RESEARCH ARTICLE

An Efficient and Secure Text Encryption Scheme for Wireless Sensor Network (WSN) Using Dynamic Key Approach

V. Elamurugu
Department of Computer Science, Rajah Serfoji Govt. College (Autonomous), Affiliated to Bharathidasan University, Thanjavur, Tamil Nadu, India
dr.parthi2009@gmail.com

D. J. Evanjaline
Department of Computer Science, Rajah Serfoji Govt. College (Autonomous), Affiliated to Bharathidasan University, Thanjavur, Tamil Nadu, India
evanjalinedj@gmail.com

Received: 25 October 2021 / Revised: 14 December 2021 / Accepted: 19 December 2021 / Published: 30 December 2021

Abstract – In a wireless sensor network (WSN), all detected data is delivered via a wireless communication channel to a sink node, then sent to an information-gathering centre for necessary actions or controls. The sensed data could be readily manipulated or eavesdropped on if security procedures are not used. For WSN, several security solutions based on classical cryptography have been devised, although the sophisticated encryption operations take significant energy. The symmetric and asymmetric key encryption provides efficient data security, but it takes high energy consumption and computational complexity. In this paper, a lightweight, energy-efficient secure text encryption is proposed using the dynamic salt key. There are three primary processes in the suggested paradigm. The first is salt generation. The next step is to encrypt secret text using format-preserving encryption based on the salt key, and the final step is to decrypt the data. The encryption process is more secure, and the hackers cannot capture key values. The proposed approach creates a safe environment for sensors to protect the data quickly, efficiently, and low-computation before sending it across a wireless network to the sink node. The proposed method simulation provides a high level of security while requiring minimal communication and computational resources.

Index Terms – Wireless Sensor Network, Lightweight Encryption, Dynamic Encryption, Salt Algorithm, Format-Preserving Encryption, Security, Dynamic Key.

1. INTRODUCTION

A wireless sensor network (WSN) contains sensor nodes that collect data from their surroundings in real-time. WSNs offer a dependable and efficient data collection mechanism. On the other hand, WSN comprises limited power, memory, computing capability, transmission length, and lifespan resources [1]. The collaboration of various devices to perceive, manage, and formulate decisions depending on environmental variables is one of the primary aspects of WSN. The involved devices may fulfil various functions such as sense, direction-finding, access, and sink to provide the necessary functionality. Mutual processing, disseminated computing, information mixture, information aggregation, remote examination and management, and observation are all features provided by WSN. Particular applications focus on one or more of these characteristics [2].

Security is a difficult concern in WSNs because sensor networks are typically installed in aggressive situations. Furthermore, few memories, weak processors, and less power of sensor nodes pose several challenges for typical cryptographic algorithms to be implemented in sensor networks [3]. As a result, WSNs necessitate innovative encryption algorithms regarding operating speed, capacity, and energy usage.

To ensure the success of WSN's applications, security is essential. For instance, when a sensor network is utilized for military purposes, it is critical to maintaining the information sensed in private and authentic [4]. Because many known security approaches for traditional networks do not apply to WSN, providing security for WSN represents a wide field of research concerns. For example, WSN necessitates lightweight security techniques to minimize security-related overhead and network performance is not affected.

Cryptography is one of the most prevalent security solutions. The key is disseminated symmetrically or asymmetrically between the nodes [5]. The unequal distribution of keys necessitates a higher cost and slower speed. Key distribution is hard in symmetric key distribution because the key must be transmitted before the message [6]. These algorithms ensure a high level of protection. Simulation results and respective visio-statistical assessment can accomplish good and reliable secure data communication over the cloud environment [7]. However, because of the large computations required by these techniques, they cannot be supported efficiently by sensors' limited processing and storage capacities [8]. This necessitates the development of secure encryption methods, require fewer calculations, and are adaptable.

The main problem of previous encryption methods is high computional and complex operation, leading to more energy consumption of sensor nodes [9]. This study presents lightweight, energy-efficient, safe text encryption using a dynamic salt key to overcome these problems. There are three primary processes in the suggested paradigm. The first is salt generation. The next step is to encrypt secret text using format-preserving encryption based on the salt key, and the final step is to decrypt the text. The main objective of this work is to reduce energy consumption and computational complexity. The simulation shows that the proposed work reduces energy usage and time complexity compared to the AES algorithm.

