ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 9, Issue 9, September 2022

Proficient and Safe Token with Secret and Public Keys Sharing Algorithm for Preventing Cryptographic Key Leakage in Wireless Sensor Network

[1] R.Nesamalar, [2] Dr.K.Ravikumar

[1] Research Scholar Department of Computer Science, Tamil University, Thanjavur, Tamil Nadu, India.
[2] Assistant Professor, Department of Computer Science, Tamil University, Thanjavur, Tamil Nadu, India.
Corresponding Author Email: [1] ranesamalar@gmail.com, [2] ravikasi2008@gmail.com

Abstract*— Sensor devices in the Wireless Sensor Network (WSN) are commonly subjected to various forms of attacks, such as flood attacks, eavesdropping attacks, etc. When an attacker compromises a sensor device, the sensor device's data contents become non-confidential and are grabbed by the attacker, putting the entire network at risk. As a result, to prevent key leaks in WSN networks, this paper proposes a Token with Secret and Public Keys Sharing (TSP-KS) algorithm. In the existence of attackers, cryptography is used to provide secure communication. A traditional public-key cryptosystem is appropriate in cryptography since it does not need the sender and receiver to supply the same secret to communicate without risk. However, they frequently rely on complex mathematical calculations, making them far less capable than equivalent symmetric-key cryptosystems. The high cost of encrypting long messages with public-key cryptography could be problematic in a wide range of applications. A hybrid system deals with it using a combination of the two. In WSN, Admin creates a token, a secret key, a public key, and a private key. Here, the token is used for access control in sensor devices and the administrator, the secret and public keys are utilized for packet encryption in sensor devices and the base station, and the private key is utilized for decryption in the administrator. Admin shares token with secret and public key for sensor devices and base station for encryption purposes. As a result, the TSP-KS algorithm was utilized to securely share these token with secret and public keys for sensor devices and base station over a distributed way. Experimental results demonstrate that the TSP-KS algorithm securely shares a token with a secret and public key.

Keywords: Security, keys sharing, token, hybrid cryptography, encryption and decryption

I. INTRODUCTION

Wireless sensor networks (WSNs) are a type of network that consists of interconnected sensor nodes that interact wirelessly in order to gather data about the environment [1]. Nodes are often low-power and distributed ad hoc and decentralized. WSNs are frequently used for area monitoring. The WSN is placed across a region where a phenomenon is to be tracked in area monitoring. The use of sensors to detect enemy intrusion is a military example. Devices in the WSN are frequently targeted by various types of attacks, including as flood assaults, eavesdropping attacks, etc [2]. Once an attacker has gained access to a sensor device, WSN information is no longer confidential and can be acquired by an adversary, putting the entire network at risk. Cryptography is required to address this problem [3].

Cryptography is a method of encrypting data and communication so that only the intended recipients can read and process it [4]. To protect communications between sensor devices, the WSN base station, and the admin, cryptographic methods use a set of encryption and decryption operations. One encryption algorithm, one authentication algorithm, and one key exchange algorithm are used in the cipher suite.

Traditional public-key cryptography is appropriate in cryptography since it does not need the sender and receiver to exchange the same secret in order to communicate safely [5]. However, because they frequently rely on sophisticated mathematical calculations, they don't always perform as well as symmetric-key cryptography. The high cost of encrypting long messages in public-key cryptography might be prohibitive for most applications. Hybrid cryptography handles it by combining the two [6].

Furthermore, access control is a data protection method that allows networks to govern who has access to data and resources [7]. Token-based access control is a form of authentication that adds an extra layer of security. Using this method, each sensor device has a token. It is impossible to use the packet without this token. Secret sharing also refers to the sharing of a secret amongst all sensor devices, each of which has been assigned a share of the secret [8]. The secret could be redo with only the right sensor device and when a sufficient number of shares are amalgamated.

