Lightweight Group Key Establishment for Reducing Memory Overhead

Siti Agustini*,**, Wirawan*, Gamantyo Hendrantoro*

*Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

**Department of Informatics, Institut Teknologi Adhi Tama Surabaya, Surabaya 60117, Indonesia

7022211005@mhs.its.ac.id, wirawan@ee.its.ac.id, gamantyo@ee.its.ac.id

Abstract— **Wireless Sensor Network (WSN) and Internet of Things (IoT) allow sensor devices to collect information about various critical sectors through wireless networks. However, when the WSNs are connected to a public network, the security of the WSN is vulnerable. Besides, WSN needs a key distribution scheme to secure data among other sensor devices. Furthermore, IoT devices have low computing, energy, and memory storage capabilities. Thus, designing a lightweight, efficient, and secure protocol communication for WSN is always a challenge due to the resource constraint of sensor devices. The existing schemes result in the number of keys stored by sensor devices depending on group size. When the group size increases, the number of the stored key by the sensor also increases. Other research proposes key establishment based on polynomial multiplicative and causes high computational capability. This paper proposed a key distribution scheme based on (p,q)-Lucas polynomial and XOR to achieve lightweight, memory overhead efficiency and security. The proposed method is evaluated in several parameters: memory overhead, communication overhead, energy consumption, computational complexity analysis, and security. The results indicate that our scheme outperforms the existing approaches regarding memory overhead, computation efficiency, and support security.**

Keywords— **Group key establishment, WSN, Lucas polynomial, memory overhead, information security, key distribution.**

I. INTRODUCTION

Wireless Sensor Network (WSN) consists of sensors working together to sense the environmental conditions and send them to the sink node through a wireless network. The characteristics of WSN are self-configured and infrastructure-less. WSN is also easy to implement for a long time and inexpensively. Based on the features and advantages, WSN can be applied in various sectors such as health monitoring [1], [2], agricultural [3]–[5], military [6], [7], fire detection [8], disaster [9], water quality [10], and many more. Nowadays, WSN has integrated with the Internet of Things (IoT). This technology allows many sensors to join a network to provide various data. However, when WSN integrates with IoT, WSN will connect to a public network, and the securities will be vulnerable to attacks. For instance, eavesdropping attacks allow the intruder to collect unencrypted communication key information, message identities, gateways, and others [11], [12]. Encryption methods can be implemented for securing WSNs. One major problem with implementing the encryption technique is properly distributing the secret key for communicating with any network node [13].

Asymmetric encryption and symmetric encryption can address the problem of key distribution. Key distribution using asymmetric encryption or Public Key Cryptography (PKC) has the advantage of not needing a secure channel for key exchange, but the computational complexity is high. It makes the performance of PKC much slower than the symmetric encryption scheme [14]–[17] and is improper for WSN because of the constrain of the sensor in resource and computation [18]. Meanwhile, working with symmetric encryption, any parties must share the same key, which must be secured from unauthorized parties. Therefore, a secure key distribution is needed. A key distribution scheme using a symmetric key has lower computational complexity than asymmetric encryption [14], so implementing symmetric encryption is still preferred over asymmetric encryption [19]. Most distribution schemes involve a Key Distribution Center (KDC) with a predistribution key mechanism where the keying information is loaded into the sensors before deployment. KDC provides end-to-end encryption so only authorized parties can compute the secret key [14].

Blom [20] proposed a key distribution scheme based on symmetrical polynomials with a pairwise key type. When two nodes want to communicate, they will calculate the shared key of the polynomial obtained from KDC and the information on the random value from other users. If the value generated by the two users is the same, then the node can be called a secure link key. This scheme has several disadvantages. Firstly, each node must store as many keys as the network size (n) – 1, so the memory overhead will be high. Secondly, KDC must distribute the key information to all network members making this approach expensive even on a small scale and impossible to implement.

