A Lightweight Encryption Algorithm to Enhance Wireless Sensor Network Security on the Internet of Things

Ntebatseng Mahlake*, Topside E. Mathonsi, Deon Du Plessis, and Tonderai Muchenje
Tshwane University of Technology, Pretoria 0122, South Africa
Email: mathonsite@tut.ac.za; DuPlessisDP@tut.ac.za; MuchenjeT@tut.ac.za

Abstract —The Internet of Things (IoT) is an anticipated future technology that promises to connect a massive number of devices over the internet. Wireless Sensor Networks (WSNs) are regarded as one of the most essential subnetworks of the IoT. Sensor networks are being utilized by IoT to gather, monitor, and send sensitive data across wireless networks. Because the information transferred through WSNs is easily exposed to cyber-attacks, data security is critical. In WSNs, the attacker's adversary aims are to deteriorate and halt the network's effective use, as well as to interrupt network services, rendering them unreachable to the users or providing a user with false feedback. Since the users don't have control over their data transmitted on the wireless medium or stored in the middleware, anyone with internet access can access it. This puts data confidentiality, authenticity, and integrity at risk since users with unauthorized access can easily access, alter, and manipulate data in transit. The proposed Lightweight Security Algorithm (LSA) is a hybrid algorithm created by combining the Security Protocol for Sensor Networks (SPINS) with the Secure IoT (SIT) encryption technique to improve WSN's data security while lowering the threshold of attacks and minimize power consumption in WSNs without impacting network performance. Furthermore, the proposed LSA reduces the key generation time by 102mS thus improving the security by 99%. During data transmission, the power consumption is reduced by an average of 411.2uJ and the Packet Drop Ratio (PDR) is between 90 and 99% when comparing it with SPN and Feistel techniques.

Index Terms—IoT, WSN, security, encryption, SIT, SPINS

I. Introduction

Data security is becoming increasingly important as these sensors are vulnerable to attack in a variety of ways. WSNs are made up of a ton of sensor nodes that have restricted resources due to their nature. WSNs are susceptible to a range of vulnerabilities. As a result, the quality and complexity of attacks will continue to rise day by day. Data confidentiality, integrity, authentication, and privacy are crucial security requirements because lately the attackers are skilled enough to manipulate or alter data while it is in transit, then resend the fabricated information to the users. Therefore, the data that is been transferred over WSNs must be protected from misuse, threads, and attacks [1].

Manuscript received July 15, 2022; revised December 12, 2022; accepted January 3, 2023.
This work was supported by the Tshwane University of Technology.
*Corresponding author email: ntebamahlake@gmail.com.

A. Security Challenges on WSNs in IoT

WSNs are resource-constrained in terms of computational capabilities, processing time, energy consumption, and efficiency. Most WSNs in IoT deployments are occasionally put in unsupervised hazardous environments to capture critical information and data [2], [3]. Surrendering the privacy and security of the user's critical data to users with unauthorized access is never an acceptable outcome. Moreover, the security vulnerabilities, attacks, and threats that threaten the data security that is been transmitted in WSNs on IoT, and makes users' life difficult and also put the security of their data at risk. As a result, essential security measures are required to safeguard users' crucial data. For improving the data integrity, confidentiality, and authenticity of wireless devices, cryptographic schemes such as data authentication and encryption of the packets transmitted are essential. In addition, WSNs are resource-constrained and are also organized to be prone to a range of security vulnerabilities and attacks, it's common practice to adopt a lightweight encryption technique to enhance data security without compromising the performance of the network. Lightweight cryptographic protection is required as the tons of critical data grow exponentially and is readily altered and jeopardized [4]. Lightweight cryptography can be implemented in resource-constrained devices such as sensor nodes to enhance computing time, resource efficiency, energy consumption, and security [2].

B. Lightweight Cryptographic Techniques for WSNs in the IoT

The drawbacks of WSNs in terms of restricted devices are widely debated and the necessity for lightweight cryptography seems to be in demand. Several lightweight cryptographic algorithms do not always take advantage of security-performance trade-offs. The block ciphers always perform massively better than the hash functions, and stream cipher [5]. A lightweight block cipher that is ideal for resource-constrained devices such as sensor nodes to improve data security without affecting network performance is called Substitution-Permutation Network (SPN). This form of encryption utilizes mathematical procedures. The substitution process entails combining linear and non-linear operations to examine the connection between plaintext, cipher text, and keys. SPN tinkers with the data in WSNs to improve data security,

utilizing a series of substitution boxes and permutation tables to prepare it for the next round. SPN employs a sequence of interconnected mathematical operations to secure the key's secrecy and the privacy of the data communicated from data security risks or attackers. It has additional intrinsic parallelism for confusion and diffusion, and it necessitates the inevitability of the S-box [6].

Feistel Network (FN) separates blocks into equal halves and performs dispersion on one-half of each side in each round to reduce power consumption and increase data security. Furthermore, the two halves are swapped at the beginning of each round. The decryption function of the Feistel type structure has a low implementation cost since it utilizes similar computer coding for both decoding and encoding operations, reducing power and memory requirements on sensor nodes. Using the same code for encryption and decryption also put the data security at risk, should the attacker know the key the whole encryption process will be destroyed. Feistel network structures are straightforward to integrate with resource-constrained devices, but they often require more rounds for safety reasons; hence, the more rounds, the higher the power usage. Furthermore, more rounds are more time-consuming for the encryption and decryption process, and when the encryption time increases the attackers get an opportunity to manipulate and alter the data. As a result, the algorithm's execution time becomes a concern, and cryptanalytic flaws may be exposed [6], [7].