This paper proposes a Token with Secret and Public Keys Sharing (TSP-KS) algorithm to avoid key leaks while sharing keys to all sensor devices for encryption. This algorithm aids the administrator and base station in the secure distribution of keys.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 9, Issue 9, September 2022

The remainder of the paper is structured as follows: Section II examines existing cryptography, access control, and secret sharing systems. TSP-KS algorithm to Avoid Keys Leakages is discussed in Section III. The experimental results and discussions of the proposed TSP-KS algorithm are provided in Section IV. Finally, Section V provides the conclusion.

II. RELATED WORK

This section explores existing cryptography, access control, and secret sharing systems.

The Abnormal Sensor Detection Accuracy (ASDA-RSA) approach proposed by Fotohi et al [9] is used to resist Denial of Sleep (DoS) assaults and reduce the amount of energy spent. The ASDA-RSA schema used in [9] contains two steps to improve security in WSNs. To avoid DoS attacks, a clustering methodology using energy and distance is employed in the first phase, and the RSA cryptographic algorithm and interlock protocol are utilized in the second phase, coupled with an authentication method. Furthermore, the ASDA-RSA approach is assessed here using extensive simulations in NS-2. In terms of average throughput, Packet Delivery Ratio (PDR), network lifetime, detection ratio, and average residual energy, the authors determined that WSN network performance metrics had improved.

Qazi et al [10] focused on security challenges in WSNs, and as a result, they were required to offer authentication and data encryption in a new way for node-to-node communication. With the help of the Elliptic Curve Digital Signature (ECDSA) cryptographic system to give an effective method for assessing key generation time, count of hello messages, and packet size, their proposed scheme not only offers security for the node to node communication network, but also hoards memory space on nodes. In addition, the Algorithm for Wireless Secure Communication (ASCW) enables key management with appropriate key length. Furthermore, ASCW aids in the secure communication of nodes, which aids in the greater and more effective security of the entire network. With the help of an authentication mechanism, ASCW further lowers the cost of risk and safety concerns on the network. According to the needed standards, a physical tested has been built based on devices and sensor nodes. The proposed methods were compared in terms of key generation time, the size of data packets, and the number of hello messages sent. According to the authors, ASCW is an appropriate and new solution for safeguarding data on nodes during WSN connection.

Mohindru et al. [12] suggested a hybrid cryptography technique to protect WSNs from node clone attacks. Along with the hash function, the proposed algorithm employs a mixture of symmetric (AES) and asymmetric (ECC) cryptographic techniques. Furthermore, the proposed approach verifies message integrity during sensor network transmission. The suggested hybrid algorithm's effectiveness was evaluated using several metrics such as communication, computation, and storage overheads. The effectiveness of the suggested hybrid algorithm is validated by a comparison of the outcomes. The suggested hybrid algorithm offers an energy sensor network solution that is both safe and effective.

Lin et al [13] presented a hierarchical secret sharing strategy based on linear homogeneous recurrent relations. On the basis of the elliptic curve public key cryptosystem paired with linear homogeneous recurrent relations, the authors propose a hierarchical multi-secret sharing system for WSNs. The authors of the suggested technique do not ensure that the contributors are only half-truthful. Additionally, the shadows of the participants could be reused. Computational security is their strategy. In their hierarchy multi-secret sharing model, only one share from each member is necessary. In comparison to current systems, it is better suited for WSNs.

Liu et al. [15] propose the Secret-Sharing-based Security Data Aggregation technique, which is a unique framework (S3DA). S3DA discusses a secret sharing mechanism and a data aggregation scheme. The first can enhance the safety and accuracy of sensing data, while the second can lower data transfer power usage. The authors concluded that their proposed framework is both efficient and accurate.

The performance of the RSA and ECC algorithms in the environment of WSNs is demonstrated by Kardi et al [19]. The differences in the achieved results, which can be described by the differences in their operation techniques, show that ECC surpasses RSA in terms of encryption key size, power expenditure, and decryption time, while RSA surpasses in terms of encryption key size, power consumption, and decryption time.

