Hindawi
Security and Communication Networks
Volume 2022, Article ID 2793998, 10 pages
https://doi.org/10.1155/2022/2793998

Research Article

Somewhat Homomorphic Encryption: Ring Learning with Error Algorithm for Faster Encryption of IoT Sensor Signal-Based Edge Devices

V. Subramaniyaswamy,[1] V. Jagadeeswari,[1] V. Indragandhi,[2] Rutvij H. Jhaveri,[3] V. Vijayakumar,[4] Ketan Kotecha,[5] and Logesh Ravi[6]

[1]*School of Computing, SASTRA Deemed University, Thanjavur, India*
[2]*School of Electrical Engineering, Vellore Institute of Technology, Vellore, India*
[3]*School of Technology, Pandit Deendayal Energy University, Gandhinagar, Gujarat, India*
[4]*School of Computer Science and Engineering, University of New South Wales, Sydney, Australia*
[5]*Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune, India*
[6]*Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India*

Correspondence should be addressed to Rutvij H. Jhaveri; rutvij.jhaveri@sot.pdpu.ac.in

Received 16 December 2021; Revised 10 January 2022; Accepted 17 January 2022; Published 24 February 2022

Academic Editor: Mamoun Alazab

In recent years, Homomorphic Encryption (HE) has shown the possibility of securely running a computation arbitrarily without performing the data decryption. Many authors have shown Somewhat Homomorphic Encryption (SHE) or Fully Homomorphic Encryption (FHE) schemes implemented practically on both the addition and multiplication operations for SHE. The recent methods for implementing the FHE methods completely depend on arbitrarily reducing the time taken to perform the encrypted multiplication operation to increase the computation power required by SHE methods. This paper aims to accelerate the encryption primitives in an integer-based SHE based on the duration between each data transmission from the sensor and data packaging method. If the number of sensors increases exponentially in an edge device environment, the signals have to be encrypted faster in a packed mode in the edge environment and transferred to the cloud without a loss in data. The presented SHE method reduces the time taken for encryption based on the input number from the sensor and invariably increases the performance of the edge device. This advantage also helps the deploying healthcare application obtain end-to-end privacy in transmitting sensitive patient data.

1. Introduction

The ecosystem required for the Internet of Things (IoT) is growing exponentially because of the large-scale availability of low-cost sensors, actuators, microprocessors, and high-speed Internet infrastructure. These devices can be integrated seamlessly for gathering data from the required environment, depending on the application. The collected data have to be processed and monitored continuously for effective implementation. The healthcare industry is one of the largest revenue-generating sectors in India with a market share of 133 billion USD by 2022 [1]. The continuous monitoring of hospitalized patients and a timely prediction of complications that arise from disease leads to early recovery and reduces patient hospital stay. Apart from increased revenue, the reduced hospital stay of the patient will translate into less strain in the current healthcare infrastructure as well as saving the patient lives through the early detection of diseases.

With the established Internet framework, multiple sensor devices integrated with individual patients transmit their vital information through the network in the Edge

Computing (EC) [2–4] environment before processing critical information in cloud computing. In this current scenario, the transmitted data from the device will be transferred insecurely through wife networks and then to the centrally located servers. There is a possibility of altering the patient details and vital signs through hacking. They are transmitted through the nonsecured integrated IoT devices to cloud computing for further processing. In this scenario, the patient will be misdiagnosed, which also leads to further complications. This type of problem can be solved by adding a layer of security before transmitting the patient's vital signs into the cloud server for further processing and timely prediction. The time is taken to encrypt the patient data, send the encrypted data, process the encrypted data, and transmit urgent messages in case of emergency to the healthcare provider as early as possible. Another scenario is the number of patients increased by 100-fold from the average number of available beds at any particular time also leads to network congestion or bottleneck in transmitting the data through IoT devices. This paper explains a new method using Fully Homomorphic Encryption with reduced encryption time before transmitting data when compared to the MORE or PORE method. Also, this new method helps in the elimination of bottleneck problems. Even with the increase in data, the time required for processing in the Edge Computing environment is less before sending the data to the cloud environment on a priority basis. The computation method increased the performance in the edge environment by carrying a data prediction in Edge Computing and storing and analyzing data in the cloud [5–8].

This paper focuses on the security issues required while transmitting data from edge devices into a cloud server. We specifically concentrate on preserving the privacy of patients' information and enhancing the security of transmitted data by encrypting them with a modified lightweight algorithm based on Somewhat Homomorphic Encryption-Ring Learning with Error. This presents a new way to securely encrypt IoT sensor signal value based on the frequency of transmitting the signal to the edge environment. Depending on the encryption function selected, the edge device will require less computing space and time to encrypt and transmit the data to the cloud server. Table 1 presents the list of abbreviations used in this paper to help readers to have a better understanding.

Table 1: List of abbreviations used.

SHE	Somewhat Homomorphic Encryption
FHE	Fully Homomorphic Encryption
HE	Homomorphic Encryption
IoT	Internet of Things
USD	United States Dollar
EC	Edge Computing
WHO	World Health Organization
HBP	High Blood Pressure
MORE	Matrix Operation and Randomization Encryption
TTP	Trusted Third Party
EHR	Electronic Health Record
ECG	Electrocardiogram
SHA	Secure Hash Algorithm
RLwE	Ring Learning with Errors

2. Motivation

The healthcare system delivers a quality of service to the people whenever and wherever they need to increase the quality of life and decrease the mortality rate. Healthcare in public life is mainly due to the advanced need to stay healthy and contribute to the economy for a long period. The health and lifestyle do not match at all period.