C. Contribution of the Study

The contribution of the study is that 1) because the user has no control over the data transmitted, attacks can be carried out by detecting interaction between two nodes or between sensor nodes and sink nodes and manipulating it without the proper authorization. Therefore, authenticating the users may reduce the high level of unauthorized access to the data in transits. Moreover, encryption may help reduce the amount of harm sustained to data integrity and confidentiality. A security method is required to ensure data consistency when it is kept on the middleware and also during transmission. Different cryptographic algorithms are proposed to address the issue, however, their usage in WSNs in the IoT is debatable because the devices within the IoT environment are unsuitable for implementation due to intensive computational encoding techniques. In addition, WSNs use resource-constrained devices such as sensor nodes to collect and monitor data on the network. Thus the use of lightweight cryptography improved the security of the data transmitted without sacrificing the performance of the network. 2) By integrating SIT and SPINS will help to reduce the level of the attack, ensure only authorized users with authorized access to have access to their data, and improve data security without affecting the performance of the network. It will ensure the cryptographic primitive named: confidentiality, authenticity, and integrity are accomplished. 3) The LSA helped to improve data security and power consumption by using a limited number of rounds. In the proposed technique the cipher's real key is not used, instead, the LSA utilizes the S-box to execute non-linear and linear adjustments, resulting in key diffusion and confusion, which increases key secrecy. As a result, attackers will have a hard time obtaining the key.

The following is the structure of this paper: the background of WSNs in IoT application, along with the objectives, the problems this research study is seeking to tackle, the existing lightweight cryptographic techniques, and the contribution of the study are presented in section I. The contribution of other researchers in WSNs in IoT was emphasized in Section II. The proposed LSA's flowchart and the implementation of the proposed algorithm were addressed in Section III. The experimental setup including the evaluation findings from MATLAB that compare algorithms, namely LSA, SPN, and Feistel scheme, were also reported in this section IV. Finally, Section V wraps up the report and discusses future projects.

II. Related Work

Several investigations on cryptographic techniques to protect WSN users' data throughout the data transmission process have been accomplished in past years.

The unique technique which analyzes the achievement of data security in sensor networks utilizing both two-key and single known as the Hybrid Encryption Technique (HET) was proposed by the authors [8]. To reduce the size of the ciphertext, the HET technique utilizes the compression technique known as Lempel-Ziv-Welch (LZW) to compress the data and encodes the data using a sophisticated encoding technique called the Elliptic Curve Cryptographic (ECC) technique. By preserving security primitives such as authentication, confidentiality, and integrity, the HET technique ensures data security. The technique requires minimal time for decryption and encryption while the power usage was not measured.

The new hybrid encoding technique to provide great security with minimal key management by integrating asymmetric and symmetric algorithms was proposed by the authors [9]. The new hybrid robust encryption method proposed combines asymmetric and symmetric cryptography techniques. The proposed efficient approach for protecting wireless sensor network communication against clone node attacks. The hybrid algorithm utilizes advanced cryptographic standards such as Advances Encryption Standard (AES), Byte-oriented Substitution-Permutation Network (BSDN), and Dual Rivest-Shamir-Adleman (RSA) encryption, as well as Message Digest 5 (MD5) hash functions. The packets are split into three parts in this approach, and Dual RSA, AES, and BSDN, encoding are applied to each segment concurrently. The proposed algorithm is evaluated using parameters such as execution time, decryption, and

encryption, and the results reveal that the proposed technique outperforms the competitors in terms of encryption and decryption time but power consumption is high while it is vulnerable to other attacks. The proposed hybrid approach, which employs a 4x4 S-Box and a fixed number of rounds, will address the aforementioned issue.

Anwar and Maha [10] proposed AES and Modified Playfair Cipher (AMPC), a lightweight hybrid cryptographic scheme for WSNs. AMPC was proposed to enhance WSN security by utilizing the first technique, Diffie-Hellman, to safeguard the key exchange process, as well as two cryptographic algorithms, modified PlayFair and AES, for data security. Although security was improved, power consumption was quite high. However, the proposed algorithm's power usage during data transmission will be reduced because LSA is using a limited number of rounds, each round will use its unique key to improve security.

The algorithm is named Lightweight Hybrid Cryptography (LWHC) which integrates the "Present"and Lightweight Encryption Device (LED) Cipher with the Speck compact key scheduling technique proposed by the authors [11]. To encrypt data the PRESENT, RECTANGLE, and LED S-Boxes are used. In addition, for a faster and more reliable system the RECTANGLE S-Box was used and on the other hand for key scheduling, the SPECK technique is also utilized. The proposed technique encodes plain data of 64 bits and also performs an XOR operation with 128 bits scheduling key using a plain text of 64 bits. The standard key technique that is prone to key attacks is employed by the LED encryption technique while on the other hand, the LWHC utilizes 128 bits key scheduling technique to make it secure and lightweight against different key-related attacks. This modified technique utilizes 128-bit keys XOR with the 64 bits block of plain text to ensure its strength. However, the LWHC technique is secure and lightweight against various key attacks but not against other security attacks. Therefore, the proposed LSA is lightweight and promises to reduce the level of attacks and enhance data security by using SPINS for key validation and authenticating the users.

Yue *et al.* [12] proposed a new hybrid encryption algorithm based on wireless sensor networks followed by an analysis of existing WSNs security vulnerabilities, by integrating the characteristics of high encoding intensity of asymmetric encryption techniques and high encoding efficiency of symmetric encryption techniques. This technique first encodes the plaintext blocks using ECC of asymmetric encryption technique and AES of symmetric encryption algorithm, then utilizes the technique of data compression to obtain cipher blocks, and then to form a complete ciphertext message it connects Media Access Control (MAC) address and AES key encoded through ECC. The results of the technique's description and implementation demonstrate that the technique can minimize the total running time complexity, decryption time, and encryption time, without sacrificing security. However, the power consumption was not tested. The proposed LSA technique requires only five rounds with different keys for each round for better security and minimal power consumption.