III. METHODOLOGY

The WSN network is made up of three entities: 1) the administrator, 2) the WSN base station, and 3) the sensor devices. Admin is the person in charge of monitoring the entire network; he could monitor it from anywhere at any time. Sensor devices sense their surroundings and provide data based on that information. Then, via the WSN Base Station, it sends the sensed data to the administrator. The WSN Base Station is a WSN network controller. It transmits the sensed data to the administrator when it has been collected. Sensor devices have a limited amount of energy, coverage area, transmission energy (Et), and receiving energy (Er). Sensor devices are placed at different positions across the network to detect heat, noise, vibrations, and pressure, among other things. To increase WSN network security, this section presented a Token with Secret and Public Keys Sharing (TSP-KS) algorithm, which was discussed in Algorithm 1.

For encryption, decryption, and access control, the admin generates three keys with one token and distributes them to all Sensor devices and base stations. They are:

1) The WSN Base Station's public key (for

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 9, Issue 9, September 2022

encryption),

2) The secret key for each sensor device (for encryption),
3) The token for each sensor device (for access control), and
4) The private key for personal use (for decryption).

After that, the administrator changes the public key, secret key, and tokens to n number of shares. After that, send these n shares to the WSN Base Station. The WSN Base Station reconstructs public keys, secret keys, and tokens based on shares after receiving any number of shares (<n). The WSN Base Station next transforms each Sensor device's secret keys and tokens into a n number of shares. Each share would be sent to the Sensor device that is authorized to receive it. All Sensor devices rebuild their own Secret Key and Token after receiving any number of shares (<n). The Token with Secret and Public Keys Sharing (TSP-KS) algorithm block diagram is shown in Figure 1.

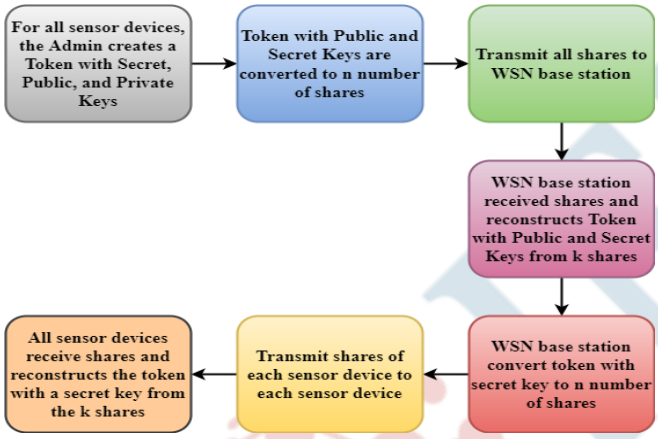

Figure 1: Block Diagram of TSP-KS algorithm

Algorithm 1: Token with Secret and Public Keys Sharing (TSP-KS) algorithm

Input : Admin, WSN base station (BS), Sensor devices (SDs)

Output : Token with Secret and Public Keys Sharing

Admin Side

Step 1 : For each sensor device SD from SDs do

Step 2 : KY[] = Keys_Generation()
// **Algorithm 2**

Step 3 : Public_Key = KY[0], Private_Key = KY[1], Secret_Key = KY[2]

Step 4 : Token = Token_Generation(SD)
// **Algorithm 3**

Step 5 : Keys = Public_Key + "||" + Secret_Key + "||" + Token

Step 6 : Let N, k

Step 7 : S[] = Shares_Generation(Keys, N, k)
// **Algorithm 4**

Step 8 : Forward S with k to BS

Step 9 : End For

WSN Base Station Side

Step 10 : For each sensor device SD from SDs do

Step 11 : BS received S with a k value

Step 12 : Take k shares for reconstructing

Step 13 : Keys = Shares_Reconstruction(k shares)
// **Algorithm 5**

Step 14 : KY[] = Keys.split("||")

Step 15 : Public_Key = KY[0], Secret_Key = KY[1], Token = KY[2]

Step 16 : Keys = Secret_Key + "||" + Token

Step 17 : Let N, k

Step 18 : S[] = Shares_Generation(Keys, N, k)
// **Algorithm 4**

Step 19 : Forward S with k to SD

Step 20 : End For

Sensor Device Side

Step 21 : SD received S with a k value

Step 22 : Take k shares for reconstructing

Step 23 : Keys = Shares_Reconstruction(k shares)
// **Algorithm 5**

Step 24 : KY[] = Keys.split("||")