Several studies with pairwise key types have solved the first problem regarding memory overhead [21]–[24]. The results of this study succeeded in reducing the memory requirements used to store the preliminary key before network deployment. However, the used concept is a pairwise key. So the complexity of KDC will increase because it must

distribute each preliminary key for each sensor in the network, and memory consumption is high because each user must store $n-1$ establishment keys due to each node must run the establishment key creation mechanism to all other nodes in the network. To overcome this problem, a group establishment key scheme was proposed [21], [25]–[27]. This scheme divides the network into groups and communicates between nodes in the group using the group establishment key. This solution can reduce memory consumption by the group size, but the memory consumption will increase with the group size. Considering the previous research, it is still necessary to develop a scheme where memory consumption is independent of the group size and network. Another problem is the constraint of IoT that has low capabilities in energy and computational [28]. So that, we must design a lightweight key establishment scheme.

In this paper, we proposed a novel group key distribution scheme based on *(p,q)*-Lucas polynomial and XOR operations to achieve a lightweight scheme and memory overhead efficiency. This research was implemented by simulation using Network Simulator-2 (NS-2), version 2.35. The contributions of this paper can be summarized below:

- Lucas polynomials can produce odd-degree or even-degree polynomials. This condition causes a decrease in computational complexity compared than uses normal polynomials. Applying the XOR operation does not change the length of the keying information until the group key is reached. Thus, the combination of *(p,q)*-Lucas polynomial and XOR operation make this scheme lightweight.
- Our research proposed a single group key for each group communication to reduce the number of keys that SN must store. Thus, any number of SNs in a group does not change the number of keys the SN must keep.
- Moreover, the motivation for using *(p,q)*-Lucas polynomial is that the polynomial can form any degree in any base number q [29], so that it can reduce the probability of an attacker guessing the used polynomial. In addition, we prove that our scheme supports forward and backward secrecy.

The rest of this paper is organized as follows: firstly, we introduce the model system used in our proposed scheme in Section II. In the next session, we will demonstrate the proposed scheme in detail. Section IV discusses performance and comparison to other existing schemes. Finally, Section V is the conclusion.

II. SYSTEM MODEL

In this section, we will describe the details of the system model in two parts; the network model and the protocol description.

A. Network Model

The details of the network model of the proposed approach are shown in Fig. 1. From the illustration, it can be informed that a WSN consists of t groups where each group has n sensor nodes or communicators. We assume that WSNs have three elements: communication groups of Sensor Nodes (SN), Base Station (BS) as sink node, and Key Distribution Center (KDC). The aspects of WSN have properties as follows:

1) Sensor Node (SN): Any group communications consist of Sensor Nodes (SNs). The SNs have constrained resources such as energy and memory. SNs are responsible for sensing the environmental condition and sending them to the Base Station. SNs accomplish key session generation and group key establishment in this proposed scheme. To join a group communication, an SN must register with the Key Distribution Center (KDC) to obtain a token.

2) Base Station (BS): BS is assumed secure and has higher processing capacity and memory than SN. BS acts as an interface between users and SNs. This BS receives all sensing data that SN obtained.

3) Key Distribution Center (KDC): KCD has high computational ability compared to other elements, such as SN and BS, for helping establish a group key. In this scheme, KDC generates and distributes keying information to all SN in WSN.

B. Protocol Description

Fig. 2 illustrates the protocol in the group key establishment scheme. This research assumes that a KDC manages the keys used in the network. The protocol of this scheme consists of 3 steps. **Step 1.** All SNs register to KDC, and KDC responds with **Step 2.** Each node in the group obtains a polynomial and a random number which will later be used to form the group key shared by KDC. **Step 3.** Then, each node inputs the random number into the obtained polynomial to obtain the session key. In the next stage, each node generates a secret input and distributes it to other nodes in the same group. An XOR operation between the session key and the secret input generates the secret key. Then, use XOR between the secret key obtained and the session key to return each user's secret input. At this stage, each node has all the secret inputs from nodes in the same group. So, calculating the group key is done by XOR for all the secret inputs. This group key is used for communication between groups.