The Hamming Residue Method (HRM) is offered as a way to protect WSNs from malicious attacks. The HRM was proposed by [13] to ensure that the confidentiality of the data is not compromised by unauthorized since the wireless network is unreliable. To detect and fix faults, Hamming codes are utilized; as a result, all communication systems are aware of these codes. WSNs are self-contained and consume less energy, and these codes can be used to secure WSNs systems without the need for extra infrastructure. The method was put to the test using the Network Simulator 2 (NS2) simulation tool. The HRM technique improves the WSNs' security, but the power consumption is higher. Furthermore, the computational complexity was not taken into account while the proposed hybrid technique will provide better security while minimizing the power consumption and computational complexity by just using a limited number of rounds without affecting the performance of the network. And each round has its key to reduce the risk of the key being exposed to attackers.

For the WSNs, a lightweight security algorithm based on Proxy Mobile Internet Protocol version 6 (PMIPv6) was proposed by the authors [14]. The proposed security authentication mechanism for WSNs was chosen to replace the inbuilt Diffie-Hellman authentication scheme in the PMIPv6 protocol. As a result, a seed-based random number session key mechanism, as well as a modified MAC address-based session key initialization protocol is proposed and evaluated. Despite the moderate security, output was poor and power consumption was low. The algorithm, however, was susceptible to saturation and differential attacks. Whilst with the proposed LSA technique adopted SPINS for key validation, authentication, and check power efficiency that ensures the secrecy of the key, each round use a different key ensuring stronger secrecy of the key. In addition, adopting SIT that uses the 4x4 S-Box for non-linear and linear transformation with mathematical function on the key expansion process, which produces confusion and diffusion to confuse the attackers will also reduce the level of attacks.

According to a review of the literature, earlier studies did suggest numerous schemes that can substantially neutralize a range of cyberattacks in WSNs. However, there is currently no balance between security, power consumption, and network performance. This research proposed an enhanced LSA algorithm to address the identified constraint and fulfill security primitives confidentiality, integrity, and authentication. The proposed technique is created by combining SIT and SPIN to achieve a robust encryption technique during data transmission in WSNs in IoT without compromising performance.

III. Proposed Algorithm

The LSA algorithm is created by combining SIT and SPINS to enhance data security within WSNs in IoT. The proposed algorithm is divided into three phases key expansion, key management, and encryption. According to the literature review, the attacker may attempt to modify or alter the data by moving closer to the targeted sensor nodes. Nevertheless, this is not the case with the proposed technique, this cannot be the case because the key management phase ensures that the user's data is protected by validating data transmitted and preventing unauthorized users without proper access from accessing the data. Furthermore, packet transmission will occur only if the network has enough power to prevent attackers from manipulating data without the user's knowledge. LSA is a symmetric block cipher key with plaintext and a 64-bit key. The key expansion involves complex mathematical operations adopted from SIT technique. The key management uses the block components adopted from SPINS for the validation process. To enhance data security through diffusion and confusion, the proposed technique's encryption procedure incorporates encryption rounds that include some mathematical functions that work on 4 bits of data. In many cases, cryptographic techniques are built to operate on 10 to 30 rounds to make the encryption phase more robust to meet the security requirements [7]. With LSA only 5 rounds will be utilized to decrease energy usage and computational complexity.

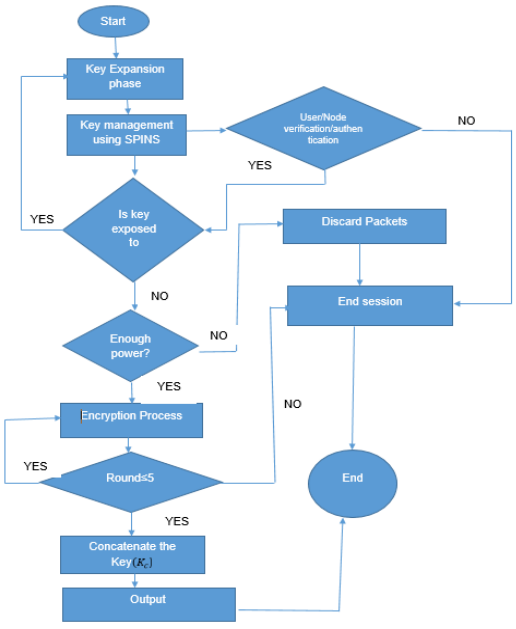

Fig. 1. LSA flowchart.

Fig. 1 depicts a high-level overview of the whole system design. The flowchart illustrates the logic behind carrying out the 3 phases of the LSA technique.

1) Key expansion

The primary mechanism for generating different keys for encoding and decoding is known as key expansion. to create diffusion and confusion lot of procedures are performed to improve the key's strength while lowering the possibility of weak keys. Based on the cipher key provided the key schedule is utilized to calculate the round keys (K_r). The process is broken down into several stages: round key selection and key generation. Logical operations (XNOR, XOR), Q-table transposition, P-table permutation, and Left shifting are performed during the key expansion process.

The most important aspect of any encoding or decoding procedure is the key. If an attacker discovers this key, the data's security is threatened. This key is responsible for the data's overall security. The proposed technique selected SIT because uses 64-bits of data with a 64-bits key to encrypt that and it is a 64-bits block cipher. The user is asked to enter a 64-bit encryption key (K_c). This key (K_c), will be utilized as the key expansion block's input. After conducting extensive operations to produce diffusion and confusion in the input key, the block will generate five different keys. According to the authors [7], the f-function in the block is influenced by the modified Khazad block cipher. Khazad is a non-Feistel cipher with a vast trial. The vast trial strategy consists of several linear and non-linear adjustments that ensure a complicated interaction between input bits and output bits. The features of the key expansion process are fully explained below:

1) The input of the 64-bit cipher key(K_C) is separated into 4-bit segments.
2) The *f*-function operates on data that is 16 bits long. Four *f*-function blocks are thus used as a result. This 16-bit for each f-function is obtained after performing an initial substitution of segments of the encryption key (K_C) as indicated in Eq. (1).

$$K_{bif} = ||_{j=1}^{4} K_{c4(j-1)+i} \quad (1)$$

where i = 1 to 4 for the first 4 round keys.