Chronic diseases are noncommunicable diseases that increase due to changes in the lifestyle of people. For instance, dramatic changes in food quality, mainly an increase in fast food, excess weight gain, work stress, alcohol, and so on, are the main reasons for chronic diseases. As per the World Health Organization (WHO) report, it increases the mortality rate. Hypertension [9] is one of the chronic conditions in which blood pressure increases in the arteries also called High Blood Pressure (HBP). HBP does not usually cause symptoms, but it leads to a major risk influence in heart failure, kidney failure, heart attack, artery disease, and so on. The systolic and diastolic are the two measurements expressed to measure blood pressure. The systolic pressures are within the range of 100–130 mmHg and for diastolic 60–80 mmHg. The BP is at or above 130/80, or 140/90 mmHg is diagnosed to be hypertension. It is further classified into primary and secondary hypertension. The HBP is classified into gestational hypertension, preexisting hypertension, and preeclampsia during pregnancy. It is essential to monitor pregnant women continuously because hypertension causes globally 16% of maternal death approximately.

The secure send and storing of personal data without being compromised play a major role in protecting the privacy of the patient [10–12]. The patient's health records or diagnosis stored over the cloud-based server need to be accessed by any physician at any time without being compromised or transferred securely.

In general, Figure 1 depicts the IoT-based fog computing healthcare monitoring system with security consisting of three steps. First, the patient's health data, especially blood pressure data, are collected by placing IoT smart matches. The collected data are sent to fog computing through wireless networks such as Bluetooth, Wi-Fi, and WiMAX. Second, fog is an extension of the cloud for the analysis of data where we use mobile in the place of fog to perform the security operation. The Homomorphic Encryption technique is carried out in fog computing. Third, the secured data are stored in the cloud subsystem.

If a patient is getting treatment from multiple hospitals, the patient's records are converted into Electronic Health Record (EHR) to be used anywhere at any time for future reference. The way of converting patients' medical records into electronic format has been implemented throughout the world with the sole purpose of available patient records accessed throughout the world at any point in time. The converted medical records were stored in the local server maintained by the institution or in the cloud for easy access anywhere in the world, with the availability of new sensors

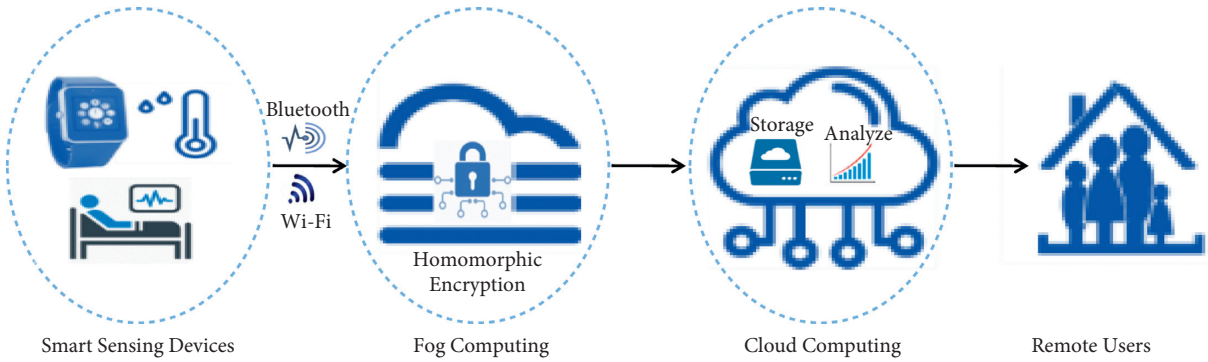

Figure 1: A conceptual framework of IoT-based fog computing healthcare system.

which can be integrated with the patient to monitor their health in real-time and store it in a server or cloud-based system. In case of emergency, integration of the entire tool alerts the medical crew; accurate and timely treatment at the hospital was carried out seamlessly in a matter of minutes. The systematic collection of that information is static and dynamic. The static contains the patient's personal information such as name, age, and gender that need not be changed, while dynamic information contains the streaming information generated by sensors which are changed from time to time.

In the combination of the Internet of Things, big data analytics, and cloud computing, the patient is continuously monitored, and disease is predicted using ensemble methods. Then the proposed method is to securely store patient data and analyze them without requiring personal information. The next step is to process the stored data and diagnose based on disease symptoms. The final step is to alert the required medical personnel without compromising patient privacy. This paper only concentrates on security and privacy for the IoT healthcare data and patient information stored in EHR.

The remainder of the paper is organized as follows: Section 2 describes the relevant works as a literature survey, and Section 3 introduces Fully Homomorphic Encryption. Later Section 4 describes the experimental setup and discusses in detail the results obtained. Finally, Section 5 concludes with a summary of the work and provides future work directions.

3. Literature Survey

In order to reduce the delays inferred by cloud computing, Fan et al. [13] proposed fog-cloud computing to increase efficiency. The authors also addressed the security and privacy challenges using ciphertext policy-attribute-based encryption. Hariss et al. [14] designed a MORE (Matrix Operation and Randomization Encryption) approach Homomorphic Encryption to provide privacy for real-world applications. Sanchez-Guerrero et al. [15] proposed a novel adaptive extended Merkle tree structure to provide privacy to extremely sensitive information stored in EHR. However, all these schemes were applied to already existing stored data, not streaming data.

