**Securing Internet-of-Things Data in a Healthcare Surveillance Network with a Hybrid Encryption Algorithm**

*B.Vijaya lakshmi*

*Associate Professor*

*Department of ECE*

*Gayatri Vidya Parishad College of Engineering for Women*

*Visakhapatnam, Andhra Pradesh, India*

*S Praveena*

*Student*

*Department of ECE*

*Gayatri Vidya Parishad College of Engineering for Women*

*Visakhapatnam, Andhra Pradesh, India*

**Abstract**

IoT-based healthcare solutions are used for remote health monitoring and diagnostics to minimize the cost of health care services. Secure data communication is crucial for treatment and monitoring. Furthermore, it is vital to safeguard the patient's information from abuse and data manipulation while it is in transit because other devices can easily trace it. IoT networks have limited CPU capability, memory, and power, making performing the expensive operations required by ciphering algorithms challenging. As a result, they require a lightweight security approach that consumes fewer resources. A hybrid cipher approach for safe data transport from IoT healthcare devices is proposed in this paper. Three cipher algorithms are used in the encryption process, including a modified, lightweight Salsa20 with 16 words (each word is 32 bits). To begin, for each block of sensing data, a five-dimensional chaotic map is employed to generate 32 keys of 64-bit length. Second, to make the produced keys appear dynamic, random shifting is performed on them. Third, an exclusive-or operation is carried out between a newly constructed key, which includes sensing data, and modified Salsa20 words. Fourth, the proposed method improves security by using the addition DNA operation between the result of the previous step and the random selection word in Salsa20 state after both have been turned into DNA form. Finally, before sending data to the server, sensor data is hashed using the SHA3-256 hash method to assure data integrity. The hash data result, the DNA result after conversion to a decimal number, and two random integers for shifting keys and picking Salsa20 words comprise the entire delivered data.Based on tests done by NIST, the suggested algorithms provide data content security features like secrecy, authentication, and non-repudiation. They also work with all types of sensors to send secure data to the final controller.

**Keywords:** *five-dimensional chaotic Map, SHA3-256 hashing; Salsa20 Algorithm, DNA cipher technique*

1. **INTRODUCTION**

E-healthcare is a major IoT use today. Internet-based healthcare's future is IoT-based. In e-healthcare, IoT sensors collect sensitive data for remote monitoring, therapy, and more. The electronic healthcare system uses it for patient care and labour reduction [1]. However, various security weaknesses may undermine internet-based patient health data transfer. Thus, data confidentiality and privacy, device and system integrity, user authentication, and data and design availability are all under attack [2].

Thus, a safe system and effective encryption are needed [3]. IoT devices have limited processing, memory, and power, making ciphering algorithm operations difficult. They need a lightweight, resource-efficient security system. IoT devices struggle with traditional authentication and encryption [4]. Lightweight cryptography uses symmetric and asymmetric algorithms. In asymmetric algorithms, public and private keys differ. Symmetric block and stream ciphers encrypt and decrypt using the same key. Symmetrical algorithms are only as good as the sender and receiver's key transmission [5]. One of the best Internet of Things communication protocols is MQTT, which can be deployed on lightweight, inexpensive, low-power, and low-memory devices. Since it lacks security, this protocol cannot provide safe authentication between communication parties. It doubts data integrity and secrecy [4]. This research presents a hybrid secure technique with a five-dimensional chaotic map-generated symmetric secret key. The MQTT communication protocol improves IoT healthcare data transformation dependability without affecting monitoring system performance. The technique collects patient IoT health data. It encrypts and decrypts data using Salsa20 5D chaotic map function, SHA3-256 hash function, and DNA ciphers with two sensors (MAX30102 and MLX90614) and one microcontroller (Raspberry Pi 3 Model B).

The paper keeps going: Section 2 covers IoT healthcare security studies. Section 3 secures IoT healthcare. Section 4 discusses the proposed hybrid cipher. Section 5 analyses performance and results, while Section 6 concludes.