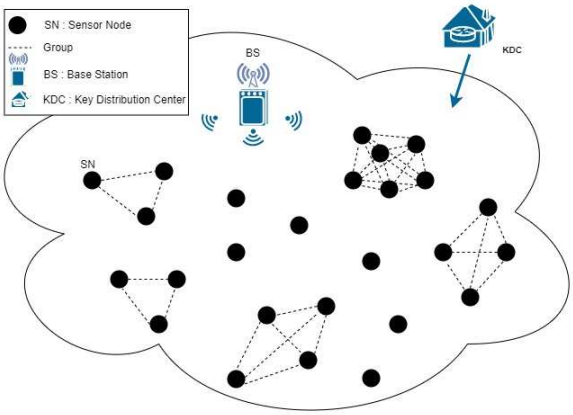

FIGURE 1. Network model of the proposed method

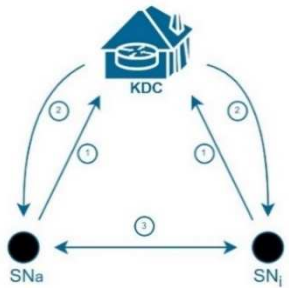

FIGURE 2. Protocol Description

III. THE PROPOSED METHOD

This section presents the proposed group key establishment scheme consisting of three stages: token generation, session group key generation, and group key establishment.

A. Token Generation

Initially, KDC generates random base number p and produces $(p, q)-$ Lucas polynomial. In general, the (p, q) −Lucas polynomial equation can be described as follows:

$$F_n(x) = \begin{cases} 2x^0 & ; n = 0 \\ 1x^1 & ; n = 1 \\ pF_{n-1}(x)x^1 + F_{n-2}(x)x^0; & n \geq 2, p \geq 1 \end{cases}$$

The first 10 (p, q) −Lucas polynomial when $p = 1$ can be seen below [29] :

$F_1(x) = x$ (1)
$F_2(x) = x^2 + 2$ (2)
$F_3(x) = x^3 + 3x$ (3)
$F_4(x) = x^4 + 4x^2 + 2$ (4)
$F_5(x) = x^5 + 5x^3 + 5x$ (5)
$F_6(x) = x^6 + 6x^4 + 9x^2 + 2$ (6)
$F_7(x) = x^7 + 7x^5 + 9x^3 + 7x$ (7)
$F_8(x) = x^8 + 8x^6 + 20x^4 + 16x^2 + 2$ (8)
$F_9(x) = x^9 + 9x^7 + 27x^5 + 30x^3 + 9x$ (9)
$F_{10}(x) = x^{10} + 10x^8 + 35x^6 + 50x^4 + 25x^2 + 2$ (10)

The weakness of using polynomials is the number of multiplication is high [21], [27]. Furthermore, when the group size increases, so does the computation complexity. For instance, Han et al. [30] form a polynomial by multiplying all secret values of SNs. This scheme produces a polynomial with degree n where n is the number of SNs in a group. However, in this scheme KDC chooses one polynomial $F_n(x)$ of those (p, q) −Lucas polynomials and generate q that is the value of variable x.

Algorithm 1 Token Generation (KDC Side)

Input: p;

Output: $F_n(x)$;

1: Generate p, q;

2: $F_n(x) = pF_{n-1}(x)x^1 + F_{n-2}(x)x^0$;

3: Send $\{q, F_n(x)\}$ to SNs

The chosen polynomial $F_n(x)$ and q will be distributed to all SN in a group. Thus, the degree of polynomials does not depend on group size. Moreover, KDC also sends the current number of valid SNs N_{SN}. So that, KDC will broadcast $K_m = \{F_n(x),\ q,\ N_{SN}\}$. The details of token generation are shown by Algorithm 1.

B. Session Group Key Generation

This stage describes the session group key generation as Algorithm 2. For n users $SN_1, SN_2 \ldots\ldots, SN_n$ in a group, each node receives polynomial $F_n(x)$ and q from KDC during initialization. The variable q of x is used to compute the value of the polynomial, which is called the session key sg_a.

Algorithm 2 Session Group Key Generation (SN side)

Input: $q, F_n(x)$;

Output: sg_a;

1: Compute: $sg_a = F_n(q)$;

C. Group Key Establishment

The last stage is group key establishment as Algorithm 3. This process begins when n users $U_{a_1}, U_{a_2}, \ldots\ldots, U_{a_n}$ in group communication group generate a secret input s_i and distribute it to all nodes in the group using the session key sg_a which has been calculated to obtain the secret key sk_{a_i}.

$$sk_{a_i} = sg_a \oplus s_i \quad (11)$$

Then, the other nodes extract the received sk_{a_i} using XOR operation between sg_a and sk_{a_i} dan for gaining s_i.

$$s_i = sg_a \oplus sk_{a_i} (12)$$

For a group key establishment, each node U_{a_i} computes group key gk_a by XOR operation between s_i and the others.

$$gk_a = s_1 \oplus s_2 \oplus s_3 \oplus \ldots\ldots \oplus s_n \ (13)$$

In the last step, each SN distributes $B_j = \{gk_a, N_{SN}\}$ to all SNs in group communication, where N_{SN} is the number of secret input s_i that SN is obtained from other SNs, including its secret input. The subscript j only when updating B.