The rest of the paper is organized as follows: Section 2 reviews various encryption methods used in WSN. Then, the preliminary concepts are described in section 3. Next, the proposed dynamic salt-based encryption scheme is explained in section 4, and Section 5 analyzes the performance of the proposed work. Finally, section 6 concluded the paper.

2. RELATED WORK

This section explains different encryption methods used in WSN and dynamic key management in WSN.

Tsai et al. [10] present a lightweight, secure encryption technique for WSN called LED. Before data is delivered to the sink node through a wireless network, this method offers a stable situation for sensors to encrypt information using an uncomplicated, safe and less computation mechanism. However, in this method, collision search vulnerabilities occur.

The Klein block cypher is a novel lightweight encryption technique suggested by Ghorashi et al. [11]. The Klein block cypher methods are examined for performance and security improvements. In addition, a new technique based on a 3-layer substitution box has been presented to reduce resource consumption while maintaining security. However, this method is inefficient, and it takes a high rate for the encryption process.

The hamming residue approach is proposed by Alotaibi et al. [6] for offering protection to WSN. Each node in the WSN generates a new security codeword, improving network security and making it easier to identify rival nodes. In addition, this approach protects WSNs from malicious attacks without the need for a key distribution method.

With consensus mechanism and asymmetric signature method, Feng et al. [12] present a blockchain-based disseminated collocation storage design for the information protection processing platform of WSN. Pournaghi et al. [13] present a new, safe, and efficient healthcare data recording and storage strategy based on blockchain and attribute-based encryption. This method safeguards confidentiality and offers fine-grained access control of healthcare information based on common information guard regulations. Furthermore, this solution employs private blockchains to increase the ability to cancel quick access, which is one of the attribute-based encryption problems.

To achieve key management progress in WSN, Sun et al. [14] present a local dynamic system based on layer-cluster topological design. The sensor nodes in this approach create a cluster based on their location and select cluster head nodes using a self-selection method. The gateway node accomplishes the majority of the calculations in the key agreement process using the Chinese Remainder Theorem. In contrast, the head-cluster nodes use a one-way hash tree to determine the group key.

For heterogeneous WSNs, Athmani et al. [15] present an effective dynamic authentication and key management approach. The major goal is to provide a single lightweight protocol that can be used for authentication and key creation while maintaining a high level of security. The key distribution method generates dynamic keys using prior knowledge and does not need a secure communication or exchange process, which increases security, energy efficiency, and memory usage.

Mesmoudi et al. [16] present an efficient and dynamic key management approach for hierarchical WSN. This approach has three submethods for establishing keys, renewing keys, and integrating new nodes. Finally, Elhoseny et al. [17] present a novel encryption approach for information broadcast security in WSNs with dynamic sensor clusters. The Elliptic Curve Cryptography (ECC) creates binary strings for each sensor, which are then combined with node ID, cluster head distance, and transmission round index to form unique keys. Encryption and decryption are accomplished quickly using XOR, replacement, and alternative procedures.

Elhoseny et al. [18] present a unique encryption method based on ECC and homomorphic encryption. The key is created by uniting the ECC key, node identification number, and cluster head distance (CH). Homomorphic encryption is used to lower CH's energy usage by allowing it to combine encoded information without decoding it.

Khashan et al. [19] offer a lightweight cryptographic technique for WSNs that is automated. A new dynamic clustering method that facilitates sensor node mobility is created in this scheme. Furthermore, it proposes a flexible, lightweight cryptographic method for managing encryption difficulty by automating the selection of encryption settings depending on the currently available resources of each sensor node.

For WSN, Nanda et al. [20] offer a hybrid encryption approach. It employs both symmetric and asymmetric key encryption techniques. Combining the two encryption algorithms can enhance the network's efficiency, resulting in smaller sent data, lower computing overhead, and faster decryption cycles. However, it lacks a key management plan and hence cannot address internal risks and attacks effectively.

Security features on the WSN, on the other hand, not only degrade overall effectiveness but also raise power/energy usage. Security, performance, space, throughput, power usage, and energy consumption are all factors to consider. As a result, this study proposes a data encryption mechanism that is simple, secure and consumes little memory and power.