1. **RELATED WORKS**

This section discusses the secure IoT-based healthcare system. Vijayalakshmi and Arockiam [6] proposed a hybrid security solution in 2018. This solution prevents unauthorized access via cryptography. Thus, this technology secures E-health data interchange. In 2019, Tao et al. [7] proposed Secure Data with four layers: IoT, Fog, cloud, and healthcare provider. KATAN and secret cipher share simulated energy cost, hardware frequency, and computation time. This difficult and expensive method's distributed cloud design prevents single points of failure. Naveen, Sharma, and Nair 2019[8] proposed using the ASE encryption approach to monitor a patient's physiological parameters every 10 seconds, addressing mistakes, duplication, and early anomaly detection. After collecting sensor data, replaced missing data with special characters to simplify data management. S. Joshi and S. Joshi [9] presented a predictive, alarm-based, internet-of-things-based sensor-based health monitoring system in 2019. An encryption-decryption architecture protects sensor-captured patient data in matrix form utilizing two encryption keys, one with a patient ID and the other with ASCII distributed. Besher et al. [10] 2020 suggested receivers decipher patient data using the inverse of the two encryption keys to get the original data. Protects patient data. Balakrishnan et al. [11] developed an intelligent fingertip heart rate sensor to remotely and continuously monitor patients' blood pressure and heart rate in 2020. It was encrypted lightly. Linear regression classifies arrhythmia. In 2021, Pandey et al. [12] introduced Dynamic Matrix Encryption (DME), which encrypts and decrypts data using transpose and swapping matrix functions. The receiver only needs an ATM PIN for the specified DME algorithms. Relocating values using matrix methods is fast. Memory space is preserved since encrypted data is the same length. Khadidos et al. [13] introduced Probabilistic Super Learning (PSL) Random Hashing (R.H.) to secure IoT-cloud healthcare data in 2022. Learning reduces IoT sensor prices in this research. The data matrix hash value is used to generate a random key for Elliptic Curve Cryptography (ECC) data security. The enhanced ECC-RH method encrypts and decrypts data using the random hash key.

1. **INTERNET OF THINGS HEALTHCARE SECURITY**

Medical data security is critical in the healthcare industry. Wearable health apps transmit critical data. Internet-connected hackers target security vulnerabilities such as DoS attacks and tampering with medical data and analytics [14]. The core cyber security concepts are confidentiality, integrity, and availability (CIA) [15]. According to the confidentiality property, information is owned when it is not shown to people, processes, or devices who are not authorized to see it; the data integrity requirement aims to ensure that the data has not been affected throughout the wireless transmission before it reaches its intended location; and users must be able to access the services and data they require at the appropriate time. Healthcare apps must be available at all times to ensure that data is accessible to patients and emergency services [15,16]. In intelligent healthcare systems, application-oriented services compromise security and privacy in exchange for a faster design time. One-dimensional security methods do not provide enough protection against complex threats. Attacks are foiled via lightweight cryptography. It covers advanced healthcare sensing. Intelligent healthcare communication, unlike other IoT components, is well-known, can be addressed remotely, and can be used as a launching pad for assaults on non-healthcare IoT applications, such as Denial-of-Service (DoS). Many modern communication security approaches are cryptographic in nature [17].

**Domination Chaotic Maps**

Mathematical subfield "chaos theory" It studied nonlinear, starting-state sensitive dynamical systems. Minor input parameter changes can affect output. Chaotic systems' output increases cryptographic security [18]. Pseudo-random generators prohibit wireless key exchange. Only the seed produces the same encryption key. Chaos generates secure pseudo-random numbers. The same pseudo-random sequence cannot be produced without starting conditions. This property encrypts device-to-device communications. The attacker cannot obtain the pseudo-random sequence even with the system's generating equations [19].

The proposed system's equations are

𝑥 = 𝑥 + (−𝑠 × 𝑥 + 𝑦 × 𝑘 − 𝑟 × 𝑝) (1)

𝑦 = 𝑦 + (−𝑦 − 𝑥 × 𝑧 + 𝑟 × 𝑥 − 𝑢 × 𝑝) (2)

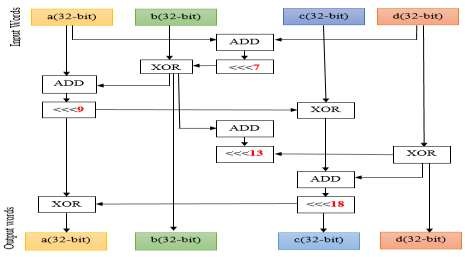
𝑧 = 𝑧 + (𝑧 × 𝑥 × 𝑦 − 1.5 × 𝑠 × 𝑝 − 𝑘) (3)

𝑘 = 𝑘 + (𝑠 × 𝑥 + (𝑢 × 𝑦 − 𝑟 × 𝑘)) (4)

p = p + (b × ((x + k)/z) + y) (5) where s=0.95, r=0.5, b=0.01, and u=1.1. System behaviour vectors are x, y, z, k, and p [20]