3) As stated in equation (2), by using the *f*-function to send the 16-bits K_{bif} to obtain K_{aif}

$$K_{aif} = f(K_{bif}) \quad (2)$$

4) The f-function is made up of P and Q tables from the SIT technique. As illustrated in Fig. 2, these are the tables used to execute non-linear and linear alterations to create diffusion and confusion.

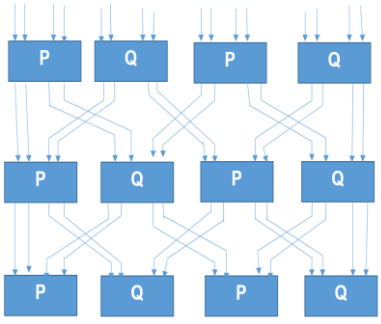

Fig. 2. SIT f-Function.

5) The matrices K_1,K_2,K_3, and K_4 are translated into four 16-bit arrays to provide round keys.
6) As indicated in equation (3) the XOR operation is used among the four keys executed from the key expansion process to obtain the fifth key. All these keys generated in this process are used in the encryption process to encrypt data.

$$K_5=K_1\oplus K_2\oplus K_3\oplus K_4 \quad (3)$$

2) Key management

Key management is the process of doing the verification and security checks on the key and packets transmitted. The authentication of the user using the MAC, and the power verification before packets can be transmitted in the network to avoid data misuse, data theft, and putting the security of data at risk take place in this process. SPINS protects sensor network communications in two ways: interactions between sink nodes and any sensor node, and interactions between sensor nodes. In a wireless sensor network, SPINS defines basic primitives for ensuring data integrity, authentication, confidentiality, and weak message freshness. It does not guarantee broadcast security [15]. It utilizes MAC for integrity and authentication, a counter for data freshness, and the counter mode for confidentiality and secrecy. It has a very minimal computational overhead, utilizes very little memory, and consumes very little power. SPINS was implemented on the proposed technique to verify and conduct security checks on the key obtained from the key expansion block (K_1,K_2,K_3,K_4, and K_5) which is used in the encryption process that is taking place between nodes or between nodes and sink. It ensures that all of the security primitive, such as data authentication, data freshness, integrity, and confidentiality, are accomplished.

Reusing the very same cryptographic key for various cryptographic primitives is a strong security design practice; this removes any possible interaction between the primitives that could cause issues. The proposed technique uses different keys for each round to make it more robust. As a result, we generate separate keys for our encryption and MAC processes. The two communicating parties, S and R, exchange a master secret key C_{SR}, from which they extract separate keys: encryption keys generated from the key expansion process $K_{SR} = F_C$ (1) and $K_{RS} = F_C$ (3) for each way of communication, and MAC keys $K_{SR} = F_C$ (2) and $K_{RS} = F_C$ (4) for each route of communication. The encrypted data is formatted as follows: E= $\{D\}_{K_{SR},C}$, where D is the data, K is the encryption key and C is the counter. M = MAC$(K, C||E)$ is the MAC.

$$R \rightarrow S: \{D\}\langle K_{RS,}C_s\rangle, MAC(K_{RS}, C_S||\{D\}\langle K_{RS,}C_s\rangle \quad (4)$$

The counter values are updated after each transmission so each time the very same message will be encrypted differently. If the counter value is sufficient it will never be repeated within the node's lifespan. Moreover, if indeed the MAC verification is successful, the receiver will then notice that the message came from the source/sender. The counter value in the MAC prevents the old massage from being replayed. If the counter was not present in the MAC, an adversary might easily replay messages. The counter state is preserved at each endpoint and does not need to be conveyed in each message, resulting in low communication overhead.

Simple SPINS only ensures a minimal amount of data freshness by ensuring a transmission order on messages inside node R, but it does not give node S a guarantee that a message was delivered by R in response to an event in node S. By using a nonce N_S, Node S ensures high data freshness for a response from Node R. Node S generates N_S, a random message and sends it to node R along with the request message R_S. In an authenticated protocol, the simplest approach to achieve strong freshness is for R to return the nonce with the response message R_R. Instead of returning the nonce to the sender, thus the proposed algorithm managed to speed up the procedure by incorporating the nonce within the MAC computation.

$$S \rightarrow R: N_{S,}\ R_S$$
$$R \rightarrow S: \{R_R\}\langle K_{RS,}C_R\rangle, MAC(K_{RS}, N_S||C_R||\{R_R\}_{\langle K_{RS}C_R\rangle} \quad (5)$$

The proposed technique ensures that the network has enough power before transmitting data. If the total energy consumption per round or total network's energy is insufficient, the packets are deleted because the weaker the network, the easier it would be for attackers to corrupt the data. If the network node is regarded alive when the energy node is more than, equal to, or greater than the minimum energy threshold, the data transmission process continues, ensuring the packets are protected from attacks.

$$E_n \geq E_{th}, \forall_n \in N \quad (6)$$

Total energy requires to transfer data from nodes to Sink Nodes (SN) or Cluster Head (CH) and data receipt at the CH or SN. It is defined as how much energy is needed for data transfer from the node to the SN. The sum of the energy consumed in each cluster is the total energy consumption each round E_{round} in the network. Below is the equation that illustrates how the energy per round is computed. The equation below shows how the energy per round is computed.

$$E_{round} = \sum_{\forall_c \in C} E_{Cluster}$$
$$E_{round} = \sum_{\forall_c \in C}\left[\sum_{\forall_n \in C}\left(E_{T_C-CH} + E_R\right) + E_{T_{SN}}\right] \quad (7)$$

By deducting the remaining energy for each node from the total energy from the network that refers to the s$E_{network}$ total network utilization. Eq. (8) shows all the steps for calculating it:

$$E_{network} = \frac{N.E_0 - \sum_{\forall n \in N} E_n}{N.\ E_0} * 100 \quad (8)$$

3) Encryption process

After the round key (K_1 , K_2 , K_3 , K_4 and K_5) validation key from the key management process, the key-encryption process begins. Some logical operations such as swapping, substitution, and left shifting to create diffusion and confusion are included in this process. For the attackers, these processes add complexity and confusion to ensure robust secrecy of the key.