Pham et al. [7] proposed a smart home healthcare service for the elders who are staying alone using various IoT sensors as a real-time application in a cloud environment. The authors also developed a robot assistant using the gradient boosting decision tree algorithm to find the activities of the body. A cloud-based system is proposed [16] to control and monitor the H1N1 virus. Doctors upload the patients' treatment information in the Amazon EC2 service for future reference in this system. To diagnose Chikungunya, Sood and Mahajan [17] designed a fog-assisted cloud-based system in which users are continuously monitored to collect information stored in the cloud and analyzed in Edge Computing. With the advancement of IoT, Rani et al. [18] also diagnosed Chikungunya and stored patients' information in the cloud. Mobile environment jointly merged with cloud computing to monitor ECG pattern of the patients in mobile designed by Zhang et al. [19]. The system has the details of patients such as name, identification number, gender, age, medical record, and ECG report. However, all these systems fail to meet security measures to prevent user's sensitive health-related data from unauthorized access.

Using the condition-based methodology, Verma and Sood [20] developed a cloud-centric diagnostic system to predict the possible disease in users using medical devices and sensors [21]. The user's personal information and their diagnosed diseases were stored and analyzed in the cloud. Here, Trusted Third Party (TTP) provides security to the users' information, but it can be hacked easily with the public key. Hossain and Muhammed [22] proposed a framework for monitoring Electrocardiogram (ECG) of disabled and older people. The monitoring information is gathered at the cloud platform to be accessed everywhere at the cost needed. Watermarking and signal enhancement techniques are used to secure the data on the client side.

In case medical treatment is done for the patient in a remote network-constrained environment, the medical data have to be transferred securely, and only relevant information can be accessed by authorized users such as healthcare providers. The secure encryption and transmission of data without any loss is also a vital part of security. The data from the sensors will be transmitted to the local edge device; the device further encrypts the data using a public key. The batch process of encryption is done using a new algorithm to encrypt to form the ciphertext whose final

size is larger than the original text size. If a hacker accesses the secure data, the data can only be viewed as a random number and is not possible to decrypt without a secure private key.

The FHE encrypted data transmitted through the edge device will perform the error estimation of the transmitted data. The final data analytics in the cloud server can be performed without any decryption for processing the final output. If the processed data breaches a certain threshold set by the healthcare provider, an immediate alert message will be delivered to the corresponding physician for further diagnosis.

4. Fully Homomorphic Encryption (FHE)

In the FHE method, the security key generated for encryption consists of both private key and public key for secure encryption. The general steps in the process involve four steps such as generation of the encrypted key (symmetric or asymmetric), encryption of data into ciphertext, decryption of the ciphertext using a private key, and evaluation of the transmitted data. In the symmetric encryption method, the general public key will be used for both encryption and decryption of data. The asymmetric encryption method is done using a general public key, while the decryption is done using a secure private key. The final text will be processed in an encrypted ciphertext without decrypting the original message sent by the sensors. In order to evaluate the robustness of the encryption method, the final processed ciphertext will be decrypted using the private key, and the plain final processed results will be displayed.

Fully Homomorphic Encryption technique encrypts the patient's input information to produce the ciphertext without knowing any information about the plaintext, which matches the operation performed on plaintext. Homomorphic Encryption is said to be FHE when it satisfies both properties of addition and multiplication. The basic properties are as follows:

For addition,

$$\left[\mathrm{Enc}_k(a) + \mathrm{Enc}_k(b)\right] \mathrm{mod} X = \left[\mathrm{Enc}_k\left(\left[a + b\right] \mathrm{mod} X\right)\right] \cdot \mathrm{mod}(X). \tag{1}$$

For multiplication,

$$\left[\mathrm{Enc}_k(a) * \mathrm{Enc}_k(b)\right] \mathrm{mod} X = \left[\mathrm{Enc}_k\left(\left[a * b\right] \mathrm{mod} X\right)\right] \cdot \mathrm{mod}(X). \tag{2}$$

The mobile IoT device where the encryption occurs should protect the patient's real identity from public view, as well as in case of an emergency; authorized users can effectively trace the patient using the mobile edge device. The identity can also be morphed by adding a randomly generated pseudonym to the user, but this will further increase the computational cost for each random generation of names in resource-constrained IoT encrypted terminals.

4.1. Concept and Proposed Model. The major drawback of the FHE is the size of the encrypted ciphertext and the minimum storage space required in the cloud server, which is directly related to the performance of the analysis. For an FHE to be fully realizable in the real application with high security, the encrypted file size should be reduced, and the time taken for encryption should be appreciably reduced with the increase in the number of connected sensors. FHE scheme implemented in a low-level language and using a parallel processor reduces the edge domain's encryption time. These are packed to a certain extent and implemented using Somewhat Homomorphic Encryption. The edge device also performs the threshold comparison with the encrypted data without knowing the raw transmitted data, and the result will be a simple yes or no. The basic point in privacy is preserving encrypted data by evaluating some ideal properties such as accuracy, no reversibility, diversity, revocability, randomness, and performance.