3. PRELIMINARIES

3.1. Salt

Salt [21] is an arbitrary string of letters, numeric digits, or special characters appended to a password so that the hash values maintain maximum randomness after hashing. In storage space systems, it's best to use distinctive salt values for each password. Salt is added to the string does not assure complete secrecy, but it does ensure that cracking the password is computationally impossible [22]. The power of salted hash values maybe that still brute force threat would take a long time to decrypt the hash. Since salts are confidential, using one public and one private will guard a password from offline password guessing threats. [23].

3.2. Format Preserving Encryption

When using Format Preserving Encryption (FPE), the input domains' patterns are preserved throughout encryption. Tweak t and domain d are two extra parameters in FPE algorithms. Tweak t provides additional randomness in the same way as cryptographic salt does. Domain d is the domain in which the simple text is found. IPv4 addresses or credit card numbers are examples of input domains.

The following are the basic algorithms of an FPE method [23].

KeyGen(σ): Based on the security parameter σ, the key and tweak tk is generated.

Encryption(pt, key, tk, d): Given a plaintext pt, key and a tweak tk, the method generates ciphertext ct such that both the pt and ct are in the same domain d. It internally uses a rank and de-rank process along with a length preserving block cipher enc '.

Decryption(ct, key, tk, d): Given a ciphertext ct, key and a tweak tk, the method returns the corresponding plain text pt. Here both ct, pt is in the same domain.

The encoding of datasets with an unstructured form [25], such as Primary Account Numbers not in binary code, is a few application instances for FPE. FPE can also be employed in communications where protocols need to be encrypted, such as in military or industrial settings [26] or when certain picture formats need to be encrypted [27].

4. METHODOLOGY

Because of its application purpose, data secrecy is one of the most important security criteria for WSN. Sensor nodes transmit sensitive data; thus, it's important to ensure no intruders or other networks may intercept the broadcasts and steal valuable information. Encrypting data with a shared key is a common security solution for ensuring data confidentiality. This ensures that only the intended recipients have access to sensitive information. The sophisticated encryption/decryption procedure enhances the cryptographic approach's security, but it uses many computational resources. Sensor nodes in a WSN are typically resource-restricted, meaning they have limited energy, processing power, and memory. In addition, lightweight batteries, which cannot be replaced or recharged, are expected to power sensor nodes. In this instance, a data encryption mechanism that is both secure and low-power is required.

This section explains the proposed lightweight, energy-efficient, secure text encryption using the dynamic salt key. The proposed method contains three phases: Salt Key Generation, Data Encryption and Data Decryption.

4.1. Salt Key Generation

Algorithm-1 explains the salt key generation

```
Step1: Assign base= 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Step2: s1 = randomly select 8 characters in base
Step3: For i = 1 to length(s1)
Step4:     a1 = ascii(s1[i])
Step5:     a2 = hex(a1)
Step6:     salt[i] = a2;
Step7: End For
Step8: Return salt
```

Algorithm 1 Salt Key Generation

In this algorithm 1, the combination of numbers and alphabets are assigned to the base string (step1). Then, the initial salt value is generated based on the base string, which contains an 8-bit length (step2). Then, for each character in the initial salt value, get the ASCII value (step4) and convert it to hexadecimal (step5). Finally, the generated salt value is used for data encryption.

4.2. Data Encryption

This section explains the proposed data encryption process. Algorithm-2 explain the data encryption process. Initially, the plain text (secret text) is converted into matrix format. Then, the generated salt key is appended with the matrix. Finally, using format-preserving encryption, the generated matrix is encrypted. The decryption process is the reverse of the encryption scheme.

```
Input: Plain Text (PT)
Output: Encrypted Text (ET)
Step1: Get the secret plain text (PT) of 16-bit length
Step2: Convert PT into matrix (MAT1) format based on (1)
Step3:Generate salt key (SK) using Algorithm-1
Step4: MAT2 = Add SK into MAT based on (2)
Step5: Convert MAT2 into Message (SM)
Step6: binSM = binary format(SM)
Step7: ET = FPE(binSM)
Step8: Return ET
```

Algorithm 2 Data Encryption

Figure 1 shows the process of encryption workflow.