1) The 16-bits sub-blocks P1, P2, P3, and P4 (PX0–15, PX16–31, PX32–47, and PX48–63) are generated from the 64-bits input block. This is to produce the segments (R1_1, R1_2, R1_3, and R1_4).
2) The swapping operation is performed to minimize data originality by shifting the order of the bits, resulting in cipher text confusion as the bits pass through each cycle. To confuse the attackers, working key keys (K_1,K_2,K_3,K_4 and K_5) validated from the key management process are utilized to address each sub-block by mixing operations from various mathematical functions, such as OR, AND, XNOR and XOR operations.
3) R1_1 is the output of the XNOR between PX 0–13 and K1.To produce EFL-1 the R1_1 feeds f-Function.
4) R1_4 is the output of the XNOR between PX 48–63 and K1.To produce EFR-1 the R1_4 feeds f-Function.
5) The product R1_4 feeds f-Function to produce EFR-1. R1-4 is the output of XNOR between PX 48–63 and K1.
6) In addition to the f-Function illustrated in Eq. 1, the f-Function contains the activities of AND, LS, substitution (S-boxes), and OR, which aids in decreasing computational complexity and computational execution time. Refer to equation 9 for more detailed information

$$F = F1 + F2 \rightarrow 32\ Bits \quad (9)$$

$$F1 = OR\left(S - Box\left(AND\left(LS\left(\frac{16\ Bits}{4}\right)\right)\right)\right) \rightarrow 16\ Bits$$

$$F2 = OR\left(S - Box\left(AND\left(LS\left(\frac{16\ Bits}{4}\right)\right)\right)\right) \rightarrow 16\ Bits$$

7) Then the XOR operation is used on results from the f-function.
8) PX 16–31 and EFL-1 are then XORed to get R1_2.
9) R1_3 is the result of XOR between PX 32–47 and EFR-1.
10) During the encryption process, a switching procedure occurs between the two internal parts. The switches are then located between (R1_1 and R1_2) and (R1_3 and R1_4).
11) The switching procedure occurred between the two internal parts during the encryption process. Between (R1_1 and R1_2) and (R1_3 and R1_4) is where the switches are then placed.
12) All of the preceding operations are supposed to enhance the complexity of the encryption, enhance data security, and reduce computational energy. Eq. (9) repeats the same processes for each round. Ensuring the key is not exposed to any possible attacks and the key secrecy is robust.

$$R_{i,j} = \begin{cases} P_{x\,i,j} \odot K_i & ;j = 1...4 \\ P_{x\,i,j+1} \oplus Ef_{li} & ;j = 2 \\ P_{x\,i,j-1} \oplus Ef_{ri} & ;j = 3 \end{cases} \quad (10)$$

13) The encoded text is then obtained using the equation

$$Ct = R_{5\_1} || R_{5\_2} || R_{5\_3} || R_{5\_4} \quad (11)$$

Because the LSA technique assures reduced computational cost, the procedure is more accurate and convenient for sending data in WSNs. To maximize the efficiency of the LSA approach, all phases will be carried out sequentially before providing user access to the user's data.

IV. Experimental Setup

A. Evaluation Parameters

To evaluate the effectiveness of the proposed LSA technique, the following criterion is used. Security, key expansion time, energy consumption, and Packet Delivery Ratio (PDR) are all factors to consider. The simulation results provided in this research are the average of ten simulations. The simulations' performance metrics concentrate on the following parameters:

1) Security: One of the most critical factors in network communication security. If security is violated, the entire network becomes vulnerable to a wide range of threats. WSNs are employed in a variety of crucial areas in IoT, and security is a major consideration.

2) Energy consumption is the overall energy consumed by the network to execute data transmission, data processing, and reception either between the sensor nodes or between sink nodes and sensor nodes.

3) Key expansion time is to the time required to execute the key expansion process; the shorter the key expansion time, the more efficient the algorithm.

4) PDR refers to the ratio of the number of packets received at the sink node or node as a destination to the number of packets sent from the nodes as a source. The PDR is demonstrated by data collisions, intermediary node failures, forced path switching, high data traffic, and intruder attacks.

B. Experimental Simulation and Evaluation Results

The implementation of the LSA is been presented in this section. The MATLAB version 2022a using the IEEE 802.11s simulation model is used to implement the proposed LSA technique. A Windows 10 machine with 8GB of RAM and an i5 Core CPU was utilized as the computing platform. In a 400*400 square meter space, the radius of 30 is used to arbitrarily distribute 100 sensor nodes; then the arrangement of the nodes distribution is clustered and random. Mesh routers are often not placed close together in the real world. As a result, the interconnection is set to 1 unit, with the shortest distance of 0.3 cm between nodes.

C. Simulation Results

The simulation results provided in this paper are the average of ten simulations. This section utilizes graphical illustrations to demonstrate and analyses the simulation outcomes performed to enhance the effectiveness of the LSA in a variety of network arrangements. The MATLAB results that were evaluated were displayed using the Xgraph. This research compares the proposed LSA algorithm's effectiveness with the commonly known techniques SPN and Feistel techniques.

Because the proposed technique utilizes symmetric block cryptography, it is compared to other lightweight cryptographic algorithms such as SPN and Feistel. Each technique has its unique way of key generation but the message is encrypted with a unique key generated from the key generation process. The proposed algorithm has an adopted security protocol for key validation and user authentication to ensure the user's data is protected from attacks before the key can be used to encrypt the data. Furthermore, when utilizing lightweight cryptography, the most serious issue is secure key distribution. This research compared the proposed LSA algorithm to cryptographic techniques based on the use of keys to encode/decode messages and the power required for data transfer from sender to receiver. And ensure the secrecy of the key is robust. However, the proposed technique took user authentication, integrity, and confidentiality into account by combining both SIT and SPINS to make sure that the encoding is quicker, there is sufficient power in the network to avoid data loss, and also the network is not targeted when delivering data across the wireless medium. In addition, it ensured that only fresh and new that is transmitted, and no old data replay was on the wireless network. Furthermore, the data security is improved and the level of attacks is minimized.