The Edge Computing devices, if implemented properly, will have less latency and stringent quality of service requirements. Figure 2 presents the architecture of polynomic Homomorphic Encryption in the healthcare system. The main idea is to bring the core cloud computing to mobile devices, connected sensor devices, and actuators. Most of the edge devices will be on the move. Therefore, the transmitted signal should not get retransmitted or duplicated in another edge node. This will further lead to memory constraints in the cloud computing storage.

The proposed model will strip the sensor data into a simple number with comma-separated phrases and fill them into a linear algebraic expression. Each polynomial will be added with a simple operator, and the idea is to shorten the sent polynomial into a manageable packet size. The original data will be added to the final text for further processing.

The network connection between the sensor terminals and the edge device is where most of the attacks can take place. The transmitted data from the sensor can be attacked by Eavesdropping, Sybil attack, sleep deprivation attack, and a man in the middle attack. The data are transmitted between the sensor and edge devices using SHA message verification techniques. The medical sensor signals have to be routed through a trusted secure network channel. Another way to prevent the attack is to add a new layer of security to the transmitted signal in the edge device before transmitting through the cloud network. One such type of protection requires less time to encrypt the continuous signal transmitted by the sensor into an encrypted signal using a public key provided by the user within a few milliseconds. Then, the encrypted signal is transferred to the cloud for further processing. The advantage of Fully Homomorphic Encryption is that the data need not be decrypted for processing; only if the signal value exceeds a certain threshold, will the authorized user receive the alert text message, including the physician in charge. The data will be decrypted by the authorized user using a private key provided by the medical institution. The physician has access only if it is decrypted by the user.

5. Preliminaries

In this section, we review our associated cryptographic schemes and mathematical concepts.

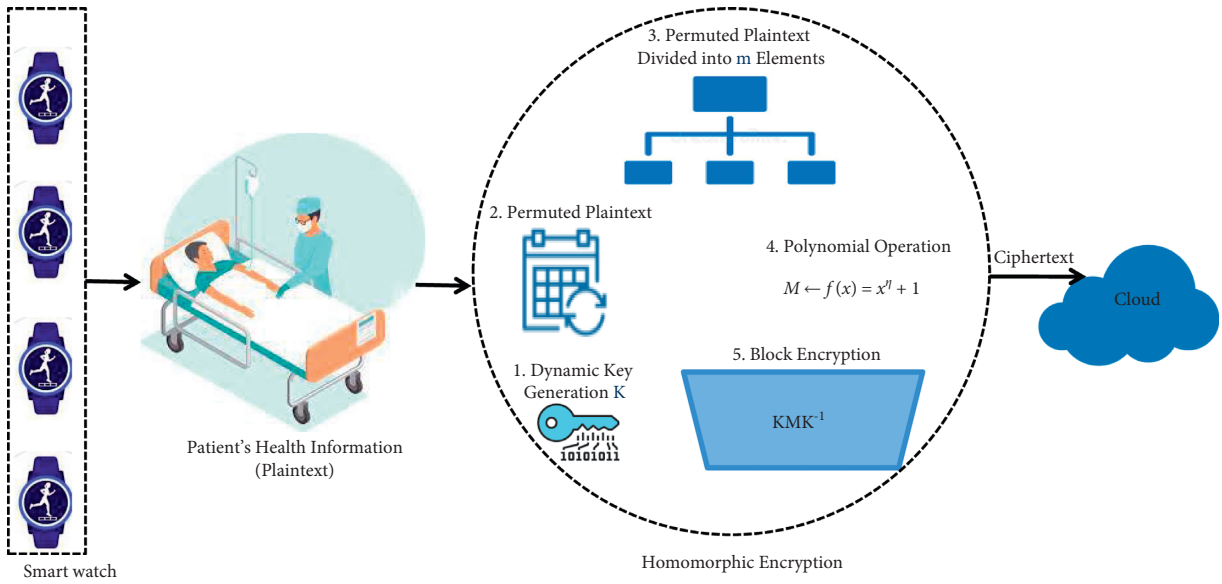

Figure 2: Architecture of polynomic Homomorphic Encryption in the healthcare system.

5.1. Homomorphism. Assume that G^* is a cyclic multiplicative group and G and G' are two cyclic additive groups of a prime order. Let kernel $K(f)$ be a kind of function from $G \longrightarrow G'$ and a and b belong to G. φ is mapping from $G/K \longrightarrow G'$, which satisfies the following properties:

(i) φ is well-defined: if $f(a)$ is equal to $f(b)$, then $\varphi(ka) = \varphi(kb)$. $ka = kb$ if and only if $ab^{-1} \in k$. Then, $f(a * b^{-1}) = e'$.

(ii) φ is one-one function: let $\varphi(ka) = \varphi(kb)$.

$$\begin{aligned} &f(a)\Delta[f(b)]^{-1} = e^{-1}, \\ &f(a)\Delta f\left(b^{-1}\right) = e^{-1}, \\ &f\left(a * b^{-1}\right) = e^{-1}, \\ &ab^{-1} \in k, \\ &ka = kb. \end{aligned} \tag{3}$$

(iii) φ is onto function: for every a which belongs to G and $x \in G'$, $f(a) = x$; then, $\varphi(ka) = x$.

(iv) φ is homomorphism: let $ka, kb \in G/k$; then $\varphi(ka \otimes kb) = \varphi(k(a * b))$.

