Journal of Information Security, 2025, 16(2), 313-329
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

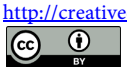
Cryptographic Algorithm Grafted onto the ESP-NOW Protocol for Security Optimization in Wireless Sensor Network Chains

Laura Motso Chatue[1], Roland Christian Gamom Ngounou Ewo[1], Leandre Nneme Nneme[1], Nelson Dada[2], Valdez Wilsons Fotso Tekam[2]

[1]Laboratory of Computer Science Engineering and Automation, Higher Normal School of Technical Education of Douala, University of Douala, Douala, Cameroon
[2]Laboratory of Computer Engineering, Data Science and Artificial Intelligence, National Higher Polytechnic School of Douala, University of Douala, Douala, Cameroon
Email: lauraassyerha@gmail.com

How to cite this paper: Motso Chatue, L., Gamom Ngounou Ewo, R.C., Nneme Nneme, L., Dada, N. and Fotso Tekam, V.W. (2025) Cryptographic Algorithm Grafted onto the ESP-NOW Protocol for Security Optimization in Wireless Sensor Network Chains. *Journal of Information Security*, **16**, 313-329.
https://doi.org/10.4236/jis.2025.162016

Received: March 17, 2025
Accepted: April 20, 2025
Published: April 23, 2025



Abstract

In this paper, we present a secure routing scheme based on the ESP-NOW protocol of ESP32 microcontrollers. In addition to security specific to this protocol, the data transmitted is encapsulated in a symmetric cryptography algorithm based on binary and hash cryptographic functions. A unique encryption and decryption key is used when sending and receiving data. The algorithm is written in C++ program in the form of a library and included in an Arduino sketch. The sensor node calls the library for encryption and the central node uses this same library to decrypt the message received and restore the encrypted information. The tests are carried out on a network made up of two ESP 32, each coupled with a DHT11 sensor which sends the information to a central node for control and monitoring. The results obtained optimize data security in networks based on ESP-NOW communication protocols. The goal is to contribute to the reinforcement of data transmission security for a sensor network within a logistics transport chain equipped with temperature and humidity sensors.

Keywords

Symmetry Cryptography, Encryption, Decryption, Wireless Sensor Network, ESP-NOW, Security Protocol

1. Introduction

Electronic data interchange is increasing rapidly, thus making it easier to access data [1]. Wireless sensor networks (WSNs) have attracted a lot of interest in recent

years. It is expected that WSNs will be applied in our daily life, in healthcare [2], video surveillance, real-time target tracking, and home surveillance [3]. These applications require the deployment of multiple sensors to cover a given region of networking interest, while establishing ubiquitous wireless sensor networks that will pervade society redefining the way we live and work [4]. The proliferation of wireless sensor networks has paved the way for a multitude of research fields which constitute the privileged sites for their deployment. The interest raised by this multitude of investigations creates ideal conditions for large fields of deployment in the near future. Nevertheless, many obstacles inherent in their specificities must be overcome before they can reach maturity. Among these obstacles, the problem of security is acute and must be solved appropriately and in accordance with the particular characteristics of WSNs. These constraining characteristics are observed in the limitation of resources such as: energy, computing power, bandwidth and memory space. Due to these constraints and their deployment in unattended and hostile environments, the various sensor nodes of an WSN are vulnerable to compromise and susceptible to physical breach. Moreover, the use of wireless transmissions makes the WSNs permeable to malevolence of all kinds, and constitutes a real security challenge to be taken up. Having sensors and their applications in the military and business increases the demand to look for ways to secure the flow of data using cryptography which is the technique used to establish secure communication between two parties in the public environment where unauthorized users and malicious attackers are present [5]. Cryptography is a well-known mechanism in which asymmetric or symmetric algorithms using public or private keys provide secure data communication [6]. The purpose of cryptography is to provide confidentiality, authenticity and integrity of data. There are two types of cryptographic techniques, namely symmetric key and asymmetric key [7]. In the first case, two communicating entities share a secret key and messages are encrypted and decrypted at either end using the same key while in the second, each entity has its own private key (secret key) and its public key [7]. Binary encryption much more used in the encryption of images, can help in other areas through its techniques and its operation [8]. Thus, there are many works highlighting cryptography in the WSNs [9]-[13]. Although several works have been done in this area, they are less successful in providing both high security and low computational complexity. The cryptography technique chosen for the WSNs must be appropriate to meet the limitations or constraints of the WSNs such as power consumption, memory, as well as the encryption/decryption times and the lifetime of the WSNs. Some public key cryptographies used in WSNs are the Diffe-Hellman Key Agreement Protocol or RSA [14] signatures. Asymmetric key encryption algorithms help preserve the authenticity and confidentiality of data. However, they are computationally intensive as they consume more energy and power; therefore, symmetric key algorithms are used to encrypt data in WSNs in accordance with the limited power source in the sensor device [15] [16] hence private key cryptography is preferred over public key cryptography [17]. We thus notice an overwhelming use of symmetric cryptography algorithms for WSNs like the AES al-