1) Key expansion time

Fig. 3 illustrates a comparison of the lightweight encryption techniques' key expansion times. The bit length determines the key expansion time, which is the time it takes to generate a key. The higher the length, the longer the time. The findings indicate that LSA uses less time than SPN and Feistel. LSA took a minimum time of 50mS on a key length of 8 bits to generate the key and a maximum of 102ms on a key length of 256 bits to generate the key. The Feistel comes in second with a maximum of 115mS on a key length of 256 bits.

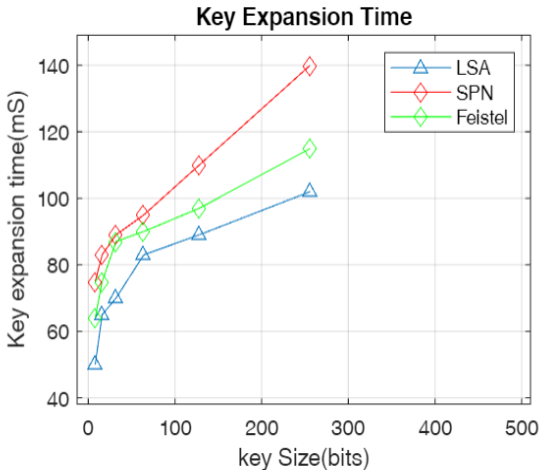

Fig. 3. Key expansion time (LSA, SPN, and Feistel) for 100 nodes.

As illustrated in Fig. 3 when the key length is 8 bits the SPN took a minimum of 75mS to generate the key and at a maximum of 256 bits key length, it took 140mS to generate the key. Because SPN has a complicated and longer key expansion process that uses mathematical operations and the time taken to execute is very long and the secrecy of the key is at risk. The attacker has a lot of time to easily manipulate or alter the key and resend it if the execution time is longer, this will affect the security of the data. The LSA outperformed both SPN and Feistel which indicates LSA provides better data security with less key expansion time.

2) Security for 100 nodes

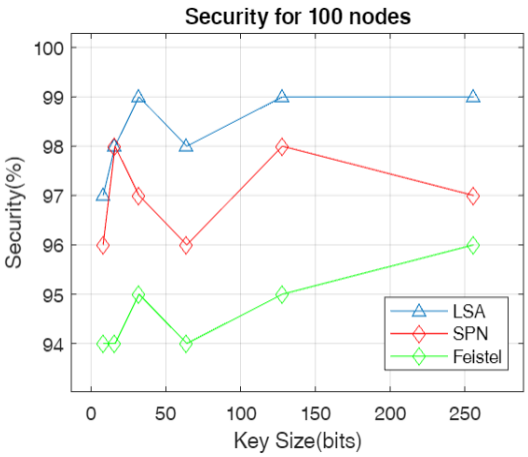

Fig. 4. Overall network security for 100 nodes.

One of the most critical key performance indicators in network communication is security. If security is jeopardized, the entire network may be at risk. Security is the most critical aspect to consider especially in the context of WSNs in IoT being utilized in sensitive locations. The size of the key is very important when it comes to confidentiality and secrecy level. There should be no faster attack than an exhaustive key search for a secure block cipher. Because exhaustive key search takes much longer for a bigger key size, a theoretical attacker

may afford to put in more effort to "crack" the larger cipher. As illustrated in Fig. 4 for key lengths from 8 bits to 256 bits during data transmission, LSA maintains a security level of 97% to 99% because after the key is generated, the key is validated in the key management process ensuring the secrecy of the key is stronger and it is not exposed to any kind of attacker. In addition, only authorized users with authorized access can have access to the data in transits. To protect the confidentiality and secrecy of data, any key length can be utilized with a significant number of rounds. LSA has five rounds indicating higher security against cryptanalysis because there is more confusion and diffusion.

SPN was the next best performer, with a level of security for 8 bits to 256bits ranging between 96% to 98%. Feistel was the least performer with a security range of 94% to 96% for 8 bits to 256 bits. However, the SPN and Feistel do not have the key management process to authenticate the users and check if the key is exposed to any attacks or do some validation on the key, which is why the secrecy of the key is seen to be at risk. This increase the chances of attacks such as key-related attacks. The secrecy of the key is one of the most important aspects, should the attackers have access to the key that simply means the security of the data especially the integrity and confidentiality of the key will easily be compromised. In all 10 simulations, the proposed LSA has outperformed the other techniques in terms of security.

3) Energy consumption for 100 nodes

One of the most critical factors, when there are a huge number of battery-powered network nodes, is energy consumption. Energy consumption is a critical metric because IoT-based WSNs nodes are generally battery-operated devices. If the is insufficient power on the network the data transmitted become more prone to data thefts, threats, and misuse. It is measured using Micro-Joule (uJ). In all 10 simulations, LSA uses less energy than other techniques. LSA consumed 386uJ for data transmission when simulated with 10 nodes. When 100 nodes are utilized, the highest energy used by LSA was 478uJ. LSA's average energy usage was 411.2uJ, which is one of the lowest compared to other techniques. SPN gets an average energy consumption of 781.6uJ and is rated as the second competitor.