5.2. Secret Sharing Scheme. A secret sharing scheme divides the secret k into x shares with a y+1 share that is unable to reconstruct. Based on Lagrange's theorem, the order of any subgroup H of a finite group G divides the order of G.

5.2.1. Secret Construction Phase

Step 1: assign a polynomial function $f(z)$ with degree s, in which all coefficients belong to Z.

Step 2: compute the share $x_i = f(i) (\mathrm{mod} N)$ for $i = 1, 2, 3, \ldots, n$.

Step 3: user publishes a list of n shares like $(x_1, x_2, \ldots..x_n)$ and each x_i is circulated among patients.

5.2.2. Secret Reconstruction Phase

Step 1: any y+1 share $(x_{n_1}, x_{n_2}, \ldots, x_{n_{t+1}})$ can be able to reconstruct the secret k.

Step 2: Compute $x = f(0) = \sum_{i=n_1}^{n_{t+1}} x_i (\prod_{j=n_1}^{n_{t+1}} y_j / y_j - y_i (\mathrm{mod} N)$.

The encryption method can compute the average, standard deviation, and regression coefficient of the prediction without performing any decryption algorithm. The method of Ring Learning with Errors (RLwE) from Peikert and Regev [23] has some assumptions described in the following. The ring $(ri) = Z_p(x)/f(x)$; the ring has n degree of polynomials with coefficient in Z_p; the assumption is that all the polynomial numbers of samples are of the same form with a random number and error number. The ciphertext is equivalent to the noise present in the error distribution instead of a uniform number. The polynomial $f(x)$ chosen for the operation is x^n+1, with n having a power of two. The ring elements after multiplication will have lesser than L^2 norm multiplication of the component, which is basic polynomial multiplication, and the addition is componentwise coefficient addition. The error distribution is of the Gaussian form D with $\varepsilon > 0$ based on the probability density function. The multiplication function performed on the ring Z_p will only increase the size of the ciphertext by a small amount and reduced time. The time, size of the key, and ciphertext are computed using MATLAB software before implementing in the edge devices environment.

5.3. Somewhat Homomorphic Encryption Algorithm. SHE = Enc (keygen, encryption, addition, multiplication, and decryption).

Step 1. Create the asymmetric encryption keygen (public and private key).

The parameter for key generation is based on a polynomial function $f(x) = x^n + 1$, where n has a power of two. The modulus s is a prime number that is related by modulus $(2n)$. Then, s variable defines the ring parameter $ri = \mathrm{mod}(Z_p(x)/f(x))$ described earlier. The Gaussian functions have a discrete error value given by $\varepsilon = (D(Z_p), \varepsilon 1)$ with the standard deviation. The discrete Gaussian $D > 0$; then only the multiplication method will work. They have to be chosen based on the hidden security parameter K. The ring element will define the secret key (pv.K) with a uniform random element. An additional error term will be added to the key. The public key (pub.K) contains the less random element and an error term to be available for encryption by the edge device. Pub.$K = \mathrm{mod}(\mathrm{number} * \mathrm{key} + \mathrm{mod}(t) * \varepsilon)$ and $pv.K = \mathrm{mod}(\mathrm{number} * \varepsilon - \mathrm{pub}.K)$. The private key is provided with the authorized person, whereas the public key for encryption is available on the edge device for encryption protocol.

Step 2. Encrypt the plain data into a secure ciphertext using a public key and private key.

The message or continuous number will be encrypted based on the degree of the polynomial (n) with a continuous or a particular text message. In general, the computation of the normal text (norm) or number into an encrypted ciphertext contains all the elements belonging to the general variable.

$$C_p = a_0 * x + t * x + \mathrm{norm}(a_1) + \mathrm{mod}(t) * x. \quad (4)$$

Step 3. Packet the additional data into a smaller ciphertext for easy transmission.

The packets from the edge device will be split into easy modules after encryption based on mod(t) value. If the degree of the polynomial is larger, then the encryption time is also more. Depending on the encryption device, the public key will be selected based on the available RAM in the edge device for easy encryption and transmission time. The complexity increases with the increase in the number of devices connected to a single edge device.

Step 4. Process the encrypted text for finding the critical threshold for the patient transmitted data.

The decryption of the whole normal text or number can be performed with the appropriate error value added to the public key, which is also a part of a private key generation. The decryption method with less error is achieved by the RLwE method. In general, the decrypted text which will be computed for mod(t) is given by

$$\mathrm{decrypt\,text} = \sum_{0}^{\delta} Cp_i Pv.K_i + \mathrm{number} \in Z_x. \quad (5)$$

The number in the decryption key will be based on the preprogrammed packaged size of the ciphertext encrypted in the edge device.

Step 5. If the threshold for the sensor is breached, alert the user and transfer encrypted data for cloud transfer ahead of the data packet. Additional functional methods are designed in the edge device, such that after the encryption of the number, the device will perform a simple logical operation. If the encrypted number of the ciphertexts exceeds a certain threshold value, an alarm text will be delivered to the patient and the attending physician as an added advantage.

Step 6. Process the sent data in the cloud without decryption, and if a critical condition is breached, alert the medical team immediately. The data pack sends to the server in a sequential time because of the large volume of the data transmission to the cloud server. The timing of the device will be adjusted based on the encryption timing required for sending the data. The low-cost sensor module will not have a facility to store the large volume of transmitting signal. Therefore, the storage, encryption, and timely delivery have to be taken care of by the edge device attached to the sensor module. Even if the network availability is not there, the device will store a certain value for a certain period and upload them when the connection is established with the cloud server.