**Lightweight Salsa20 Cipher Algorithm**

Salsa20 encrypts meter readings and other data using stream cipher, a simple security mechanism. Keystream on salsa20 processes, which are mathematical operations utilizing salsa20, produce a 512-bit keystream block sequence from an input of 256-bit key K = (k0, k1, k2, k3, k4, k5, k6), 128-bit key K, or 64-bit nonce N = (n0, n1) and 32-bit words [21]. Salsa20 outputs the C.T. block from the nonce, key, and 64-bit block counter C = (c0, c1) around i [22]. This function operates on a (4×4) 32-bit matrix [22,23]:



**Figure 1.** Salsa20 Quarter-Round.

Z = X+X(20), where X(r) = Round(X) signifies the round equation of Salsa20 and the "+" denotes word wise addition modulo 232. When Z = X + X(r), it is referred to as "r-round Salsa20" or "Salsa20/r". The round equation is composed of the following quarter-round (nonlinear) equations: [21]: A four-word vector (a, b, c, d) is modified

𝑏 ← 𝑏⊕ ((𝑎 + 𝑑) < 7) (6)

𝑐 ← 𝑐⨁ ((𝑏 + 𝑎< 9) (7)

d ← d⨁ ((c + b) <13) (8)

𝑎 ← 𝑎⨁ ((𝑑 + 𝑐) < 18) (9)

The Quarter-Round (QR) equations are exercised to columns (x0, x4, x8, x12), (x5, x9, x13, x1), (x10, x14, x2, x6), and (x15, x3, x7, x11) in odd rounds, and rows (x0, x1, x2, x3), (x5, x6, x7, x4), and (x10, x11, x8, x9) in even rounds [23].

**DNA Cipher Algorithm**

Biological DNA may be used as the storing media for steganography and cryptography. It is possible to perform molecular computations on biological DNA structures and then apply them to conventional ciphers [24]. The digitized DNA datasets from NCBI are available for use in a variety of genome sequencing projects for cryptographic purposes. DNA is made up of two lengthy strands of nucleotides and is helical in shape. The four bases that make up a nucleotide are guanine, adenine, cytosine, and thymine. The coding approach can be used to swiftly translate any digital data into a DNA sequence using Table 1 [24].

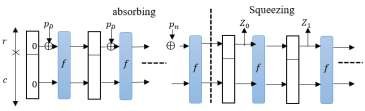
**Table1.**DNA code.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Binary Numbers | 00 | 01 | 10 | 11 |
| DNA Code | A | C | G | T |

DNA algorithm's guiding principle [25] is as every text character is converted to an ASCII code, the binary representation is converted to DNA sequence conversion utilizing the DNA Genetic Code (Table 1).

**Secure Hash Function Algorithm (SHA-3)**

The SHA-3 Algorithm, which is the newest member of the secure hash family of algorithms (SHA), is the basis for several technologies, including Blockchain, security apps, and protocols like TLS, SSL, PGP, SSH, IPsec, and S/MIME. A group of sponge functions known as the Keccak family of cryptography functions, which includes the SHA-3. The National Institute of Standards and Technology (NIST) served as the event's host. Minor changes in the input lead to significant changes in the output [26]. Figure 2 [27] depicts how the sponge was made.

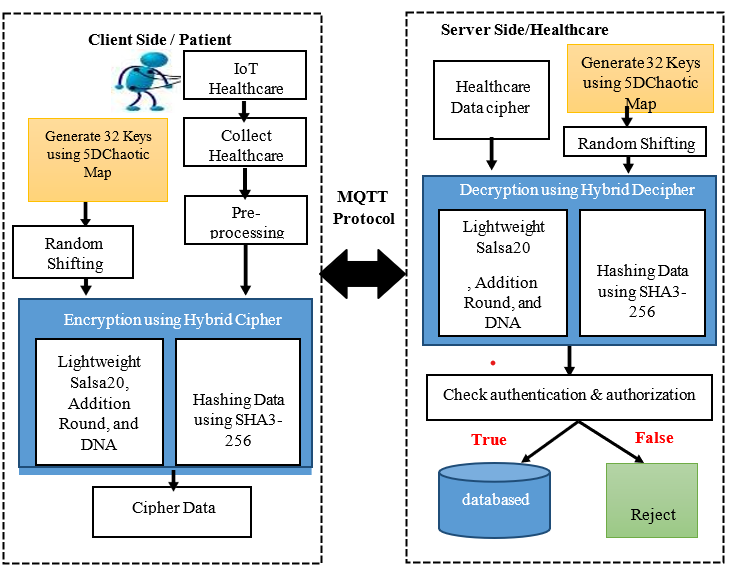


**Figure 2:** Sponge Development Mechanism

For 1600, add R and c. The padding function enters the supplied message. Pad data with 10000000 bytes [27]. Padding one is followed by padding zero and the number of zeroes. Keccak-f [1600] has 55 64-bit lanes for 1600 bits. Therefore, utilize 24 rounds of this type. Figure 3 [28] depicts Keccak –f's 24 cycles' five internal operations [1600][27].