gorithm. The latter is used in many WSN works. I. Sultan, B. J. Mir, and M. T. Banday in [18] performed a simulation analysis on the effect of changing key size and mode types of AES encryption. The results obtained do not take into account the nature of the transmission media which can considerably affect this result, but also the simulation results obtained, which are far from realistic and practical results. Another article that focuses on the comparison between hardware and software implementation of AES encryption is presented in [19]. In this article, the authors studied the power consumption of the AES algorithm for different modes. However, neither the effect of changing the key size, nor the effect of the communication media or the encryption done by the wireless module on the energy consumption has been studied. Other works like those of [20] make AES software optimization modifications where the main objective was to propose an optimized code capable of encrypting/decrypting information at a speed comparable to the speed of the communication module. ZigBee (close to 250 Kbps) [21]. The interest of this article is to contribute to the literature by implementing a lightweight symmetric cryptography algorithm comparable to AES, consuming less resource and memory, for data security and optimization of data transmission and reception times. In our particular case, we will focus on the ESP-NOW transmission protocol on which our algorithm will be grafted in order to optimize the level of security. As shown in **Figure 1**, symmetric cryptography represents the foundation of our approach with a shared key for both encryption and decryption processes. We therefore propose a custom algorithm for WSN that will ensure data confidentiality by protecting the network against various attacks thanks to a double security algorithm. Obviously, this research will enrich the literature on data security. The rest of the article is organized as follows: the second section presents the architecture of the proposed system; the third section presents the results and the analysis of the security performances and then a conclusion.

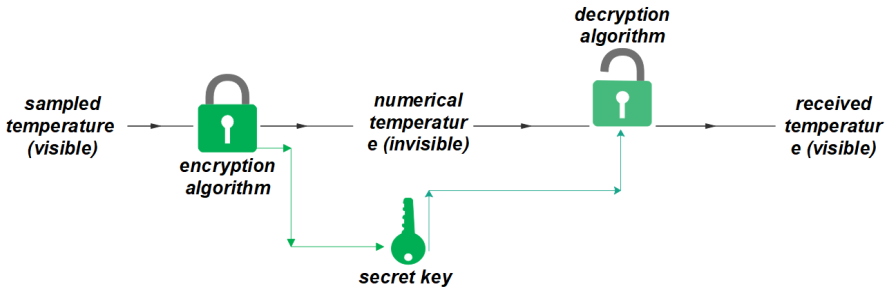


Figure 1. Symmetric cryptography.

2. Architecture of the Proposed System

2.1. The Proposed Solution

To overcome the problem of securing communications within wireless sensor networks, we propose in this article an encryption/decryption algorithm based on symmetric cryptography with a unique encryption/decryption key, which consti-

tutes one of its fundamental concepts. We apply this security within a network of wireless sensors based on the ESP_NOW protocol of ESP32 microcontrollers by Espressif for real-time monitoring of the temperature and humidity parameters of a supply chain.

2.2. Hardware

- **ESP32**

ESP32 is a low-cost microcontroller which succeeded to the famous ESP8266 designed in 2016 by Espressif System. This microcontroller has proven its capability for Wi-Fi-automation as a working solution. This controller features a variety of its company's features and supports to be adopted for a wide range of applications ranging from IoT to audio streaming. Thus, its specifications offer an extreme arsenal for programmers with a 240 MHz dual core, 520 KB of RAM and peripherals including ADC, DAC, I2C, UART, SPI, GPIO and costing only $10. Additionally, the power management unit and ultra-low-power coprocessor enable ESP32 to operate at 1mA in hibernate mode. These features make ESP 32 an excellent choice for low power applications. In terms of usability, ESP32 can be programmed on several development tools such as Arduino-IDE, AT command or by using Espressif's officially open development framework based on FreeRTOS [22].

- **DHT11**

The microcontroller measures temperature from the digital integrated circuit (DIC) sensor DHT11 which detects the humidity and temperature of the environment. This sensor can measure a temperature range of 0˚C to 50˚C and a humidity range of 20% to 90% with an accuracy of ±1˚C and ±1% [23]. It works with operating voltages from 3.5 V to 5.5 V and operating currents up to 0.3 mA. The DHT11 sensor module gives the measured temperature value in Celsius degrees. When an ambient temperature increases by one degree Celsius, the sensor generates a voltage of 10 mV. **Figure 3** shows the schematic diagram of the proposed system on the sensor node side.