The advantage of the current method is that the transmitted data will be sent in packets and reassembled in the cloud storage device for further processing.

6. Implementation

The biological signal-like ECG device was mimicked using a random signal generator in the MATLAB environment. The signal from the generator is stored in a series of column values for every 30 seconds before sending it to an encryption algorithm. Similarly, in the edge environment, the sensor data will be processed in a batch mode every 30 seconds before the encryption algorithm processes the incoming message. Each column value will be encrypted based on the number size sent from the sensor. For example, suppose the temperature and pressure sensor are attached to the patient. In that case, a single bit temperature and pressure value will be taken at every 2-minute interval, whereas the ECG-like signal will be transmitted for every second from the patient. The encryption text size will vary at the Edge Computing device depending on the monitoring sensor. Therefore, the size of the number sent for encryption is given by toy (2), medium (128), or large numbers (1024). The function for encryption is defined in MATLAB for easy software implementation, and later it can be converted into a basic C code for edge device implementation. The current simulation for Somewhat Fully Homomorphic Encryption with RLwE was implemented for simple cases with three different scenarios in Intel Core 2 Duo machine at 1.6 GHz with 16 GB RAM, which were performed to analyze the encryption time and size in the edge environment before transmitting to a cloud server. The size of the encryption along with the public key also plays a vital role in the performance of the edge device to process the incoming values continuously without any delay.

7. Results and Discussion

The algorithm helps to solve the time taken and the computation necessary for the edge device to handle many sensors from each patient in a shorter period of time to encrypt the message without affecting the performance of the system. The code should safeguard from a certain type of attack against the vulnerabilities posed by the FHE code. One way is to implement the secure wireless connection from the sensor module to the edge device module with SHA1 hashes to verify the integrity of the received signal from the wireless module. The current paper explains the time required to perform the encryption in edge device based on the numbers and text received from the various sensor modules.

8. Encryption Time of Text and Numbers

The encrypted text and values at various levels are based on the received text size and the time available at the edge node to perform the required encryption operation. In the case of numbers, the sum and product of the two can be directly encrypted from the available space without any delay. If the text message is sent from the sensor, then the text has to be converted into a number first before performing the encryption operation. Such conversion will save the time required for encryption at the edge interface. The encryption performed using a public key at the edge device will have to separate the encryption timing based on the received value as well as the integrity of the incoming number or text. The encryption of a smaller number will require less time to process; the encryption based on n number of prime numbers chosen indicates that the mod(t) should be greater than the length of the received signal. The initial parameters s and n also need to be smaller for resource constraint edge device or based on the time interval at which the signal is transferred from the sensor (e.g., temperature sensor and pressure sensor) attached to the patient. The physiological signals like ECG have to be monitored continuously for any variation in the patient's heart rate, which is critical for monitoring patient health. The larger signal will have to be packed in a certain order for easy processing at the edge environment. The addition operation of the encryption performed much faster, even for the larger size of the number. The multiplication of the prime number polynomial in the encryption takes a larger time to convert and encrypt. The idea here is to choose the multiplication factor based on the size of the text and the degree of a polynomial the edge device will select based on the incoming signal. This will be a trade-off in the edge device performance versus the selection of appropriate encryption for each received signal from the sensor module.

9. Implementing Numbers Packaging in Edge Device

Here, we propose converting the incoming signal into a separate column for easy conversion of signals into a useful number. First, the ciphertext encrypted with a smaller text will be placed separately with a sensor number in the cloud server, and the larger number will be encrypted into multiple files for easy transfer. For example, the encrypted size of the ciphertext based on the incoming package will have a separate number selected during the private key generation. The selected number will add a single value at the appropriate interval for performing the decryption mechanism. The ciphertext sends from the edge device will have a unique number that will match the private key based on the size of the ciphertext received.

10. Decryption and Comparison of Encrypted Text

The ciphertext sent from the edge device after performing the specific encryption depends on the degree of polynomial selected for encryption based on the size of the packed number. In a Fully Homomorphic Encryption system, the received signals can be operated without performing any decryption algorithm. For example, the average value of the temperature sensor from the encrypted message can be calculated in the cloud server without performing any decryption of the real data. If the threshold is breached in the server, an alert message would be delivered from the cloud platform for a list of authorized users. If the decoding of data has to be performed, it will be performed only with the patient consent that possesses the authorized key. The decryption of the encoded message was based on the size of the packed number sent and the duration of the value received by the edge platform. The personal information and sensitive patient details will be saved using bitwise encryption rather than packed encryption as described previously. The sensitive data requires more encryption power than the packed encryption performed for fast encryption of the sensor readings.

To avoid the issue of large bit size key, the prime number was chosen for the public key to determine the size and the time required for the encryption of the number and the text is also converted into a random number before performing an encryption mechanism. The two ring elements were used for both the public key and the ciphertext encryption, and the whole method was performed in a single degree reduction during the multiplication operation of the encryption step. The developed method was implemented in MATLAB based on the message space modulus t and power of the prime number factor n dimension; the ciphertext degree for each type of encryption method is shown in Table 2. The time required for encryption using a public key and a private key in general is based on the text size chosen based on packet size (tiny, medium, and large). The public encryption key required for performing individual addition and multiplication based on the general public key is evaluated to assess the efficiency of the multiplication encryption mechanism. From the calculated result, it is evident that the addition requires less time than the multiplication of prime numbers. The calculation also shows the maximum time required for performing simple encryption if the complexity of the ciphertext size increases. The encrypted size of the public key size increases with the degree of the

Table 2: Encryption operation time and size for various message spaces.