1. **THE PROPOSED APPROACH**

As seen in Figure 3, the suggested method is divided into two stages: the patient stage (client side) and the healthcare stage (server site). Furthermore, some activities are performed on the client side as forwards operations (encryption process), while others are performed on the server side as backward operations (decryption process). The proposed hybrid encryption method enables the Internet of Things to securely and reliably communicate health care data without the risk of being captured by an attacker or other unauthorized parties.



**Figure 3:** IoT Health Care Monitoring and Tracking System Block Diagram

**Reading IoT Healthcare Data**

The suggested system uses MAX30102 [29,30] and MLX90614 [31,32] wearable sensors. I2C sends vital indications like SpO2 and pulse rate from the MAX30102 to the Raspberry Pi3. Remotely measuring a patient's body temperature with MLX90614 is easy with a Raspberry Pi 3. O2, H.R., and TEMP indicate pulse oximetry, heart rate, and temperature in the proposed system. Validating the reading data before saving it into O2, H.R., and TEMP ensures that it is error-free and integer-formatted. Table 2 shows the system's IoT sensor data**.**

**Table2** Data Sensor Input Samples

|  |  |  |  |
| --- | --- | --- | --- |
| **Noofsample** | **IoT Data Related Medical Care** | | |
| O2 | HR | TEMP |
| 1 | 091 | 078 | 37 |
| 2 | 093 | 076 | 38 |
| 3 | 099 | 082 | 36 |

**The Initial Data Processing**

The proposed system makes use of a padding mechanism, and it is only put into effect if the lengths of the three parameters stray from the values that were pre-specified. For instance, addition is necessary whenever the size of O2 is equal to three; type "0"to the left of the O2 value, and type "00"to the left of the O2 value that is ten or less. The identical process was carried out on all of the other sensing data.

**Reading Data Sensors Utilizing Concatenation**

In this stage of the procedure, the suggested system will implement the padding process by stringing together O2, H.R., and TEMP to form a state. In order to facilitate the encryption and hashing processes, the length of the string state will be set to eight characters.

1. **PROPOSEDHYBRID ENCRYPTION**

A 5-D chaotic system generates 32-keys to encrypt sensor data. A random shift each time (for dynamic change), then Salsa20 encryption technique to generate 512 bits block of keystream matrix 4×4 (16 words), addition round function, and DNA operation to avoid threats. Figure 5 displays the proposed technique flowchart. Stages of the suggested method: starting with the collection of health data for an individual patient, data preprocessing, then generation of secret keys (32-keys of length 64-bits) using five-dimensional chaotic maps, shifting operation applied on the generated key to increase security and make the key dynamic by using a random number between (0, 63), using this secret key to encrypt/decrypt the data using a combination of three encryption algorithms. modified light Salsa20 generates random numbers in the sequence of x, y, z, k, and p using a 5D chaotic map and column-wise operation. Multiple exclusive-or operations on sensing data utilized Salsa20 and the 5D chaotic Map. The DNA procedure produces encrypted data for MQTT transmission to the healthcare canter from a hash cipher SHA3-256 Algorithm. The authors created an IoT health care monitoring and tracking system using the hybrid cipher approach. Encryption stages:

***Create Secret Key***

5-D chaotic maps provide secret keys that are altered each time using a random shift to change the sequences throughout time for variables x, y, z, and k, which are started with zero position (initial position) using eq. 1, 2, 3, 4, and 5. In the suggested method, 32-key with 64-bit length is the required number in each sequence before applying the equations to get the request numbers. The fraction part's fixed size is used to convert the resulting number to decimal.

***Upgraded Lightweight Salsa20***

The proposed approach generates a 512-bit keystream matrix 4×4 (16 words) using a modified Salsa20. In each sensing data encryption state, a proposed Salsa20 is utilized twice: once for exclusive-or operation with sensing data and a generated key, and once for DNA operation by randomly selecting two words that convert to DNA.