Algorithm 3 Group Key Establishment (SN side)

Input: sg_a, s_i;

Output: gk_a;

1: Compute: $sk_{a_i} = sg_a \oplus s_i$;

2: Send $\{sk_{a_i}\}$ to SNs.

3: Compute: $s_i = sg_a \oplus sk_{a_i}$;

4: Compute: $gk_a = s_1 \oplus s_2 \oplus s_3 \oplus \ldots\ldots \oplus s_n$;

IV. RESULTS AND DISCUSSIONS

This section presents the performance metrics to evaluate the proposed method. This research is being done with the following assumptions: KDC and BS have high resources and trustworthiness that cannot be compromised, all SNs in this simulation are static, and the SNs have similar resource

capability in energy, memory storage, computational power, etc. The Wireless Sensor Network environment was simulated using Network Simulator-2 (NS-2) version 2.35 with the following simulation parameters: initial SN's energy: 10 J, number of SN: 1000, network topology: 500 m × 500 m, transmit power and receive power: 1 J, simulation time: 100 s.

A. Performance Analysis

1) Memory Overhead: Memory constraint is one of the issues in WSN. Therefore, the key distribution scheme must be designed effectively regarding memory consumption. This section discusses memory overhead in two terms: preliminary key and the number of the key stored. For the preliminary key, each SN in a network receives a polynomial and value q and generates a secret input s_i during initialization. Assume that polynomial, value q, and secret input s_i take k, s, and j space, respectively. The total memory space consumed by an SN can be written as $T_m = k + s + j$.

Another major issue in group key establishment is the number of the common secret keys needed to be stored in each SN increasing as the group size. Compared with the existing scheme, our scheme is efficient in terms of memory consumption. In our scheme, each SN in a group receives one polynomial and implements XOR for all SNs' secret input. This technique takes only one common secret key needed to store in memory. It is static and doesn't increase when the group size increases.

2) Communication Overhead: Energy is another issue of WSN's constraint. Therefore, the key distribution scheme must have less information exchange to reduce the energy consumed by radio transmission. The energy consumed during the key distribution process depends on the number of packet exchanges and the size of each packet. Our schemes only need one keying information exchange to compute the group key. The secret input s_i is distributed to all SNs in a group network and makes this scheme efficient.

3) Computational Complexity Analysis: This scheme involves Lucas polynomial in generating a token in Step 1. Lucas polynomial sequences consist of odd-degree polynomial and even-degree polynomial as follow [21]:

$$F_n(x) = w_0x^0 + w_1x^2 + w_2x^4 + \cdots + w_nx^{2n}$$

or

$$F_n(x) = w_0x^1 + w_1x^3 + w_2x^5 + \cdots + w_nx^{2n+1}$$

where $w_1, w_2, w_3, \ldots, w_n$ are constants. Therefore, the computational costs for establishing a group key with an even-degree polynomial and odd-degree polynomial are as follows:

$$\frac{(2+n)\times n}{4} \times c + \frac{n}{2} \times m$$

and

$$\frac{(1+n)^2}{4} \times c + \frac{n}{2} \times m$$

If we consider the normal polynomial as below:

$$F_n(x) = w_0x^0 + w_1x^1 + w_2x^2 + \cdots + w_nx^n$$

So, we can express the computational complexity of a normal polynomial as:

$$\frac{(1+n)\times n}{2} \times c + n \times m$$

where c and m are multiplicative cost and additive cost, therefore, the computational complexity of the Lucas polynomial is lower than the normal polynomial.

Our scheme uses the XOR function to establish the group key in Step 2. XOR function is a simple arithmetic calculation by binary addition. Therefore, our scheme is lightweight and computation efficient.