2.3. The ESP-NOW Protocol

ESP-NOW is a kind of connectionless communication protocol defined by Espressif [24]. Compared to normal Wi-Fi, this method is more energy efficient and faster to deploy with a range of 200 to 220 meters outdoors. Inside the ESP-NOW network, all devices can communicate by 3 main methods: broadcast, unicast and multicast with a data rate of 1 Mbps and above. While unicast and multicast require initial pairing, acknowledgment response and a limited number of receivers up to 20 [22], in this protocol application data is encapsulated in a vendor-specific action frame and then transmitted from one Wi-Fi device to another without a connection. The CTR with the CBC-MAC protocol (CCMP) is used to protect the action framework for security. The format of the vendor-specific action framework is shown in **Figure 2**.

Mac Header	Category Code	Organization Indentifier	Random Values	Vendor Specific Content	FCS
24 bytes	1 bytes	3 bytes	4 bytes	7-255 bytes	4 bytes

Figure 2. ESP-NOW protocol frame format.

This frame is transmitted by ESP-NOW using the CCMP security method, which is described in IEEE Std. 802.11-2012, to protect the vendor-specific action framework. The Wi-Fi device maintains a master key (PMK) and several local master keys (LMK). The lengths of PMK and LMK are 16 bytes.

- PMK is used to encrypt LMK with AES 128 algorithm.

Call esp_now_set_pmk () to set PMK. If PMK is not defined, a default PMK will be used.

- LMK of paired device is used to encrypt vendor-specific action frame with CCMP method.

The maximum number of different LMKs is six. If the paired device's LMK is not set, the vendor-specific action frame will not be encrypted. Multicast provider-specific action frame encryption is not supported.

2.4. Description of the Proposed Algorithm

The algorithm we propose is based on the principles of binary cryptography and cryptographic hash leading to a condensed message which corresponds to a fixed size on 10 bits. Like any symmetric algorithm, it consists of a unique 8-bit secret key present in the encryption and decryption programs. The encryption process is as follows:

- Sampling of encrypted value. It is a real variable with two decimal places. Example $n.\ m$ degrees Celsius for a temperature value. n being the whole part and m the decimal part. They are numbered identically. $n.\ m = n + m \times 10^{-2}$
- We define an 8-bit binary key. Key = 173 or $(10101101)_2$ in binary.
- We define (k)2 as the result of the operation coded on 8 bits.

$$(k)_2 = \overline{(n)_2 \ \mathbf{XOR} \ (key)_2}$$

- So we have

$$(k)_2 = (a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0) \ avec \ a \in \{0,1\}$$

- k being on 8 bits, we decide to dissect it into sections of two-two bits going from LSB to MSB, knowing that on two bits the maximum value is $(2^2 - 1) = 11$. We obtain numbers belonging to {00, 01, 10, 11} and decide to set up the following correspondences:

$$\begin{cases} 00 = "\$" \\ 01 = "@" \\ 10 = "\#" \\ 11 = "\%" \end{cases}$$

• Depending on the correspondences with k, these values enable us to obtain a new 4-bit character string type variable defined as follows:

$$m = \sum_{i=0}^{3} a_i \tag{1}$$

• We define a 15-element string array as follows: Tab = {"1", "^", "+", "*", "4", "/", "-", "&", "_", "=", "]", "{", ">", ".", ":"}.

• The final value will include a combination of the values of m with those of the preceding array according to an algorithmically defined hash function.

• The decryption process works according to a decomposition of the string received during the encryption of the data which will lead to integer values which will be combined by identification of the whole and decimal parts as shown in Figure 7.

To mitigate key compromise risks in our symmetric encryption scheme, we leveraged the ESP-NOW protocol for its inherent security features, including AES-CCM 128-bit encryption [24]. The 8-bit secret key is embedded within the ESP32 firmware and never transmitted over the network, while flash encryption safeguards against firmware extraction attempts, reinforcing physical security. The MAC address-based authentication further restricts network access to pre-authorized devices. Additionally, a key rotation mechanism allows periodic updates without disrupting operations, maintaining both security and computational efficiency. This multi-layered approach was a key factor in selecting ESP32 and ESP-NOW for our implementation.