***Initial Encryption***

The first stage receives sensor data, a 5D chaotic map, and Modified Salsa20. In this stage, two procedures are applied for thirty-two rounds: if the round index is even, the exclusive-or operation is applied between sensing data, 5D chaotic Map (xi, yi, zi, ki, and pi), and a modified Salsa20 words; if odd, it is applied between sensing data and subset key from 5D chaotic Map.

***DNA Operation (Second Encryption)***

The suggested system converts previous step output into DNA for the DNA operation. Next, two random modified Salsa20 words are transferred to DNA. After applying the additional operation to the two preceding strings, the resultant DNA strand is transformed back to decimal form.

**Data Sensors**

The proposed technique uses SHA3-256 to protect networked data. The data sensor was padded with "1" and "0" to 1088 bit (r). Sponge construction initializes the State variable s of b = r + c bit to zero and modifies it each iteration. Absorbing and squeezing make sponges. In the absorption phase, the input block is padded with zeroes to increase its length from r (1088 bits) to b (1600 bits). 4.6 Suggested Decryption The healthcare server that receives hash data, encrypted data, random numbers for shift key and selected Salsa20-word uses this component of the suggested technique. It separates them first. Next, reverse the encryption operation. DNA removal replaces DNA addition in encryption. Selected words from created Salsa20 with encrypted data (receiving data) employ this operation. The same 5D chaotic Map and its shift were used to get the same values at the encryption stage for exclusive-or operation to retrieve the precise value of original data. Finally, the decrypted data is hashed with SHA3-256 and compared to the received hash data to verify its integrity before being divided and put in the patient database.

1. **EXPERIMENT AND RESULTS**

The result of an experiment testing the implementation of the suggested Internet of Things system for the health care system's sensing and transceiver signals. In the first step of the process, the data sensing is hashed using the hash function after being concatenated, encrypted, and then encrypted again using the proposed approach. The health care server receives the encrypted data together with the hashed data along with specific parameters in order to perform an analysis and identify any abnormalities in the state of the data. The proposed solution underwent evaluations to see how well it performed and how efficiently it used IoT resources.

**Table 3**: presents the sensing data samples collected from women every 2.0 seconds.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Two Stage Sensing data** | **No** | **O2 Sensor** | **HR Sensor** | **TEMP Sensor** | **Combine Data** |
| **1st Sensing data** | 1 | 95 | 51 | 30 | '09505131' |
| 2 | 96 | 130 | 29 | '09613029' |
| 3 | 99 | 105 | 33 | '09910533' |
| 4 | 99 | 37 | 29 | '09903729' |
| 5 | 99 | 80 | 24 | '09908024' |
| **2nd Sensing data** | 1 | 97 | 95 | 29 | '09709529' |
| 2 | 96 | 68 | 30 | '09606830' |
| 3 | 98 | 44 | 33 | '09801433' |
| 4 | 99 | 37 | 30 | '09903730' |
| 5 | 95 | 55 | 29 | '09505529' |

Table 3 shows how numerous people's first and second sensing data are collected and integrated in one sequence. The suggested method uses two random numbers for each sensory data. The first number shifts the produced key (5D chaotic Map) to make it dynamic. The second one selects two modified Salsa20 words for security. Table 4 shows these values for each.

**Table 4:** The salsa method is used to spice up the random samples.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **#** | **Randomkeys**  **Shift** | **Randomword**  **selection** |  | **#** | **Randomkeys**  **Shift** | **Randomword**  **selection** |
| **1st Sensing data** | 1 | 16 | 12 | **2nd Sensing data** | 1 | 55 | 5 |
| 2 | 20 | 15 | 2 | 29 | 3 |
| 3 | 44 | 8 | 3 | 19 | 8 |
| 4 | 0 | 7 | 4 | 21 | 14 |
| 5 | 10 | 15 | 5 | 16 | 12 |

The first encryption stage exclusive-ors sensing data with created keys (5D chaotic Map) and adjusted Salsa20 output (Table 5).

Table 5: Salsa add round number samples.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **#** | **1st encryption stage result** |  | **#** | **2nd encryption stage result** |
| **1st Sensing data** | 1 | 66989306389507768 | **2nd Sensing data** | 1 | 65367094928434939 |
| 2 | 28333940693335873 | 2 | 14800772731867072 |
| 3 | 64255463993093324 | 3 | 1603151064451174 |
| 4 | 43996289509461942 | 4 | 40107830565257632 |
| 5 | 43925851781535147 | 5 | 56032449953398644 |

In the second stage, the output of the first stage is added to two random modified Salsa20 words in DNA form. Table 6 shows a DNA sample.