Step 25 : Secret_Key = KY[0], Token = KY[1]

Keys Generation:

The key in cryptography is a sequence of bits used to transform a message into an unreadable format (ciphertext) and conversely. Three keys are needed in hybrid cryptography. Encryption uses public and secret keys, while decryption uses private and secret keys. Algorithm 2 describes the Key Generation Algorithm. It starts by taking two prime numbers, p1 and p2, at random. This algorithm also multiplies both prime numbers by m1. It also multiplies itself as m2 by subtracting one from each prime numbers. The algorithm estimates the co-prime of m2 after m2 creation. Co-prime numbers are two numbers that have only one common factor.

e1 has been assigned the co-prime of m2. The modular multiplicative inverse of e1 and m2 is then computed. The multiplicative inverse of e1 is then calculated. e2 is given this modular multiplicative inverse value. For the public key, the values e1 and m1 are used, as well as e2 and m1 for the private key. The AES technique was used to generate a secret key. To begin, this approach generates an AES key generator that functions as a secret (symmetric) key generator. This approach is then used to start a key generator with a 128-bit

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 9, Issue 9, September 2022

key size. This key generator also generates a secret. This algorithm then encodes the secret into the Secret Key.

Algorithm 2: Keys_Generation		
Input	:	Two Random Prime numbers (p1 and p2)
Output	:	KY[]
Step 1	:	Let KY[] = {}
Step 2	:	m1 = p1 * p2
Step 3	:	m2 = (p1-1) * (p2-1)
Step 4	:	e1 = getCoprime(m2)
Step 5	:	e2 = modInverse(e1, m2)
Step 6	:	KG = KeyGenerator.getInstance("AES
Step 7	:	KG.init(128)
Step 8	:	secret = KG.generateKey()
Step 9	:	Public_Key = e1, m1
Step 10	:	Private_Key = e2, m1
Step 11	:	Secret_Key = Base64.encode(secret.getEncoded())
Step 12	:	KY[0] = Public_Key, KY[1] = Private_Key, KY[2] = Secret_Key
Step 13	:	return KY[]

Token Generation:

This section uses the HMAC-SHA-1 method to generate a token. HMAC is a cryptographic hash function-based mechanism for confirming the authenticity of messages. HMAC can combine any iterative cryptographic hash function with a shared secret key, such as SHA-1 or MD5. The elements of the fundamental hash function determine HMAC's cryptographic strength. Token creation is described in Algorithm 3. To create HMAC-SHA-1 token for each sensor device.

Algorithm 3: Token_Generation(SD)		
Input	:	Sensor Device (SD), HmacSHA1
Output	:	Token
Step 1	:	SK = new SecretKeySpec(SD.getBytes(), HmacSHA1)
Step 2	:	mac = Mac.getInstance(HmacSHA1)
Step 3	:	mac.init(SK)
Step 4	:	RH[] = mac.doFinal(SD.getBytes())
Step 5	:	Token = new String(encode(RH))
Step 6	:	return Token

The admin wants to deliver the public key, the secret key, and a token to the WSN base station after Key Generation. The WSN base station also intends to provide the secret key together with a token to all sensor devices. However, due to the dynamic nature of a WSN network, key sharing is computationally risky and unreliable. This challenge necessitates the use of a secure key sharing approach. All keys must be merged before producing shares. The merged keys are then separated into several shares, which are then sent to the WSN base station or sensor devices. Algorithm 4 explains how to generate shares. The inputs for this algorithm are all Keys, N, and k. Here, N stands for the number of shares to be created, and k stands for the number of shares needed to reconstruct. The initial step in this algorithm is to assign an all Keys to a0. After that, it generates k-1 random numbers. For the sake of simplicity, k is set to three in this algorithm. As a result, two random numbers (k-1 = 3-1 = 2), a1 and a2, are required. This algorithm calculates N shares after generating random numbers. The architecture of share generation is depicted in Figure 2.

