW-PoW is an improved version of the PoW consensus algorithm, where the authors divided the operation of reaching an agreement into several rounds. The nodes initiate the first round by attempting to find a valid hash. Once a node discovers such a hash, it shares it and then waits to receive the remaining hashes, which are expected to be found by the other

nodes. The node that successfully finds the hash is granted the privilege to partake in the subsequent round and repeat the PoW process. Conversely, if the network has not yet accumulated the required number of hashes, the candidate node will remain in a waiting state. In the final round, the miner will be the first node to discover the hash. In CW-PoW, the proof of round *i* is determined by the following condition:

$$Hash(Block + ID_{Round_{i-1}} + Nonce) < Target \quad (7)$$

Initially, ID_{Round_i}=1 After that, ID_{Round} equals the sum of nonces found in the previous round. It is defined in the following equation:

$$ID_{Round_k} = \sum_{i=1}^{NbrS} (Nonce_i \; of Round_{k-1}) \quad (8)$$

where, *NbrS* is the number of solutions to find in each round.

Let NbrR be the number of rounds and NbrP the number of processes (nodes). Figure 6 represents the efficiency of CW in energy preservation.

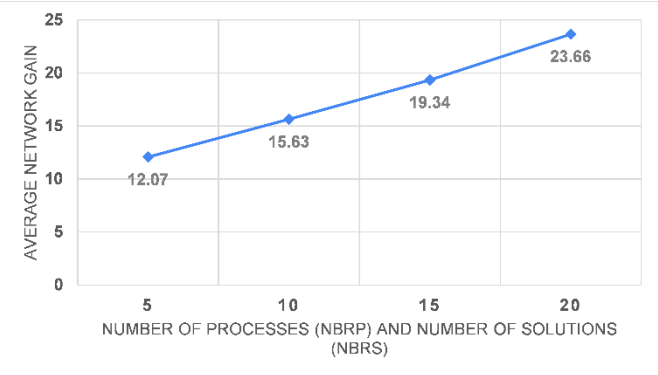

Figure 6. Compute and Wait PoW energy preserving example

Below, we provide a brief overview of known attacks that can be mitigated through the use of the CW-PoW consensus algorithm.

- 51% attack (the majority attack): In PoW, mining a block by a miner or a group of miners means having more than 51% computing power. Thus, if someone (attacker) has more than 50% of the computing power, it can control the underlying blockchain, i.e., the attacker can initiate a double spending attack, modify, reject, or reverse transactions, etc. By implementing CW-PoW in our model, the success rate of a 51% attack is significantly diminished, given that the likelihood of a candidate winning (dominating) is greatly reduced. Furthermore, the mining process in CWPoW is accomplished through multiple rounds, each requiring the introduction of ID_{Round} in the next proof of round. This approach can serve as a deterrent against double spending attacks, as it effectively slows down the process of creating an alternative branch.
- Distributed DOS (DDOS) attack: In DDOS, attackers try to impede or overload the network by generating useless traffic. An attacker may compromise and utilize certain individuals' IoT devices to target other devices, exploiting weaknesses or vulnerabilities within the underlying system. Fortunately, the CW-PoW consensus algorithm maintains the mining process even if some nodes leave the blockchain network.
- Sybil attack: Employing multiple pseudonyms can enhance privacy but also introduce the risk of Sybil attacks. A hostile node may exploit this anonymity to engage in illicit activities with the support of a majority. In CW-PoW, two interesting techniques are combined: multi-rounds and standard deviation. To emerge as the ultimate victor, false identities must construct a longer branch than the public one. Nevertheless, their progress is hindered by the multi-round technique, while the second technique further diminishes their prospects, significantly reducing the vulnerability to Sybil attacks.

5.3 Robustness of the encryption technique against attacks

In healthcare, security and privacy are critical factors, the system needs to use suitable and secure parameters. The proposed scheme relies on the hardness of the number's factorization problem, where attacker A must factorize the modulus N that equals p×q to get private keys p and q, and finally extract the secret key k. Therefore, the modulus N has to be large.

In addition, the technique used focuses on the difficulty of polynomial reconstruction problems. Hence, one of the pivotal factors for the effectiveness of cryptosystems lies in the length of the ciphertext.

Deterministic methods are vulnerable to Chosen Plaintext Attack (CPA). In CPA, the attacker has a ciphertext c and wants to find the original plaintext m. the attacker can choose m′ and get c′. Then, he computes Dec(c×c′) to obtain $m \times m'$ and consequently m.