**Table 6**: Algorithm DNA samples.

|  |  |  |
| --- | --- | --- |
|  | **DNASequence** | |
| **1st Sensing data** | 1 | 'ATTAGCGTTCTTCCTGCGTTCTTGGATTGATC' |
| 2 | 'TGCTTGTAGTGTATAAACAAGCAATGTAGGAA' |
| 3 | 'ATTAATGTCTCCTTGGATAGGTCCGAAGGTAT' |
| 4 | 'GACTGCACGGCCAGAGTGACGTGAAGCATTAC' |
| 5 | 'TGCTCCAACCTCGTCCGTACAGTCAACTACGG' |
| **2nd Sensing data** | 1 | 'CTCGGTTGCCTCATTGGTTACGATAAAATCCT' |
| 2 | 'ATGCGCTTTCGGCCCTCCGAGGTGTCCCGTGG' |
| 3 | 'TGCTGACCGGACTGTTAACGCTTTTCCGGCCT' |
| 4 | 'ATTACCCACCATCGATTTTCCCCGTTGTAACT' |
| 5 | 'TCACGCTCTCTGTTCATTGGAAAGCTCCGTCG' |

The previous action is converted to decimal for merging using the hash function. Table 7 shows encryption results.

**Table 7:** The encrypted data samples

|  |  |  |
| --- | --- | --- |
|  | No. | EncryptedData |
| **1st Sensing data** | 1 | '6584846575677184846784846767847667718484678484717165848471658467' |
| 2 | '8471678484717465718471846584656565676565716766658471846571716565' |
| 3 | '6584846565847184678467718484717165846571718467677165657167846584' |
| 4 | '7165678471656567717167676571657184716567718484656571676584846567' |
| 5 | '8471678467676565656784677184676771846567657184676565678465677171' |
| **2nd Sensing data** | 1 | '6784676771848471676784676584847171848465677165846565656584676784' |
| 2 | '6584716771678484846571716767678467677165717184718467676771847171' |
| 3 | '8471678471658467717165678471848465656771678484848467717171676784' |
| 4 | '6584846567676765676765846771658465848467676767718484718465657184' |
| 5 | '8467656771678467846784718484676571846571656565716784676771846771' |

As shown in table 8, sensing data is hashed and blended with encrypted data before being delivered to the server to improve data integrity.

**Table 8:** Female data hashing samples every 2 seconds.

|  |  |  |
| --- | --- | --- |
|  | **No.** | **HashingResult** |
| **1st Sensing data** | 1 | '2289342d1f698cf61104adbC60e27624bef8741ade45d1bdc9a469788e3f6c20' |
| 2 | 'b5af682f3a6883ed2f3C71ff9043e4ed31bacb9c171039767725277f8b666799' |
| 3 | '207d3f1fecc915999b2fbd45a547421f16b8d77bf1f4d5cf255b04a834c735f4' |
| 4 | '91781b000fa280e65470b4578b3290ad396b9165bd8df217100ec12c78e67403' |
| 5 | '9d936a5833721ab23da19207bee8ab64895b3d48fe348ab578b57d720295e736' |
| **2nd Sensing data** | 1 | '9d5893f0bdc974e3f5dfb46c1048c7cc24579ea4b88994305d5a5bc06b65e847' |
| 2 | '8767780198d264f6aa1b7c5d20a4038c2a93acfd22b22d2141ba0a0bbe799c26' |
| 3 | '3cc20d54a7fe2eb3e747a2b4c18a0379d24fe070001d41a7278b571a946890fb' |
| 4 | 'a211883a8a5a4b4234bd5d6d7b8c89db9dd3049bf641487fcba02801982672bd' |
| 5 | '4268796acd9aeb58b6d82683751ead3acd8ae3e421644de3fb8f991ea3dce337' |

The NIST randomness test is used on the suggested method's final result. Table 9 shows the NIST tests applied to the proposed Algorithm with varied bit widths to confirm security and resistance to various assaults. The p-value from testing a stream of encryption bits passes all NIST tests for the proposed algorithm.