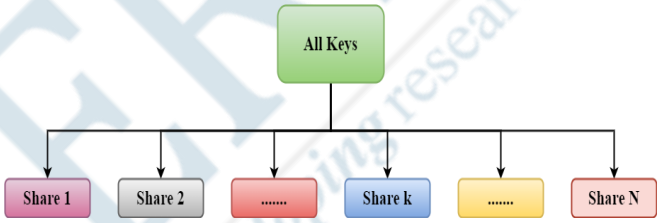

Figure 2: Shares generation

Algorithm 4: Shares_Generation(Keys, N, k)		
Input	:	Keys (Public and Secret Key with Token), N, k
Output	:	Shares (S)
Step 1	:	Let a0 = Keys, S = {}
Step 2	:	Generate Random (k-1) numbers (a1 and a2)
Step 3	:	Let f(p) = a0 + (a1 * p) + (a2 * p^2)
Step 4	:	For p=1; p<=N; p++
Step 5	:	S[p-1] = (p, f(p))
Step 6	:	End For
Step 7	:	return S

Shares Reconstruction:

A WSN base station or sensor device picks k shares after receiving all N shares in order to reconstruct all Keys. The proposed TSP-KS algorithm has this benefit. If any intermediary device becomes malicious, any of the shares may be lost. As a result, this algorithm requires k shares, which are sufficient for reconstruction. Algorithm 5 explains the reconstruction procedure. This algorithm uses a Lagrange polynomial formula to apply k shares. It offers a0, a1, a2, and so forth. For ease, k values are 3 (it gave while shares generation), thus this algorithm merely provides a0, a1, and

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 9, Issue 9, September 2022

a2. All Keys are contained in a0. Figure 3 depicts the architecture of shares reconstruction.

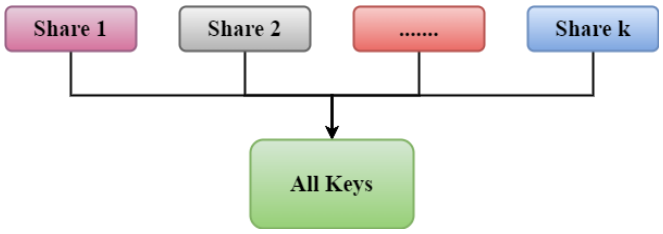

Figure 3: Shares reconstruction

Algorithm 5: Shares_Reconstruction(k shares)		
Input	:	k shares (p0,q0), (p1,q1), ... , (pk,qk)
Output	:	Keys
Step 1	:	f(p) = $\sum_{j=0}^{k}$(qj ∗ lj(p)) // Lagrange polynomial
Step 2	:	f(p) = a0 + (a1 * p) + (a2 * p^2)
Step 3	:	Keys = a0

IV. RESULTS AND DISCUSSIONS

The findings of the TSP-KS algorithm's experimentation and analysis on the WSN network are presented in this section. The majority of sensor devices in the WSN network are still dispersed randomly in this simulation. Temperature, moisture, brightness, wind speed, rainfall, and smoke are all measured by these sensor devices. Readings from these devices are sent to the admin through the WSN base station. The TSP-KS algorithm was evaluated using Java. Compare the proposed TSP-KS algorithm to other secret sharing algorithms such as Rabin's Information Dispersal Algorithm (IDA), HugoKrawczyk's Secret Sharing made short or Computational Secret Sharing Scheme (CSS), and AdiShamir's Perfect Secret Sharing Scheme (PSS) [20] by assessing the key sharing algorithm. Table 1 indicates the time it takes to construct a share (in milliseconds) (n = 5, k = 3).

Table 1: Time it takes to construct a share (in milliseconds) (n = 5, k = 3)

Algorithm	**Data Size (in KB)**		
	16	**32**	**64**
IDA	12.82	19.59	21.19
CSS	19.45	25.03	31.80
PSS	28.78	40.01	42.89
TSP-KS	10.97	17.19	19.68

Figure 4 depicts a graph of time versus data size in the creation of shares when n = 5, k = 3.