The core of the proposed encryption is the $c = m \times k$ technique. This encryption scheme is vulnerable to certain attacks. Given (c, m), the attacker can easily obtain the secret key k, because $c \times m^{-1} = m \times m^{-1} \times k = k$. Now assuming that $c = c_1 + c_2 = m_1 \times k_1 + m_2 \times k_2$, where, $k_1 \neq k_2$. If the attacker has m_1 (or m_2), he multiplies c by m_1^{-1} (or m_2^{-1}) to get $k_1 + m_1^{-1} \times m_2 \times k_2$ or $m_1 \times m_2^{-1} \times k_1 + k_2$. This does not give the attacker sensitive information; he can not get either $k_1 or \; k_2$. When the attacker has m_1 and m_2, then $c \times m_1^{-1} \times m_2^{-1} = m_1 \times m_1^{-1} \times m_2^{-1} \times k_1 + m_2 \times m_2^{-1} \times m_1^{-1} \times k_2 = m_1^{-1} \times k_2 + m_2^{-1} \times k_1$. He must also possess a distinct set (m_1', m_2') to do the subtraction and obtain k_2 (or k_1).

In the context of the proposed cryptosystem, let's consider that we have *i* messages to encrypt within a record, where *i*>2.

To compromise the security, an attacker would need i^i plaintext messages along with their respective encryptions. This scenario is implausible, as the attacker doesn't have access to $m \times k$ plaintexts independently; instead, they are embedded within the record. To encrypt a record, we use i fields in one value. Therefore, A must have i records with their corresponding ciphertexts to obtain ki. Thus, the proposed system is secure against both known plaintext attacks (KPA) and chosen plaintext attacks (CPA).

In the man-in-the-middle attack, the attacker is positioned between users A and B to intercept their exchanged data. Initially, A selects a secret key k, creates a public key pk, and subsequently sends it to B. The attacker intercepts this key, chooses a private key k', and generates another public key pk', which is then forwarded to B, deceiving B into thinking the message is from A. Simultaneously, the attacker also sends the same key pk' to A, this time pretending to be B. The attacker has successfully established a shared session key with both A

and *B*. With these session keys in place, the attacker can intercept the data exchanged between *A* and *B*, decrypt it, manipulate it, and then re-encrypt and forward it. To mitigate this type of attack and prevent *B* from being misled by the attacker posing as *A*, A's public key can be verified by an independent certification authority. Additionally, in our cryptosystem, multiple keys are used to encrypt comprehensive information (records), adding an extra layer of complexity to thwart the attacker's objectives.

The probabilistic encryption approaches suffer from collision attacks. This type of attack can be defined by performing certain manipulations over ciphertexts to extract the whole or part of the initial plaintext. Therefore, the system will be broken when using a low entropy plaintext. To avoid collision attacks, some directives should be followed to reach a secure model. Besides increasing the number of fields *i*, the most important one is to use large secret numbers k_i which will increase the entropy.

In the Brute-Force Attack (BFA), the attacker must test all possible values from smallest to largest or opposite. In BFA on the public modulus *N*, the complexity for this attack to be successful equals *O(N)*. In BFA on the ciphertext, the complexity for this attack to be successful equals $O(\prod k_i)$ because he must try all possibilities of all keys.

5.4 Implementing the proposed system to healthcare

The use of homomorphic encryption in blockchain is in growth to include more industrial sectors. We just note the health privacy of functions such as critical disease prediction algorithms as well as the global state of the system. The most vital for this type of technology is statistical healthcare systems. This section illustrates a general architecture that shows how to use holomorphic blockchain-based encryption using anonymous statistical data of patients, for example, respiration or heart rate. We will use multi-levelled data of patients. The anonymous patient data m_1 is encrypted using the first key pk_1 and the introduced data by the Technician $m_2$will be encrypted using the second key pk_2. The doctor's data will be encrypted in the third level so that anyone who has the private key of the doctor can only read the doctor-embedded data. Furthermore, no one could read the anonymous patient data only after removing the doctor and technician-embedded data.

In fact, the blockchain homomorphic-based principle offers the possibility to provide private and traceable operations over multileveled encrypted data. In case the cloud needs such computation, the main entity (the doctor) should permit by release of his embedded data. and the cloud immediately performs the next level of processing which offers an advantage in the proposed scheme by using levelled public encryption. The cloud-unknown user can perform computation and cannot find out the real value of the stored data. On the other hand, the database owner (medical company) may extract only his data level stored in the cloud using its level key. The patient as well can read his data only after the doctor and the technician reveal their data.