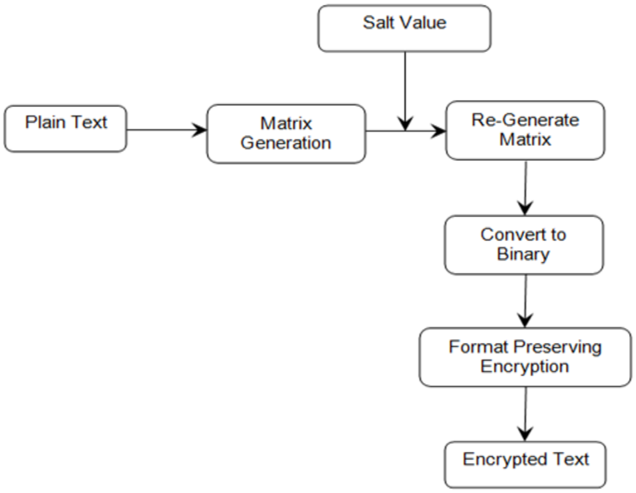

Figure 1 Encryption Workflow

The length of the plain text is must be 16 bits. Let msg ={m_1,m_2,m_3,…m_{16}} be the secret message. Convert secret message into matrix format.

$$sm1 = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \\ m_9 & m_{10} & m_{11} & m_{12} \\ m_{13} & m_{14} & m_{15} & m_{16} \end{bmatrix} \qquad (1)$$

Generate salt key with 16-bit length. Let saltkey={k_1, k_2, k_3,…,k_{16}}. Append salt to sm,

$$sm2 = \begin{bmatrix} X_0 & k_1 & k_2 & k_3 & k_4 & X_0 \\ k_5 & m_1 & m_2 & m_3 & m_4 & k_6 \\ k_7 & m_5 & m_6 & m_7 & m_8 & k_8 \\ k_9 & m_9 & m_{10} & m_{11} & m_{12} & k_{10} \\ k_{11} & m_{13} & m_{14} & m_{15} & m_{16} & k_{12} \\ X_0 & k_{13} & k_{14} & k_{15} & k_{16} & X_0 \end{bmatrix} \qquad (2)$$

Convert matrix sm2 into message sc={X_0, k_1, k_2, k_3, k_4, X_0, k_5, m_1, m_2, m_3, m_4, k_6, k_7, m_5, m_6, m_7, m_8, k_8, k_9, m_9, m_{10}, m_{11}, m_{12}, k_{10}, k_{11}, m_{13}, m_{14}, m_{15}, m_{16}, k_{12}, X_0, k_{13}, k_{14}, k_{15}, k_{16}, X_0}.

The binary string is generated from sc, and format-preserving encryption is applied to encrypt the text.

4.3. Data Decryption

Figure 2 shows the decryption workflow. First, the format-preserving decryption is applied to decode the encrypted text. The decoded message is a binary message, which is again converted into the string format. Then, the matrix is generated from the decoded message. Finally, the plain text is extracted from the matrix.

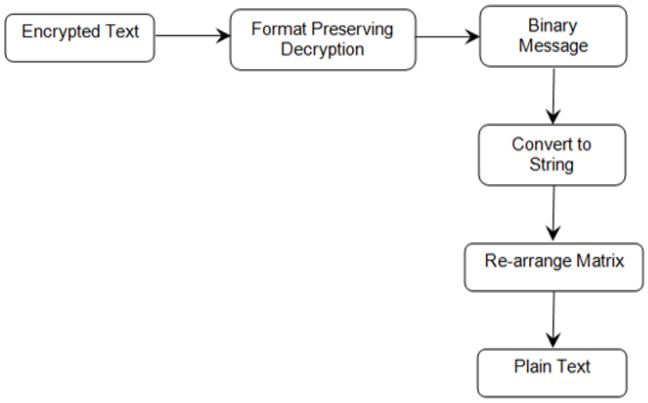

Figure 2 Decryption Workflow

5. EXPERIMENTAL RESULTS

This section evaluates the performance of the proposed encryption method. The proposed secure data encryption is implemented using Java (version 1.8), and the experiments are performed on an Intel(R) Pentium(R) processor with a speed of 2.30 GHz and 4.0 GB RAM using Windows 10 64-bit Operating System.