**Table9:** NIST Algorithm Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Test** | **status** | **P-Value** |
| 1 | TrackTest | Successfully complete | 0.61459 |
| 2 | SequentialTest | Successfully complete | 0.87405 |
| 3 | randomtripvarianttest | Successfully complete | 0.83664 |
| 4 | randomexcursiontest | Successfully complete | 0.73521 |
| 5 | Overlyingtemplatematchingtest | Successfully complete | 0.99001 |
| 6 | Non-overlappingpatternmatchingtest | Successfully complete | 0.98761 |
| 7 | FrequencyMonobitTest | Successfully complete | 0.99648 |
| 8 | Maurer’s universal statistical test | Successfully complete | 0.98559 |
| 9 | longest block test run | Successfully complete | 0.99257 |
| 10 | Complexity Test | Successfully complete | 0.62794 |
| 11 | Block-frequency test | Successfully complete | 0.88549 |
| 12 | Discrete Fourier Transform test | Successfully complete | 0.65629 |
| 13 | Cumulative sums Test | Successfully complete | 0.98701 |
| 14 | estimated entropy test | Successfully complete | 0.96589 |
| 15 | binary matrix rank test | Successfully complete | 0.98999 |

1. **CONCLUSION**

For secret key generation, the proposed unique method combines three cypher algorithms (lightweight Salsa20, Addition Round, and DNA) plus a Five-domination chaotic map. Several experiments, such as the NIST test, were used to evaluate the suggested technique on the proposed Algorithm. The suggested technique includes data security aspects such as secrecy, authentication, and non-repudiation. It works with all sensors to securely transmit data to the final administrator. The patient data is maintained in the health cloud, and medical practitioners can access it at any time to monitor the patient's progress. Furthermore, IoT networks necessitate a limited amount of computational capability, memory, and power. As a result, they require a lightweight security solution that consumes few resources. The approach is lightweight and can be utilized in Internet of Things (IoT) systems, Wireless sensor networks (WSN), and any other application that requires a low complexity and low computation encryption algorithm.

**Reference**

1. R. R. Chaudhary, and K. Chatterjee, "An Efficient Lightweight Cryptographic Technique for IoT based Healthcare System," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 991–995, 2020.
2. H. Noura, R. Couturier, C. Pham, and A. Chehab, "Lightweight stream cipher scheme for resource constrained IoT devices," In the 2019 International Conference on Wireless and Mobile Computing, Networking and Networking and Communications (WiMob), pp. 1-8, IEEE,2019.
3. F. Andriopoulou, T. Dagiuklas, and T. Orphanides’s, "Integrating IoT and fog computing for healthcare service delivery," In Components and services for IoT platforms, pp. 213-232, Springer, Cham, 2017.
4. S. Amanlou and K.A.A. Bakar, "Lightweight security mechanism over MQTT protocol for IoT devices," International Journal of Advanced Computer Science and Applications, Vol.11, No.7,2020.
5. S.Rajesh, V. Paul, V. G.Menon, and M.R. Khosravi, "A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices," Symmetry, vol.11, no.2, pp. 293,2019 .
6. A. Vithya Vijayalakshmi and L. Arockiam, "Hybrid security techniques to protect sensitive data in E-healthcare systems," Proceedings of the International Conference on Smart Systems and Inventive Technology, ICSSIT 2018. pp. 39–43, 2018.
7. H. Tao, M. Z. A. Bhuiyan, A. N. Abdalla, M. M. Hassan, J. M. Zain, and T. Hayajneh, "Secured Data Collection with Hardware-Based Ciphers for IoT-Based Healthcare," IEEE Internet Things J., vol. 6, no. 1, pp. 410–420, 2019.
8. Naveen, R. K. Sharma, and A. R. Nair, "IoT-based Secure Healthcare Monitoring System," Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019. 2019.
9. S. Joshi and S. Joshi, "A sensor-based secured health monitoring and alert technique using IoT," In 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 152-156, IEEE, 2019.
10. K. M. Besher, Z. Subah, and M. Z. Ali, "IoT Sensor Initiated Healthcare Data Security," IEEE Sens. J., vol. 21, no. 10, pp. 11977–11982, 2020.
11. 11. D. Balakrishn, T. DhiliphanRajkumart, and S. Dhanasekaran, "An intelligent and secured heart rate monitoring system using IOT," Materials Today: Proceedings, 2020.
12. 12. C. Pandey, S. Sharma, and P. Matta, "DME: An Efficient Encryption Technique for Body Sensor Network in Healthcare Internet of Things (H IoT)," in Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021, vol. 2021, pp. 80–85,2021.
13. 13. A.O. Khadidos, S. Shidharth, A.O. Khadidos, K. Sangeetha, and K.H. Alyoubi, "Healthcare Data Security Using IoT Sensors Based on Random Hashing Mechanism," Journal of Sensors, vol.2022.pp,17,2022.
14. 14. M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, "Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities," IEEE Internet of Things Journal, vol. 8, no. 13. pp. 10474–10498, 2021.
15. 15. Y. Sun, F. P. W. Lo, and B. Lo, "Security and Privacy for the Internet of Medical Things Enabled Healthcare Systems: A Survey," IEEE Access, vol. 7. pp. 183339–183355, 2019.
16. D. Liestyowati, "Public key cryptography," in Journal of Physics: Conference Series, 2020.
17. H. Habibzadeh, K. Dinesh, O. RajabiShishvan, A. BoggioDandry, G. Sharma, and T. Soyata, "A Survey of Healthcare Internet of Things (HIoT): A Clinical Perspective," IEEE Internet of Things Journal, vol. 7, no. 1. pp. 53–71, 2020.
18. 18. J. Wang et al., "A logistic mapping-based encryption scheme for Wireless Body Area Networks," Futur. Gener. Comput. Syst., vol. 110, pp. 57–67, 2020.
19. 19. Z. M. J. Kubba and H. K. Hoomod, "Modified PRESENT Encryption algorithm based on new 5D Chaotic system," in IOP Conference Series: Materials Science and Engineering, vol. 928, no. 3, pp. 032023, 2020.