6. CONCLUSION

In this article, we contributed a model for integrating blockchain and homomorphic encryption in IoT systems using healthcare as a use case. The proposed model is based on embedding data using natural addition, giving an encryption that verifies the property of homomorphic addition. This is a highly needed function in healthcare applications, where statistics on patient conditions are maintained without compromising their privacy. When information is accessed exclusively in a hierarchical manner, the same record is encrypted with multiple keys that are combined to generate a transaction. A user with a greater number of private keys can access this record (transaction). We improved the proposed model with blockchain technology to offer decentralization, transparency, and anonymity. To add a new block to the blockchain, we based it on the CW-PoW algorithm, which has proven efficient in energy preserving within IoT systems. By using five fields per record in our experiments, name, ID, age, gender, and blood glucose level, we achieved an encryption time equals 0,19ms and decryption time equals 0,98ms vs. 15,7 and 1,6 respectively in the best comparison technique. In communication cost, we achieved 1KB vs. 5KB.

Our experimental analysis highlights the effectiveness of our technique in reducing energy consumption. Future research could delve deeper into energy optimization strategies to ensure the sustainable and long-term operation of IoT systems. Additionally, we envision the potential adaptation and application of the proposed model in diverse contexts beyond healthcare, including areas like agriculture. IoT devices are increasingly being used in precision agriculture to monitor soil conditions, crop health, and environmental factors. By integrating these IoT devices with blockchain and homomorphic encryption, as in our healthcare use case, we can ensure the secure collection, storage, and processing of agricultural data. Adapting the proposed model to the agricultural domain will require addressing specific challenges and making necessary adaptations. For example, the types of data collected and the analytical requirements may differ from those in healthcare.

REFERENCES

[1] Vermesan, O., Friess, P. (2022). Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers. https://doi.org/10.1201/9781003338659

[2] Roman, R., Zhou, J., Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. Computer Networks, 57(10): 2266-2279. https://doi.org/10.1016/j.comnet.2012.12.018

[3] Wang, X., Zha, X., Ni, W., Liu, R.P., Guo, Y.J., Niu, X., Zheng, K. (2019). Survey on blockchain for internet of things. Computer Communications, 136: 10-29. https://doi.org/10.1016/j.comcom.2019.01.006

[4] Dai, H.N., Zheng, Z., Zhang, Y. (2019). Blockchain for Internet of Things: A survey. IEEE Internet of Things Journal, 6(5): 8076-8094. https://doi.org/10.1109/JIOT.2019.2920987

[5] Biswas, S., Sharif, K., Li, F., Nour, B., Wang, Y. (2018). A scalable blockchain framework for secure transactions in IoT. IEEE Internet of Things Journal, 6(3): 4650-4659. https://doi.org/10.1109/JIOT.2018.2874095

[6] Shrestha, R., Kim, S. (2019). Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities. In Advances in Computers. Elsevier, 115: 293-331. https://doi.org/10.1016/bs.adcom.2019.06.002

[7] Solomon, R., Weber, R., Almashaqbeh, G. (2023). Smartfhe: Privacy-preserving smart contracts from fully homomorphic encryption. In 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P), Delft, Netherlands, pp. 309-331. https://doi.org/10.1109/EuroSP57164.2023.00027

[8] Assiri, A., Almagwashi, H. (2018). IoT security and privacy issues. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, pp. 1-5. https://doi.org/10.1109/CAIS.2018.8442002

[9] Zhou, L., Wang, L., Sun, Y., Lv, P. (2018). Beekeeper: A blockchain-based IoT system with secure storage and homomorphic computation. IEEE Access, 6: 43472-43488. https://doi.org/10.1109/ACCESS.2018.2847632

[10] Zhou, L., Wang, L., Ai, T., Sun, Y. (2018). BeeKeeper 2.0: Confidential blockchain-enabled IoT system with fully homomorphic computation. Sensors, 18(11): 3785. https://doi.org/10.3390/s18113785

[11] Loukil, F., Ghedira-Guegan, C., Boukadi, K., Benharkat, A.N. (2021). Privacy-preserving IoT data aggregation based on blockchain and homomorphic encryption. Sensors, 21(7): 2452. https://doi.org/10.3390/s21072452