Message space (mod(t))	n (ciphertext degree)	Encryption operation time				Encrypted size	
		Pub. key ms	Pv. key ms	Add. ms	Mult. ms	Pub. key KB	Ciphertext KB
2	512 (1)	25	57	<1	<1	12	24
	4096 (4)	210	520	2	150	386	781
	8192 (10)	425	1410	7	860	2256	4526
128	1024 (1)	55	112	<1	<1	29	60
	4096 (4)	225	610	4	160	490	975
	8192 (10)	448	1654	7	1140	5534	11052
1024	1024 (1)	52	108	<1	<1	35	69
	4096 (4)	225	620	4	214	535	1091
	8192 (10)	455	1825	9	554	6010	12105

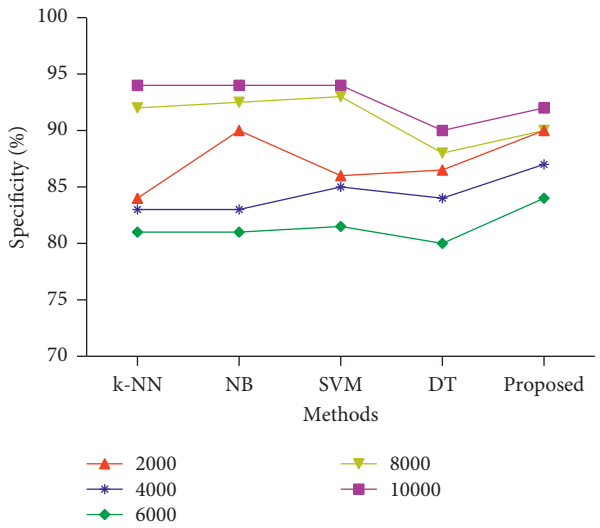

Figure 3: Analysis of specificity.

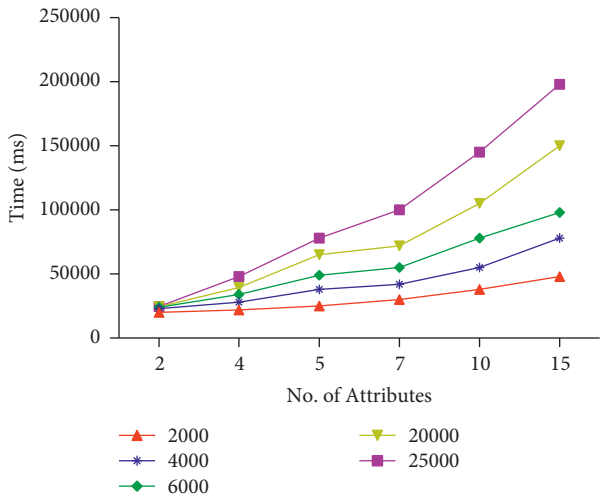

Figure 4: Time taken to encrypt attributes.

ciphertext to perform the operation. The algorithm helps to solve the time taken and the computation necessary for the edge device to handle in a shorter period of time to encrypt the message to the server without affecting the performance of the system. The code should safeguard from a certain type of attack against the vulnerabilities posed by the FHE code. The code implemented by the SHE based on the new algorithm is lightweight more suited for the encryption at the edge device itself for preserving the privacy of the patient.

For Somewhat Homomorphic Encryption (SHE), the RLwE method using the parameters given in the algorithm is implemented in MATLAB. The time and size are taken for key encryption along with addition and multiplication operation performed during encryption operation.

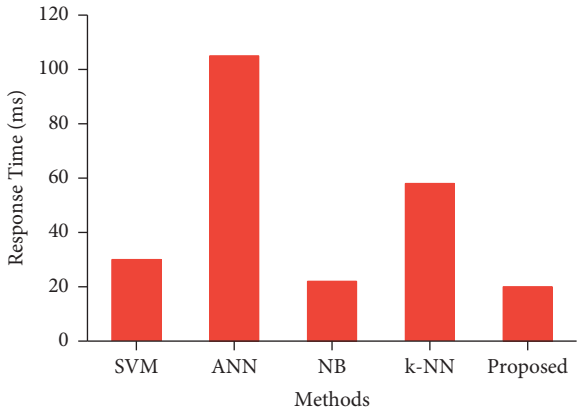

Figure 5: Analysis of response time.

11. Experimental Results

The experiment has been accomplished to evaluate the proposed model using different instances with four different models such as k-Nearest Neighbor (k-NN), Naïve Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT).

The specificity analyzed between the proposed model and four different models is presented in Figure 3 for clear understanding. From the figure, it is noted that the proposed method outperforms the existing baselines.

It is observed from Figure 4 that the total time taken to encrypt 2000 records is less than others. Although, it increases gradually when there are more numbers of records in the input dataset.

It is noted from Figure 5 that the response time taken by the proposed model is less when compared with other existing baseline models.