[20] L. Ding, "Improved Related-Cipher Attack on Salsa20 Stream Cipher," IEEE Access, vol. 7, pp. 30197–30202, 2019.

1. 21. A.H.Fadel, R.S. Hameed, J.N. Hasoon, S.A. Mostafa, and B.A.Khalaf, "A lightweight ESalsa20 Ciphering based on 1D logistic and chebyshev chaotic maps," Solid State Technology, vol.63, no.1, pp.1078-1093,2020
2. 22. D.Afdhila, S.M. Nasution, and F. Azmi, "Implementation of stream cipher Salsa20 algorithm to secure voice on push to talk application," In 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), pp. 137-141, ISBN 978-1- 5090-2767-5,2016.
3. 23. S. Dey and S. Sarkar, "Improved analysis for reduced round Salsa and Chacha," Discrete Applied Mathematics, vol.227, pp.58-69,2017.
4. 24. R. Ahmed andI.J. Mohammed, "Developing a new hybrid cipher algorithm using DNA and RC4," International Journal of Advanced Computer Science and Applications, vol.8, no.10, pp. 034-039,2017.
5. 25. E.R. Soni, E. V., Soni, and E.S.K. Mathariya, "Innovative field of cryptography: DNA cryptography," In International Conference on Information Technology Convergence and Services,2012.
6. 26. A. Sideris, T.Sanida, and M.Dasygenis, "High throughput implementation of the keccak hash function using the niosII processor," Technologies, vol.8, no.1, pp. 15,2020.
7. 27. T. Zhou, Y. Zhu, N. Jing, T. Nan, W. Li, and B. Peng, "Reliable SoC Design and Implementation of SHA-3-HMAC Algorithm with Attack Protection," in Proceedings - 2020 IEEE International Conference on Smart Cloud, Smart Cloud 2020, vol.2020 pp. 88–93,2020.
8. 28. Y. Yang, D. He, N. Kumar, and S. Zeadally, "Compact Hardware Implementation of an SHA-3 Core for Wireless Body Sensor Networks," IEEE Access, vol. 6, pp. 40128– 40136, 2018.
9. 29. Z. Ashfaq et al., "A review of enabling technologies for Internet of Medical Things (IoMT) Ecosystem," Ain Shams Engineering Journal, vol. 13, no. 4. 2022.
10. 30. A. Ruiz, P. Cordova, and C. Gordon, "Telemedicine System to Avoid Sudden Death Syndrome by Continuous Monitoring of Vital Signs," 2018 5th International Conference on democracy and eGovernment, ICEDEG 2018. pp. 212–217, 2018.
11. 31. S. Costanzo and A. Flores, "A non-contact integrated body ambient temperature sensors platform to contrast COVID19," Electron., vol. 9, no. 10, 2020.
12. A. Radici, Y. Q. Chen, F. Padula, and A. Visioli, "A Laboratory Setup for an Introduction to Fractional Order Systems," in IFAC-Papers Online, vol. 52, no. 9, pp. 74–78,2019