[12] Kara, M., Karampidis, K., Sayah, Z., Laouid, A., Papadourakis, G., Abid, M.N. (2023). A password-based mutual authentication protocol via zero-knowledge proof solution. In International Conference on Applied CyberSecurity. Cham: Springer Nature, Switzerland, pp. 31-40. https://doi.org/10.1007/978-3-031-40598-3_4

[13] Qu, W., Wu, L., Wang, W., Liu, Z., Wang, H. (2022). A electronic voting protocol based on blockchain and homomorphic signcryption. Concurrency and Computation: Practice and Experience, 34(16): e5817. https://doi.org/10.1002/cpe.5817

[14] Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K., Liang, Y. (2021). Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. IEEE Transactions on Industrial Informatics, 18(6): 4049-4058. https://doi.org/10.1109/TII.2021.3085960

[15] Singh, P., Masud, M., Hossain, M.S., Kaur, A. (2021). Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid. Computers & Electrical Engineering, 93: 107209. https://doi.org/10.1016/j.compeleceng.2021.107209

[16] Yan, X., Wu, Q., Sun, Y. (2020). A homomorphic encryption and privacy protection method based on blockchain and edge computing. Wireless Communications and Mobile Computing, 2020(1): 8832341. https://doi.org/10.1155/2020/8832341

[17] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 223-238. https://doi.org/10.1007/3-540-48910-X_16

[18] Whitmore, A., Agarwal, A., Da Xu, L. (2015). The internet of things-A survey of topics and trends. Information Systems Frontiers, 17: 261-274. https://doi.org/10.1007/s10796-014-9489-2

[19] Vaudenay, S. (2005). A classical introduction to cryptography: Applications for communications security. Springer Science & Business Media. https://doi.org/10.1007/b136373

[20] Hellaoui, H., Koudil, M., Bouabdallah, A. (2017). Energy-efficient mechanisms in security of the internet of things: A survey. Computer Networks, 127: 173-189. https://doi.org/10.1016/j.comnet.2017.08.006

[21] Kara, M., Laouid, A., Hammoudeh, M., AlShaikh, M., Bounceur, A. (2022). Proof of chance: A lightweight consensus algorithm for the internet of things. IEEE Transactions on Industrial Informatics, 18(11): 8336-8345. https://doi.org/10.1109/TII.2022.3168747

[22] McKay, K., Bassham, L., Sönmez Turan, M., Mouha, N. (2016). Report on lightweight cryptography. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8114

[23] Kara, M., Karampidis, K., Papadourakis, G., Laouid, A., AlShaikh, M. (2023). A probabilistic public-key encryption with ensuring data integrity in Cloud Computing. In 2023 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), Crete, Greece, pp. 59-66. https://doi.org/10.1109/ICCAIRO58903.2023.00017

[24] Ren, W., Tong, X., Du, J., Wang, N., Li, S.C., Min, G., Bashir, A.K. (2021). Privacy-preserving using homomorphic encryption in Mobile IoT systems. Computer Communications, 165: 105-111. https://doi.org/10.1016/j.comcom.2020.10.022

[25] Rivest, R.L., Shamir, A., Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2): 120-126. https://doi.org/10.1145/359340.359342

[26] Hashim, F., Shuaib, K., Zaki, N. (2022). Sharding for scalable blockchain networks. SN Computer Science, 4(1): 2. https://doi.org/10.1007/s42979-022-01435-z

[27] Yu, G., Wang, X., Yu, K., Ni, W., Zhang, J.A., Liu, R.P. (2020). Survey: Sharding in blockchains. IEEE Access, 8: 14155-14181. https://doi.org/10.1109/ACCESS.2020.2965147

[28] Vispute, A., Patel, S., Patil, Y., Wagh, S., Shirole, M. (2021). Scaling blockchain by autonomous sidechains. In Proceeding of Fifth International Conference on Microelectronics, Computing and Communication Systems: MCCS 2020. Springer Singapore, pp. 459-473. https://doi.org/10.1007/978-981-16-0275-7_38

[29] Jayabalan, J., Jeyanthi, N. (2022). Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy. Journal of Parallel and Distributed Computing, 164: 152-167. https://doi.org/10.1016/j.jpdc.2022.03.009

[30] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Satoshi Nakamoto.

[31] King, S., Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Self-Published Paper. https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf.

[32] Kara, M., Laouid, A., AlShaikh, M., Hammoudeh, M., Bounceur, A., Euler, R., Amamra, A., Laouid, B. (2021). A compute and wait in pow (CW-POW) consensus algorithm for preserving energy consumption. Applied Sciences, 11(15): 6750. https://doi.org/10.3390/app11156750