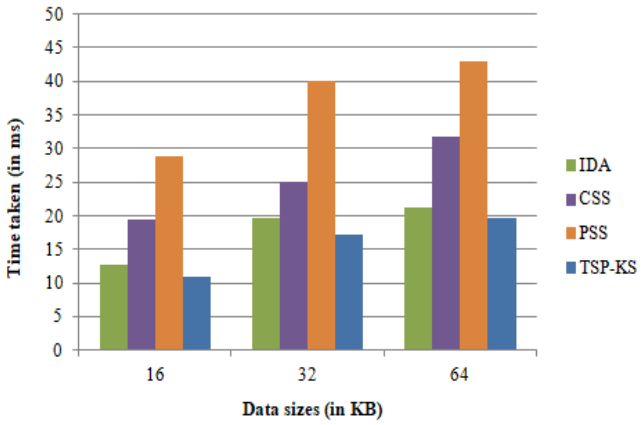

Figure 4: When n = 5, k = 3, the graph of the size of the data versus the time it takes to create a share

Tables 1 and Figure 4 show that, regardless of data size, TSP-KS is the fastest algorithm. In terms of the time it takes to create a share, IDA is second, CSS is third, and PSS is last. In compared to the other three algorithms, PSS has more scalability problems as data size grows. Table 2 also illustrates the time taken in (ms) for share recreation (n = 5, k = 3).

Table 2: Time spent on share recreation (in ms) (n = 5, k = 3)

Algorithm	**Data Size (in KB)**		
	16	**32**	**64**
IDA	17.82	19.51	22.57
CSS	19.27	21.63	26.11
PSS	30.01	20.00	23.89
TSP-KS	15.76	17.04	20.49

When n = 5, k = 3, the graph of time vs data sizes in share recreation is shown in Figure 5.

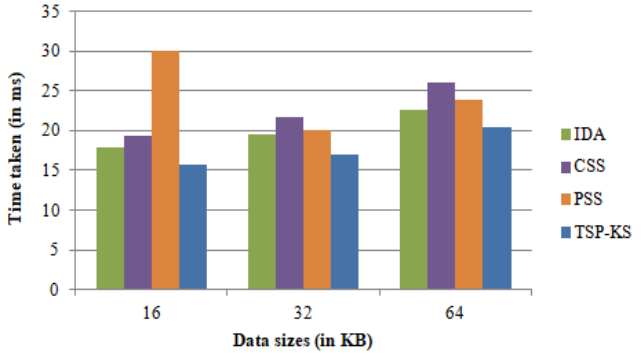

Figure 5: When n = 5, k = 3, the graph of the size of the data versus the time it takes to share recreation

TSP-KS is the quickest algorithm regardless of data size, as seen in Tables 2 and Figure 5. In terms of time spent on share recreation, IDA is second, CSS is third, and PSS is last. In compared to the other three algorithms, PSS has more scalability problems as data size grows.

V. CONCLUSION

To mitigate key leaks in the WSN, this paper proposed the Token with Secret and Public Keys Sharing (TSP-KS) algorithm. A public key, a private key, a secret key, and a token are all created by the administrator in a WSN network. The public and secret keys for packet encryption, the private key for decryption, and the token for access control. The admin distributes a public and secret key with the token for WSN base station and sensor devices for encryption purposes. As a result, the TSP-KS algorithm provides security while keys shared in a distributed manner. Experimental Results have shown that TSP-KS algorithms provide secure key sharing with reduced share creation and recreation time in WSN monitors when compared to prior key sharing algorithms.

REFERENCES:

[1] Abdulkarem, M., Samsudin, K., Rokhani, F. Z., & A Rasid, M. F. (2020). Wireless sensor network for structural health monitoring: a contemporary review of technologies, challenges, and future direction. Structural Health Monitoring, 19(3), 693-735.

[2] Riaz, M. N., Buriro, A., & Mahboob, A. (2018). Classification of attacks on wireless sensor networks: A survey. International Journal of Wireless and Microwave Technologies, 8(6), 15-39.