Working Example

Plain Text = Secret Message

The salt value can be generated based on the combination of numbers (0-9) and alphabets (a-z and A-Z).

Initial Salt = g K 9 U z F z T

Convert initial salt to ascii value

Ascii Value = 103 75 57 85 122 70 122 84

Convert ascii value to hexa decimal value

Hexa Decimal = 67 4b 39 55 7a 46 7a 54

Salt Key = 674b39557a467a54

Salt Hash = 71455869c06fd67bfa96bd6e79ea2d0d

Salt Key Generation

```
String base =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";

int saltsize=8;

Random random = new Random();

StringBuilder b = new StringBuilder();

for(int i = 0; i < saltsize; i++)

{

    b.append(base.charAt(random.nextInt(base.length())));

}

String s1= b.toString();

char ch[]=s1.toCharArray();

for(int i=0;i<ch.length;i++)

{

     int a1=(int)ch[i];

     String a2= Integer.toHexString(a1);

      salt=salt+a2;

}

String salt key = salt;
```

Convert Plain Text to Matrix

$$sm1 = \begin{bmatrix} S & e & c & r \\ e & t & & M \\ e & s & s & a \\ g & e & & \end{bmatrix}$$

Add salt key into Matrix

$$sm2 = \begin{bmatrix} - & 6 & 7 & 4 & b & - \\ 3 & S & e & c & r & 9 \\ 5 & e & t & & M & 5 \\ 7 & e & s & s & a & a \\ 4 & g & e & & & 6 \\ - & 7 & a & 5 & 4 & - \end{bmatrix}$$

Convert Matrix into Message

Message = -674b-3Secr95et M57essaa4ge006-7a54-

Convert Message into Binary value

```
00101101   00110110   00110111   00110100   01100010
00101101   00110011   01010011   01100101   01100011
01110010   00111001   00110101   01100101   01110100
00100000   01001101   00110101   00110111   01100101
01110011   01110011   01100001   01100001   00110100
01100111   01100101   00110000   00110000   00110110
00101101   00110111   01100001   00110101   00110100
00101101
```

Apply format preserving encryption for binary string

The encrypted message is

```
15498755   34374776   62015158   75812233   10840413
15498755   86006997   30458948   90415726   90396417
44479778   21976178   24775954   90415726   64612567
83526362   13719839   24775954   62015158   90415726
27055839   27055839   19327447   19327447   75812233
22766089   90415726   00703397   00703397   34374776
15498755   62015158   19327447   24775954   75812233
15498755
```

The decryption process is reverse of the encryption scheme.

The performance of the proposed work is evaluated in terms of execution time and memory consumption. The proposed algorithm is compared with the AES algorithm.

Table 1 shows the comparison of execution time for AES and proposed work.

Algorithm	**Encryption Time (ms)**	**Decryption Time (ms)**
AES	1066	45
Proposed	841	37

Table 1 Execution Time Comparison

Figure 3 shows the encryption and decryption time comparison for the AES and the proposed work. The result shows that the proposed algorithm has less time to take encrypt and decrypt the text.

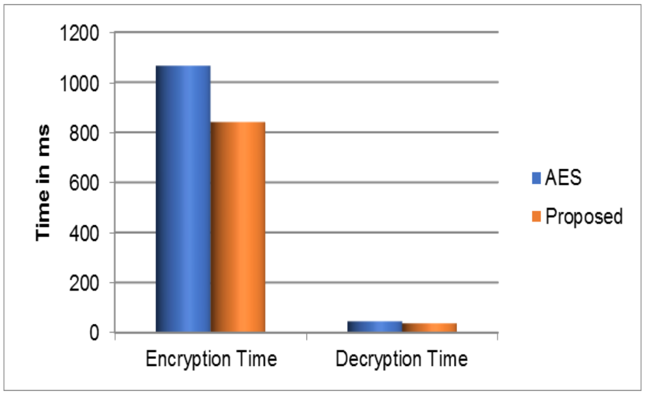

Figure 3 Encryption and Decryption Time Comparison

Table 2 shows the comparison of memory in bytes for the encryption and decryption process.