4) Energy Consumption: This section discusses the energy consumption of a single SN by applying a different number of SNs in network communication. To simplify the simulation, we assume the used Lucas polynomial from Eq. (2) and q is a random number in Z_n, where n = 10. Fig. 4 shows the energy consumption of a single SN when the number of SNs in the network is between 100-1000. Fig. 4 shows that the consumed energy always increases with the number of SNs. In our scheme, each SN shares its secret input with others in group communication. It reflects why energy consumption is increasing.

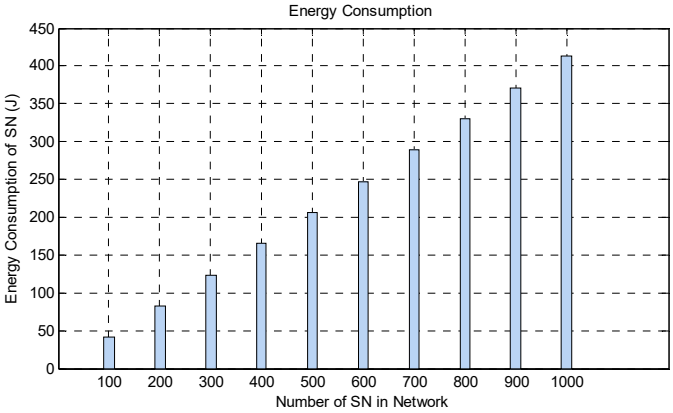

FIGURE 4. Energy consumption in various numbers of SN

B. Security Analysis

This section discusses the security analysis of the proposed scheme, which involves forward secrecy, backward secrecy, and group key confidentiality.

1) Forward secrecy*:* when an SN has been revoked from a group, then it cannot obtain any group key gk_a. Since the group key can be derived among members of group communication, the group key will be regenerated. Moreover, the KDC will renew the N_{SN} that indicates there is SN has been removed.

2) **Backward secrecy:** if a new node joins a group, it cannot access the previous group key since it is obtained by involved nodes in group communication. The new SN will generate a secret value s_i and share it with all SNs so that the new group key will be established by Eq. (13) and refresh the previous group key. The N_{SN} also feature gives information that the number of member's group increase.

3) **Group Key Confidentiality:** This parameter relates to SN outside the group and cannot derive the key to a group. Only SN that has got a specific polynomial $F_n(x)$ from KDC can calculate the key to that group. When SN does not get a polynomial from KDC, then the SN cannot get the secret value s_i belonging to other SNs because the shared secret value s_i is hidden with the output value of the polynomial sg_a like Eq. (11). To get the actual value of the secret value s_i of all SNs, SN must perform calculations such as Eq. (12), namely extracting sk_{a_i} with sg_a using the XOR operation. When an attacker has not registered with KDC, he must find which polynomial to use for a group. With the use of (p, q)-Lucas polynomial, each value p will produce a sequence of Lucas polynomials $F_n(x)$ with varying degrees of polynomials, it will be difficult for attackers to guess which used polynomials. When the attacker cannot find out the polynomial $F_n(x)$ and the value p is used, the attacker cannot extract the secret value s_i of each SN in the group.

B. *Comparison with Other Similar Schemes*

Now, we focus on performance comparison regarding memory overhead and communication overhead. Table 1 shows the memory overhead comparison between 2 existing schemes in terms of the number of key storages by a single SN. We compare our scheme with Manasrah et al.'s [22] and Dinker et al.'s [31]. The number of keys stored in Manasrah et al.'s scheme and Dinker et al.'s are $t-1$ and $n-1$, respectively, where t is the network size and n is group size.

TABLE 1. Performance Comparison

Scheme	Key Type	Memory Overhead	Number of keying Information exchanges
Manasrah et al.'s [22]	Pairwise	$t-1$	3
Dinker et al.'s [31]	Pairwise	$n-1$	1
Our scheme	Group	1	1

V. CONCLUSIONS

In this paper, we proposed a group key establishment scheme based on (p, q) −Lucas polynomial and XOR operations. Our approach contains three elements: KDC, BS, and a group of SNs. First of all, KDC generates p and q and then computes sequences of (p, q) −Lucas polynomial. KDC selects one of the polynomials in the sequence and shares it along with value q. Secondly, all SNs in a group create a session group key by computing the received polynomial and value q. At least, each SN generates a random value and broadcasts it to other SNs to establish a group key. We conducted performance evaluations that show the memory overhead and communication overhead. We also analyzed the security in terms of forward security and backward security. Finally, we also compared our proposed group key distribution with two existing schemes and showed the superiority. The current schemes provide a key distribution scheme where the number of keys stored depends on group size or network size. On the other hand, the memory consumption in our scheme is static and free from group size. Furthermore, our approach used Lucas polynomial and XOR operation to obtain the group key. It makes our scheme lightweight. This paper includes an analysis of the security and performance evaluation of the proposed scheme. We can conclude that our scheme is lightweight, efficient in memory overhead, and also support backward and forward secrecy.