[3] Radhappa, H., Pan, L., Xi Zheng, J., & Wen, S. (2018). Practical overview of security issues in wireless sensor network applications. International journal of computers and applications, 40(4), 202-213.

[4] Mirvaziri, H., & Hosseini, R. (2020). A novel method for key establishment based on symmetric cryptography in hierarchical wireless sensor networks. Wireless Personal Communications, 112(4), 2373-2391.

[5] Bashaa, M. H., Al-Alak, S. M., & Idrees, A. K. (2019, April). Secret key generation in wireless sensor network using public key encryption. In Proceedings of the international conference on information and communication technology (pp. 106-112).

[6] Prakash, S., & Rajput, A. (2018). Hybrid cryptography for secure data communication in wireless sensor networks. In Ambient Communications and Computer Systems (pp. 589-599). Springer, Singapore.

[7] Luo, M., Luo, Y., Wan, Y., & Wang, Z. (2018). Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT. Security and Communication Networks, 2018.

[8] Haseeb, K., Islam, N., Almogren, A., Din, I. U., Almajed, H. N., & Guizani, N. (2019). Secret sharing-based energy-aware and multi-hop routing protocol for IoT based WSNs. IEEE Access, 7, 79980-79988.

[9] Fotohi, R., Firoozi Bari, S., & Yusefi, M. (2020). Securing wireless sensor networks against denial-of-sleep attacks using RSA cryptography algorithm and interlock protocol. International Journal of Communication Systems, 33(4), e4234.

[10] Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 12(1), 547-566.

[11] Ali, S., Humaria, A., Ramzan, M. S., Khan, I., Saqlain, S. M., Ghani, A., ... & Alzahrani, B. A. (2020). An efficient cryptographic technique using modified Diffie–Hellman in wireless sensor networks. International journal of distributed sensor networks, 16(6), 1550147720925772.

[12] Mohindru, V., Singh, Y., & Bhatt, R. (2020). Hybrid cryptography algorithm for securing wireless sensor networks from Node Clone Attack. Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering), 13(2), 251-259.

[13] Lin, Y., Zhu, H., Xu, G., & Xu, G. (2022). Hierarchical secret sharing scheme for WSN based on linear homogeneous recurrence relations. International Journal of Distributed Sensor Networks, 18(3), 15501329221088740.

[14] Mahalat, M. H., Karmakar, D., Mondal, A., & Sen, B. (2021). PUF based Secure and Lightweight Authentication and Key-Sharing Scheme for Wireless Sensor Network. ACM Journal on Emerging Technologies in Computing Systems (JETC), 18(1), 1-23.

[15] Liu, X., Ma, W., Yu, J., Yu, K., & Xiang, J. (2021, June). A Secret-Sharing-based Security Data Aggregation Scheme in Wireless Sensor Networks. In International Conference on Wireless Algorithms, Systems, and Applications (pp. 303-313). Springer, Cham.

[16] Iqbal, U., & Mir, A. H. (2020). Secure and practical access control mechanism for WSN with node privacy. Journal of King Saud University-Computer and Information Sciences.

[17] Liu, Z., Ma, Q., Liu, W., Sheng, V. S., Zhang, L., & Liu, G. (2018). Access control model based on time synchronization trust in wireless sensor networks. Sensors, 18(7), 2107.

[18] Hamsha, K., & Nagaraja, G. S. (2019, February). Threshold cryptography based light weight key management technique for hierarchical WSNs. In international conference on ubiquitous communications and network computing (pp. 188-197). Springer, Cham.

[19] Kardi, A., Zagrouba, R., & Alqahtani, M. (2018, April). Performance evaluation of RSA and elliptic curve cryptography in wireless sensor networks. In 2018 21st Saudi Computer Society National Computer Conference (NCC) (pp. 1-6). IEEE.

[20] W. J. Buchanan, D. Lanc, E. Ukwandu, L. Fan, G. Russell and O. Lo, “The Future Internet: A World of Secret Share”, Future Internet 2015, 7, 445-464, DOI:10.3390/fi7040445.