Algorithm	Encryption	Decryption
AES	13178	14120
Proposed	8307	8996

Table 2 Memory Consumption in Bytes

Figure 4 shows the memory consumption comparison for AES and the proposed algorithm. The proposed work needs 8307 bytes for encryption and 8996 bytes for the decryption process. Therefore, the proposed method saves 58.63% memory consumption for the encryption process and saves 56.95% for the decryption process compared with AES.

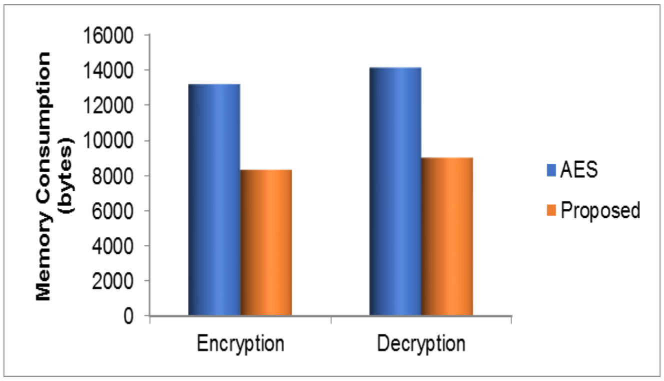

Figure 4 Encryption and Decryption Memory Consumption

6. CONCLUSION

WSN is one of the most helpful technology with a wide range of uses. WSN's primary challenges are security and privacy. To offer an acceptable level of protection, symmetric and asymmetric encryption techniques can be used. However, in a WSN, these algorithms will not be the best option because they involve complex operations that require high processing capabilities that the sensors lack. Because of the limited context of WSNs, flexible and lightweight encryption techniques are required. This paper proposes a secure and efficient data encryption method for WSN. The proposed method uses the salt value as a key for encryption. The format-preserving encryption model is used to encrypt the message. The proposed method performance was evaluated, which proves that the proposed encryption method is lightweight and secure.

REFERENCES

[1] H. Hayouni, M.A. Hamdi, "A novel energy-efficient encryption algorithm for secure data in WSNs", Journal of Supercomputing, vol. 77, pp. 4754–4777, 2021

[2] S.A. Bragadeesh and A. Umamakeswari, "Secure Data Aggregation for Wireless Sensor Network using Lightweight Cryptography", Indian Journal of science and technology, vol. 9, no. 48, 2016

[3] K. Biswas, V. Muthukkumarasamy, E. Sithirasenan, K. Singh, "A Simple Lightweight Encryption Scheme for Wireless Sensor Networks", International Conference on Distributed Computing and Networking, pp. 499-504, 2014

[4] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu and J. Liu, "A Novel Block Encryption Algorithm Based on Chaotic S-Box for Wireless Sensor Network," in IEEE Access, vol. 7, pp. 53079-53090, 2019

[5] S. Rajesh, V. Paul, V.G. Menon, 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, 2019

[6] M. Alotaibi, "Security to wireless sensor networks against malicious attacks using Hamming residue method", Journal of Wireless Communications and Networking, vol. 8, 2019

[7] Denis R, Madhubala P, "Evolutionary Computing Assisted Visually-Imperceptible Hybrid Cryptography and Steganography Model for Secure Data Communication over Cloud Environment", International Journal of Computer Networks and Applications (IJCNA), 7(6), PP: 208 - 230, 2020, DOI: 10.22247/ijcna/2020/205321.

[8] H. Tawalbeh, S. Hashish, L. Tawalbeh, A. Aldairi, "Security in Wireless Sensor Networks Using Lightweight Cryptography", Journal of Information Assurance and Security, vol. 12, pp. 118-123, 2017

[9] T. A. Alghamdi, "Secure and Energy-Efficient Path Optimization Technique in Wireless Sensor Networks Using DH Method," in IEEE Access, vol. 6, pp. 53576-53582, 2018

[10] K.L. Tsai, F.Y. Leu, T.H. Su, and Y.C. Chang, "A Light Weight Data Encryption Method for WSN Communication", International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 788-795, 2018

[11] S.R. Ghorashi, T. Zia and Y. Jiang, "Optimisation of Lightweight Klein Encryption Algorithm With 3 S-box," 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 1-5, 2020

[12] L. Feng, H. Zhang, L. Lou and Y. Chen, "A Blockchain-Based Collocation Storage Architecture for Data Security Process Platform of WSN," 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)), pp. 75-80, 2018