However, SPN uses complicated mathematical procedures during key expansion and the number of rounds generated is made up of key mixing, substitution layer, and permutation layer that increase the power usage. Since Feistel requires a lot of rounds for robust data security and ensuring the secrecy of the key is strong, it has a higher power utilization of 705uJ to 924uJ for 10 to 100 nodes. The average energy consumption is 795.3uJ for all 10 simulations. The lesser the number of rounds the energy consumption decreases. The proposed algorithm also checks if the network has enough power before the data can be transmitted to avoid data loss. The results showed that LSA energy consumes less energy compared to the SPN and Feistel

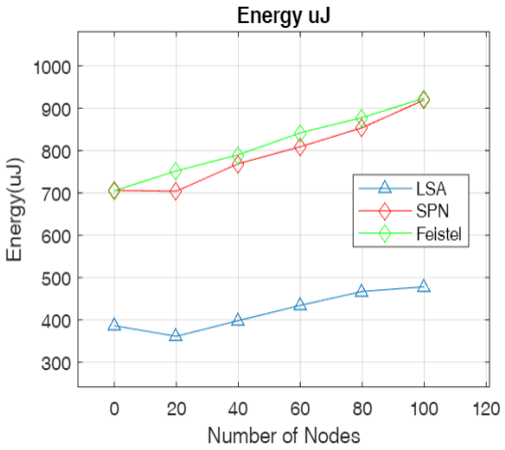

Fig. 5. Energy consumption for 100 nodes.

4) PRD for 100 nodes

PDR is defined as the ratio between the number of packets sent from the source node to packets successfully delivered to the destination node. The high rating of PDR indicates minimal packet losses and data collisions. The PDR has maintained between 90% and 99% percent when using the proposed LSA. LSA has a PDR of 94.4% on average. The second-best PDR results were obtained by SPN, with a minimum of 78%, a maximum of 95%, and an average of 88.6%. The least performed algorithm is Feistel with a minimum of 75%, a maximum of 96%, and an average of 82.5%. The simulation results showed that the LSA is more fault-tolerant when it comes to security threats. The highest ratio symbolizes the low number of packet drops and the low number of data collisions. Thus, the number of possible attacks on the data transmitted in WSNs in IoT was reduced (see Fig. 6).

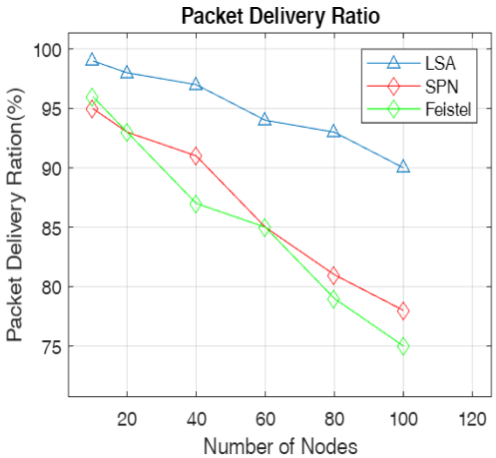

Fig. 6. The overall packet delivery ratio for 100 nodes.

V. Discussion

Recent advanced cybersecurity or cryptography can be divided into two categories: symmetric and asymmetric. Symmetric ciphers have shorter key lengths than

asymmetric algorithms, making them less secure due to their lower complexity. On the other hand, asymmetric ciphers, for it to secure the IoT communication network uses greater complexity, but the longer key length makes them inefficient. After considering all of these important factors, it is necessary to develop an algorithm that would use minimal power, require less time, provide the first and acceptable security to low-end IoT devices and also reduce complexity.

TABLE I: A COMPARISON ANALYSIS WITH PREVIOUS TECHNIQUES

Author	Technique	Cipher and type of network	Key size, Block size, and rounds	Features
Noura *et al*. [16]	ORC	SPN, FN	One round	Resistant to sensitivity tests, statistical analysis, and visual degradation.
Chatterjee and Chakraborty [17]	Modified PRESENT	SPN, FN	80 bits key, 64 bits plain text,25 round	Better performance
Shantha and Arockiam [18]	SAT_ Jo	SPN, Block Cipher	80 bits key, 64 bits block,21 round	For resource-constrained IoT devices, this technique provides a better combination of performance and resource constraints.
Ahmed *et al*. [19]	G-TBSA	WSN		It is suitable for WSNs and provides lower power consumption
Hamzaab *et al*. [20]	Chaos_Based PRNG	Symmetric, Block Cipher		Resistant to different attacks like statistical, exhaustive, and differential attacks for locating secret keys
	Proposed technique	SPN, FN, Symmetric Block Cipher	64 bits block,64 bits key size, 5 rounds	This technique provides a better combination of performance and security with lower power consumption

Table I illustrates a comparison with some of the existing techniques. ORC is an SPN and FN technique that is resistant to statistical analysis, however, it has a relatively higher latency. The modified PRESENT technique used 25 rounds of the technique, requiring less computational power with better performance. However, it is resistant to key attacks but vulnerable to others. G-TBSA requires minimal power but is only appropriate for WSNs. The proposed algorithm and SAT-Jo, and are block cipher techniques that are appropriate for resource-constrained devices in an IoT environment. Chaos_Based PRNG is resistant to different attacks like statistical, exhaustive, and differential attacks for locating secret keys. The proposed technique provides a better combination of performance and security with lower power consumption for resource-constrained devices.

VI. CONCLUSION AND FUTURE WORK

This paper presents the implementation of the LSA technique by combining SIT and SPINS to enhance the data security of WSNs in IoT. The proposed LSA algorithm's effectiveness in WSNs is compared to the SPN and Feistel techniques. The findings of the analysis are reported based on the network parameters including security, key expansion time, PDR, and power consumption. The simulation results demonstrate that the proposed LSA technique surpassed the commonly known lightweight cryptographic techniques SPN and Feistel techniques in all of the above-mentioned important network parameters. Moreover, during data transmission, the power consumption is reduced by an average of 411.2uJ and the Packet Drop Ratio (PDR) was between 90 and 99%. The results indicate that the proposed LSA reduces the key generation time by 102mS thus improving the security by 99% when comparing it with SPN and Feistel techniques.

Because the attention of this paper is on enhancing data security without compromising network performance. It is assumed that the computational time is not sustained during the key expansion and encryption process. And the performance of the network is not taken into consideration. Furthermore, the computational or execution time that covers the key expansion speed will be taken into account in the future to ensure robust data security in WSNs. Additionally, ensure that the effectiveness of the LSA technique is evaluated in a more sophisticated network environment.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Ntebatseng Mahlake contacted the research and wrote the paper. Topside E. Mathonsi, Deon. Du Plessis, and Tonderai Muchenje analysed data. All the authors approved the final version of the paper.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Tshwane University of Technology for financial assistance and confirm that there is no conflict of interest in this paper.