12. Conclusions

In the current paper, we proposed a modified LWrE method based on Somewhat Homomorphic Encryption, which can be practically applied to a medical-oriented sensor device that requires confidentially protecting patient information. This method helps to solve the problem by systematically monitoring patient health in real-time without divulging sensitive data. The proposed work provides a secure key that the patient can only authorize to decrypt the original raw data from the sensor attached to the patient. The encryption performed in the MATLAB software helps to emulate a real device scenario and helps to fine-tune the operation required for the particular type of sensor and the number of packaging methods required for efficient operation of the edge device time. The encryption scheme can be programmed based on the number of sensors attached to the particular edge environment by estimating the number of times a device transmits data to the edge device. In the future, the proposed approach will be enhanced to meet the requirements of time-sensitive applications. Also, the proposed SHE shall be considered for the Federated Learning applications for ensuring better security.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the Science and Engineering Research Board (SERB), Department of Science and Technology, India, for financial support through the Mathematical Research Impact Centric Support (MATRICS) scheme (MTR/2019/000542). The authors also acknowledge SASTRA Deemed University, Thanjavur, for extending infrastructural support to carry out this research work.

References

[1] Indian Healthcare Industry Analysis, "Indian Healthcare Industry Analysis," October-2018, https://www.ibef.org/industry/healthcare-presentation.

[2] C. S. Nandyala and H.-K. Kim, "From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals," *International Journal of Smart Home*, vol. 10, no. 2, pp. 187–196, 2016.

[3] A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[4] P. Verma and S. K. Sood, "Fog assisted-IoT enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[5] M. S. Hossain, M. A. Rahman, and G. Muhammad, "Cyber-physical cloud-oriented multi-sensory smart home framework for elderly people: an energy efficiency perspective," *Journal of Parallel and Distributed Computing*, vol. 103, pp. 11–21, 2017.

[6] S. K. Sood and I. Mahajan, "Fog-cloud based cyber-physical system for distinguishing, detecting and preventing mosquito borne diseases," *Future Generation Computer Systems*, vol. 88, pp. 764–775, 2018.

[7] M. Pham, Y. Mengistu, H. Do, and W. Sheng, "Delivering home healthcare through a cloud-based smart home environment (CoSHE)," *Future Generation Computer Systems*, vol. 81, pp. 129–140, Apr. 2018.

[8] V. Subramaniyaswamy, R. Logesh, M. Abejith, S. Umasankar, and A. Umamakeswari, "Sentiment analysis of tweets for estimating criticality and security of events," *Journal of Organizational and End User Computing*, vol. 29, no. 4, pp. 51–71, 2017.

[9] D. Li, H. W. Park, E. Batbaatar et al., "Application of a mobile chronic disease health-care system for hypertension based on big data platforms," *Journal of Sensors*, vol. 2018, pp. 1–13, 2018.

[10] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, 2021.

[11] A. Rehman, S. U. Rehman, M. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering*, vol. 8, 2021.

[12] R. H. Jhaveri, S. V. Ramani, G. Srivastava, T. R. Gadekallu, and V. Aggarwal, "Fault-resilience for bandwidth management in industrial software-defined networks," *IEEE Transactions on Network Science and Engineering*, IEEE, vol. 8, no. 4, pp. 3129–3139, 2021.

[13] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, "A secure and verifiable outsourced access control scheme in fog-cloud computing," *Sensors*, vol. 17, no. 7, p. 1695, 2017.

[14] K. Hariss, H. Noura, and A. E. Samhat, "Fully Enhanced Homomorphic Encryption algorithm of MORE approach for real world applications," *Journal of Information Security and Applications*, vol. 34, pp. 233–242, 2017.

[15] R. Sanchez-Guerrero, F. A. Mendoza, D. Diaz-Sanchez, P. A. Cabarcos, and A. M. Lopez, "Collaborative eHealth meets security: privacy-enhancing patient profile management," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1741–1749, 2017.

[16] R. Sandhu, H. K. Gill, and S. K. Sood, "Smart monitoring and controlling of pandemic influenza A (H1N1) using social network analysis and cloud computing," *Journal of Computational Science*, vol. 12, pp. 11–22, 2016.

[17] S. K. Sood and I. Mahajan, "Wearable IoT sensor based healthcare system for identifying and controlling chikungunya virus," *Computers in Industry*, vol. 91, pp. 33–44, 2017.

[18] S. Rani, S. H. Ahmed, and S. C. Shah, "Smart health: a novel paradigm to control the chickungunya virus," *IEEE Internet of Things Journal*, vol. 4662, p. 1, 2018.

[19] X.-S. Zhang, F.-Y. Leu, C.-W. Yang, and L.-S. Lai, "Healthcare-based on cloud Electrocardiogram system: a medical center experience in middle taiwan," *Journal of Medical Systems*, vol. 42, no. 3, p. 39, 2018.

[20] P. Verma and S. K. Sood, "Cloud-centric IoT based disease diagnosis healthcare framework," *Journal of Parallel and Distributed Computing*, vol. 116, pp. 27–38, 2018.
[21] H. Mostafaei and M. S. Obaidat, "Learning automaton-based self-protection algorithm for wireless sensor networks," *IET Networks*, vol. 7, no. 5, pp. 353–361, 2018.
[22] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial internet of things (IIoT) - enabled framework for health monitoring," *Computer Networks*, vol. 101, pp. 192–202, 2016.
[23] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Advances in Cryptology—EUROCRYPT 2010*, vol. 6110, pp. 1–23, 2010.