ACKNOWLEDGMENT

This reported work was made possible by the BPI scholarship funding awarded to the first author from the Indonesian government through PUSLAPDIK and LPDP.

REFERENCES

[1] T. P. Latchoumi, R. Swathi, P. Vidyasri, and K. Balamurugan, "Develop New Algorithm To Improve Safety On WMSN In Health Disease Monitoring," in *2022 International Mobile and Embedded Technology Conference (MECON)*, 2022, pp. 357–362.

[2] T. M. Saravanan, T. Kavitha, S. Hemalatha, and M. M. Ajmal, "IoT Based Health Observance System using Fog Computing: A Precise Review," in *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, 2022, pp. 1–5.

[3] B. Ravinder and P. Reddy, "An Advanced Agriculture IoT Technology with Wireless Application," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021, pp. 809–812.

[4] S. Anulekshmi and D. R. Durga, "Performance Improvement of the Wireless Sensor Network with Proficient Power Management with Supervised Multimodel Data Regression Algorithm In Precision Agriculture," in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, 2022, pp. 754–761.

[5] F. Sun, W. Zang, H. Huang, I. Farkhatdinov, and Y. Li, "Accelerometer-Based Key Generation and Distribution Method for Wearable IoT Devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1636–1650, Feb. 2021,

[6] J. J. Kang, "A Military Mobile Network Design: mHealth, IoT and Low Power Wide Area Networks," in *2020 30th International Telecommunication Networks and Applications Conference (ITNAC)*, 2020, pp. 1–3.

[7] Lixianli, P. Wei, A. Jianyong, and W. Ping, "The Application Research on Military Internet of Things," in *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2020, pp. 187–191.

[8] A. Ullah *et al.*, "A Survey on Continuous Object Tracking and Boundary Detection Schemes in IoT Assisted Wireless Sensor Networks," *IEEE Access*, vol. 9, pp. 126324–126336, Sept. 2021.

[9] S. Singh, A. S. Nandan, A. Malik, N. Kumar, and A. Barnawi, "An Energy-Efficient Modified Metaheuristic Inspired Algorithm for Disaster Management System Using WSNs," *IEEE Sens. J.*, vol. 21, no. 13, pp. 15398–15408, July 2021.

[10] P. Madhurima, K. Yadav, R. Gupta, and J. S. Jadon, "Real Time Smart Water Management System using IoT," in *2022 International Mobile and Embedded Technology Conference (MECON)*, 2022, pp. 563–566.

[11] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE*

Commun. Surv. Tutorials, vol. 22, no. 1, pp. 616–644, 2020.

[12] J. Lee, S. Yu, M. Kim, Y. Park, and A. K. Das, "On the Design of Secure and Efficient Three-Factor Authentication Protocol Using Honey List for Wireless Sensor Networks," *IEEE Access*, vol. 8. pp. 107046–107062, June 2020.

[13] G. Manikandan and U. Sakthi, "A comprehensive survey on various key management schemes in WSN," *Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2018*, pp. 378–383, 2019.

[14] D. R. Stinson and M. B. Paterson, *Cryptography: Theory and Practice*, 4th ed., A. Boggess and K. Rosen, Ed. London, New York:Taylor & Francis Group, 2019.

[15] A. Murtaza, S. J. Hussain Pirzada, and L. Jianwei, "A New Symmetric Key Encryption Algorithm With Higher Performance," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–7.

[16] A. Hamza and B. Kumar, "A Review Paper on DES, AES, RSA Encryption Standards," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2020.

[17] Q. Zhang, "An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption," in *2021 2nd International Conference on Computing and Data Science (CDS)*, 2021, pp. 616–622.

[18] M. S. Yousefpoor and H. Barati, "Dynamic key management algorithms in wireless sensor networks: A survey," *Computer Communications*, vol. 134. pp. 52–69, 2019.