REFERENCES

[1] M. A. Burhanuddin, A. A. Mohammed, R. Ismail, M. Hameed, A. N. Kareem, and H. Basiron, "A review on security challenges and features in wireless sensor networks: IoT perspective," *Journal of Telecommunication, Electronic, and Computer Engineering*, vol. 10, pp. 17-21, 2018.

[2] D. Aakash and P. Shanthi, "Lightweight security algorithm for wireless node connected with IoT," *Indian Journal of Science and Technology*, vol. 9, no. 30, pp. 1-8, May 2020.

[3] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges, and solutions," *Journal of Ambient Intelligence and Humanized Computing*, May 2017.

[4] P. Tiwari, V. P. Saxena, R. G. Mishra, and D. Bhavsar, "Wireless sensor networks: Introduction, advantages, applications and research challenges," *HCTL Open International Journal of Technology Innovations and Research*, vol. 14, pp. 1-11, April 2015.

[5] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison, and research opportunities," *IEEE Access*, vol. 9, pp. 28177-28193, 2021.

[6] D. Sehrawat and N. S. Gill, "Lightweight Block Ciphers for IoT based applications: A review," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 2258-2270. 2018.

[7] M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "SIT: A lightweight encryption algorithm for secure internet of things," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, April 2017.

[8] A. Tripathy, S. K. Pradhan, A. K. Nayak, and A. R. Tripathy, "Hybrid cryptography for data security in wireless sensor network," in *Data Engineering and Intelligent Computing. Advances in Intelligent Systems and Computing*, V. Bhateja, S. C. Satapathy, C. M. Travieso-González, V. N. M. Aradhya, Eds., vol. 1407, Springer, Singapore, 2021.

[9] L. Sachin, S. Bhushan, and Surender, "Hybrid encryption algorithm to detect clone node attack in wireless sensor network," in *Proc. International Conference on Innovative Computing & Communications*, 2020, pp. 1-6.

[10] N. B. Anwar and M. M. Maha, "AMPC: A lightweight hybrid cryptographic algorithm for wireless sensor networks," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 6, pp. 1142-1146, June 2020.

[11] V. Prakash, A. V. Singh, and S. K. Khatri, "A new model of light weight hybrid cryptography for internet of things," in *Proc. 3rd International Conference on Electronics, Communication and Aerospace Technology*, September 2019, pp. 282-285.

[12] T. Yue, C. Wang, and Z. X. Zhu, "Hybrid encryption algorithm based on wireless sensor networks," in *Proc. IEEE International Conference on Mechatronics and Automation*, August 2019, pp. 690-694.

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

[14] A. Anandhavalli and A. Bhuvaneswari, "Lightweight security algorithm on PMIPV6 protocol for IOT based wireless sensor networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 1790-1799, October 2019.

[15] F. Ullah, T. Mehmood, M. Habib, and M. Ibrahim, "SPINS: Security protocols for sensor networks," in *Proc. International Conference on Machine Learning and Computing*, January 2011, vol. 3, pp. 333-337.

[16] H. Noura, A. Chehab, L. Sleem, M. Noura, R. Couturier, and M. M. Mansour, "One round cipher algorithm for multimedia IoT devices," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 18383–18413, September 2019.

[17] R. Chatterjee and R. Chakraborty, "A modified lightweight present cipher for IoT security," in *Proc. International Conference on Computer Science, Engineering and Applications*, March 2020, pp. 1-6.

[18] M. J. R. Shantha and L. Arockiam, "SAT_Jo: An enhanced lightweight block cipher for the internet of things," in *Proc. Second International Conference on Intelligent Computing and Control Systems*, 2019, pp. 1146-1150.

[19] S. F. Ahmed, M. R. Islam, T. D. Nath, B. J. Ferdosi, and A. S. M. T. Hasan, "G-TBSA: A generalized lightweight security algorithm for IoT," in *Proc. 4th International Conference on Electrical Information and Communication Technology*, 2019, pp. 1-6.

[20] R. Hamza, Z. Yancd, K. Muhammade, P. Bellavistaf, and F. Titouna, "A privacy-preserving cryptosystem for IoT E-healthcare," *Information Sciences*, vol. 527, pp. 493-510, July 2020.

Ntebatseng Mahlake was born in Limpopo Province, South Africa, in 1990. She received the National Diploma in Information Technology specializing in communication networks from the Tshwane University of Technology (TUT) in 2014 and the Btech in Information technology specializing in communication networks for the Tshwane University of Technology (TUT) in 2018 both in the Faculty of Information Communication Technology. She is currently pursuing a Master's degree with the Department of Information at TUT.

Her research interests include WSN security in IoT, encryption algorithms, and data security.

Topside E. Mathonsi received his B-Tech in 2013, M-Tech in 2015 and DComp in 2020 from the Tshwane University of Technology, Pretoria, South Africa. He is currently the Senior Lecturer at Department of Information Technology, Tshwane University of Technology, South Africa. His research interests include 5G, IoT/IoE, Cybersecurity, and AI.

Dr. Tonderai Muchenje is a lecturer at Tshwane University of Technology, in the Department of Information Technology. He holds a PhD in Information Technology (Nelson Mandela University), MCom Computer Auditing (University of Johannesburg), MSc in Computer Science (University of Fort Hare), Certified Trainer CCNA1, Certified Trainer CCNA2: Routing, Switching and Wireless Essentials, Certified Trainer: CCNA Cybersecurity Operations, Advanced Business Analysis (University of Pretoria) Currently lecturing computing systems and information security and supervising Masters at TUT. My research areas of interests are Wireless networks security, Internet of Things, Cloud Computing, and Smart technologies.