[13] S.M. Pournaghi, M. Bayat, and Y. Farjami, "MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption", J Ambient Intell Human Comput, vol. 11, pp. 4613–4641, 2020

[14] B. Sun, Q. Li, and B. Tian, "Local Dynamic Key Management Scheme Based on Layer-Cluster Topology in WSN", Wireless Pers Commun, vol. 103, pp. 699–714, 2018

[15] S. Athmani, A. Bilami, D.E. Boubiche, "EDAK: An Efficient Dynamic Authentication and Key Management Mechanism for heterogeneous WSNs", Future Generation Computer Systems, vol. 92, pp. 789-799, 2019

[16] S. Mesmoudi,B. Benadda,A. Mesmoudi, "SKWN: Smart and dynamic key management scheme for wireless sensor networks", International Journal of Communication Systems, vol. 32, no. 7, 2019

[17] M. Elhoseny, X. Yuan, H.K. El-Minir, and A.M. Riad, "An energy efficient encryption method for secure dynamic WSN", Security and Communication Networks, vol. 9, no. 13, pp. 2024–2031, 2016

[18] M. Elhoseny, H. Elminir, A. Riad, and X. Yuan, "A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption", Journal of King Saud University-Computer and Information Sciences, vol. 28, no. 3, pp. 262–275., 2016

[19] O.A. Khashan, R. Ahmad, N.M. Khafajah, "An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks", Ad Hoc Networks, vol. 115, 2021

[20] A. Nanda, P. Nanda, X. He, A. Jamdagni, D. Puthal, "A hybrid encryption technique for Secure-GLOR: The adaptive secure routing protocol for dynamic wireless mesh networks", Future Generation Computer Systems, vol. 109, pp. 521-530, 2020

[21] J. Jeong, D. Woo and Y. Cha, "Enhancement of Website Password Security by Using Access Log-based Salt," 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS), pp. 1-3, 2019

[22] N.S. Ali, M.H. Alattar A.A. Farawn, "Anti-continuous collisions user-based unpredictable iterative password salted hash encryption", In Int. J. Internet Technology and Secured Transactions, volume 8, page 619–634, 2018.

[23] P. Chandrashekar, S. Dara and V. N. Muralidhara, "Efficient Format Preserving encrypted databases," 2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1-4, 2015

[24] S. Boonkrong, C. Somboonpattanakit, "Dynamic Salt Generation and Placement for Secure Password Storing", International Journal of Computer Science, vol. 43, no. 1, 2015

[25] B. Cui, B. Zhang, and K. Wang, "A data masking scheme for sensitive big data based on format-preserving encryption", in Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Comput., (EUC), pp. 518–524, 2017

[26] I. Oh, T. Kim, K. Yim, and S.-Y. Lee, "A novel message-preserving scheme with format-preserving encryption for connected cars in multi-access edge computing", Sensors, vol. 19, no. 18, 2019.

[27] H. Sun, H. Luo, and Y. Sun, "Data hiding for ensuring the quality of the host image and the security of the message", IEEE Access, vol. 7, pp. 64767–64777, 2019

Authors

Mrs. V. Elamurugu is doing Ph. D in Computer Science, Department of Computer Science, Rajah serfoji Government College (Autonomous), Tanjore, India. Her area of research is classification using Big Data Analytics and wireless sensor networks. She has presented papers in conferences and seminars in diverse perspectives.

Dr. D. J. Evanjaline is working as an Assistant Professor in the Department of Computer Science, Rajah Serfoji Government College (Autonomous), (Bharathidasan University), Tanjore, TamilNadu. She obtained the Ph. D Degree in Computer Science from Bharathidasan University. She has more than 20 years of teaching experiences. Also she guiding seven research scholars from Bharathidasan University. She has published more than 25 research papers in top most Elsevier and Scopus Indexed Journal.

How to cite this article:

V. Elamurugu, D. J. Evanjaline, “An Efficient and Secure Text Encryption Scheme for Wireless Sensor Network (WSN) Using Dynamic Key Approach”, International Journal of Computer Networks and Applications (IJCNA), 8(6), PP: 788-794, 2021, DOI: 10.22247/ijcna/2021/210726.