[19] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks," *Secur. Commun. Networks*, vol. 7, no. 12, pp. 2495–2508, Dec. 2014.

[20] R. Blom, "Non-public key distribution," in *Advances in Cryptology.* Springer, pp.231-236, 1983.

[21] A. K. Gautam and R. Kumar, "A key management scheme using (p, q)-lucas polynomials in wireless sensor network," *China Commun.*, vol. 18, no. 11, pp. 210–228, Nov. 2021.

[22] A. M. Manasrah, A. R. AL-Rabadi, and N. A. Kofahi, "Key pre-distribution approach using block LU decomposition in wireless sensor network," *Int. J. Inf. Secur.*, vol. 19, no. 5, pp. 579–596, Oct. 2020.

[23] E. Khan, E. Gabidulin, B. Honary, and H. Ahmed, "Matrix-based memory efficient symmetric key generation and pre-distribution scheme for wireless sensor networks," *IET Wirel. Sens. Syst.*, vol. 2, no. 2, pp. 108–114, June 2012.

[24] S. J. Choi, K. T. Kim, and H. Y. Youn, "An energy-efficient key predistribution scheme for secure wireless sensor networks using eigenvector," *Int. J. Distrib. Sens. Networks*, vol. 2013, May 2013.

[25] M. Padmashree, K. Ranjitha, S. Arunalatha, and R. Venugopal, "CKDAC: Cluster-Key Distribution and Access Control for Secure Communication in IoT," *7th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Electron. Comput. Eng. UPCON 2020*, 2020.

[26] Q. Cheng, C. Hsu, and L. Harn, "Lightweight Noninteractive Membership Authentication and Group Key Establishment for WSNs," *Math. Probl. Eng.*, vol. 2020, May 2020.

[27] A. Albakri and L. Harn, "Non-Interactive Group Key Pre-Distribution Scheme (GKPS) for End-To-End Routing in Wireless Sensor Networks," *IEEE Access*, vol. 7. pp. 31615–31623, Feb. 2019.

[28] A. Maria, I. Nazurl, and J. NZ, "A Lightweight and Secure Authentication Scheme for IoT Based E-Health Application," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 1, pp. 107–120, 2019.

[29] Thomas Koshy, *Fibonacci and lucas number*. John Wiley & Sons, Inc, 2019.

[30] S. Han, M. Gu, B. Yang, J. Lin, H. Hong, and M. Kong, "A secure trust-based key distribution with self-healing for internet of things," *IEEE Access*, vol. 7, pp. 114060–114076, Augt. 2019.

[31] A. G. Dinker and V. Sharma, "Polynomial and matrix based key management security scheme in wireless sensor networks," *J. Discret. Math. Sci. Cryptogr.*, vol. 22, no. 8, pp. 1563–1575, 2019.

Siti Agustini was born in 1990. She received a bachelor's degree in telecommunication engineering from Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Surabaya, Indonesia, in 2012 and a master's degree in Department of Electrical Engineering from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia in 2014. Currently, she is pursuing a Ph.D. degree with the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. Her research interests include wireless communication, computer networks, wireless sensor networks, and cryptography, especially information security.

Wirawan was born in 1963. He received his bachelor's degree in 1987 from the Department of Electrical Engineering, Institute Teknologi Sepuluh Nopember, Surabaya. In 1996 and 2003, he received the DEA degree from Ecole Nationale Superieure d'Informatique (ESSI), Universite Nice Sophia Antipolis (UNSA), France, and the Ph.D. degree from Telecom ParisTech, Paris, France, (ENST), respectively. He is a lecturer in the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya. His research area involves wireless ad hoc networks, wireless communications, multimedia signal processing, underwater acoustics communication and networking, and sensor networks.

Gamantyo Hendrantoro (Senior Member, IEEE) was born in 1970. He received the B.Eng. degree in 1992 from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, and the M.Eng. and Ph.D. degree from Carleton University, Ottawa, Canada in 1997 and 2001, all in the Department of Electrical Engineering. He is currently a Professor in the Department of Electrical Engineering, ITS. His research interests include wireless communications, channel modeling, wireless system for tropical areas, millimeter-wave propagation, signal processing, and radio propagation channel modeling.