2.5. System Implementation

Our task consists in networking several communicating sensor nodes with a master node which is the point of reception and monitoring of the various data that can be stored in a database or routed on a website for example. The sensor is responsible for acquiring data which is then transmitted to the microcontroller, and then to the central node via the ESP_NOW protocol. We therefore have the diagrams of the figures which present the synoptic architecture of the points of our network. Figure 3 presents the sensor node diagram, showing the essential components and connections required for data acquisition and transmission in our network architecture.

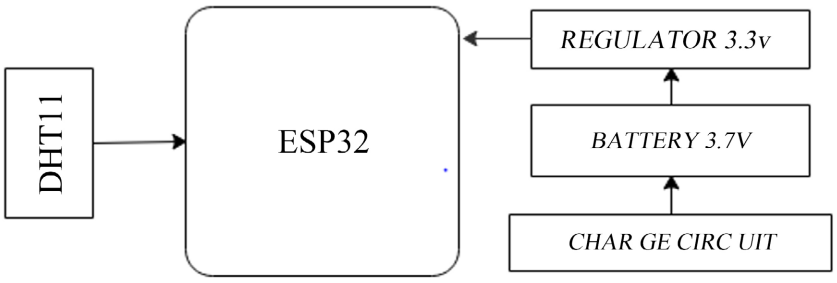


Figure 3. Sensor node diagram.

The central node with its monitoring capabilities is depicted in Figure 4, illustrating how data from multiple sensor nodes is received and processed.

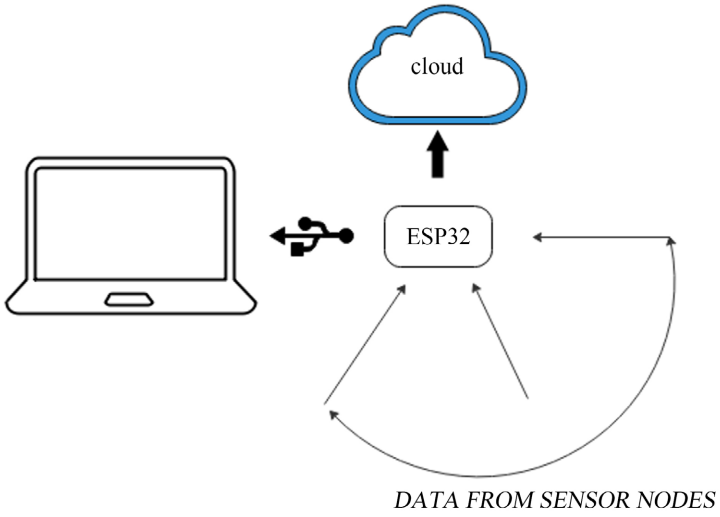


Figure 4. Central node with monitoring possibilities.

Communication from one node to another is done through the protocol. Establishing this communication requires to set the code to be transmitted in the ESP32 microcontroller and this by including the esp_now.h library. The sensor nodes couple to the central node by adding its mac address to their code. A node that does not have the mac address of the central node cannot therefore communicate with it. The process of establishing connections between nodes in the network is shown in Figure 5, demonstrating how MAC addresses are used for secure pairing.

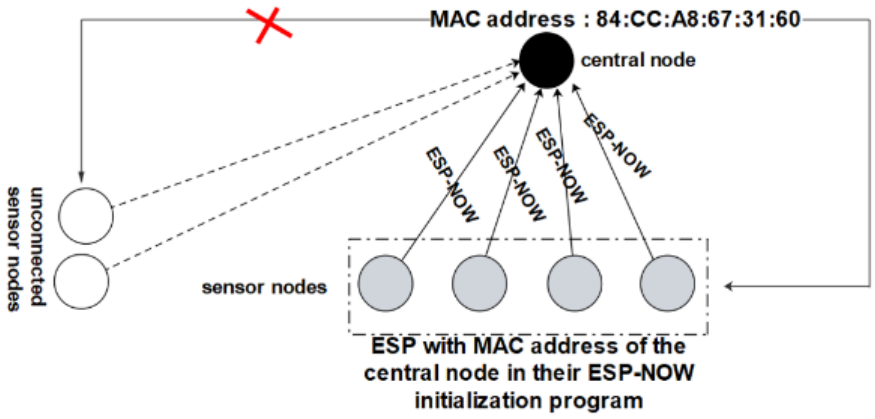


Figure 5. Process of connecting nodes in the network.

Since a level of security is already included in the protocol, we add an algorithm to it. This algorithm is responsible for strengthening it in order to make our system more robust against external attacks. The data being encapsulated in the frame format, the diagram of the level of security is illustrated in Figure 6.

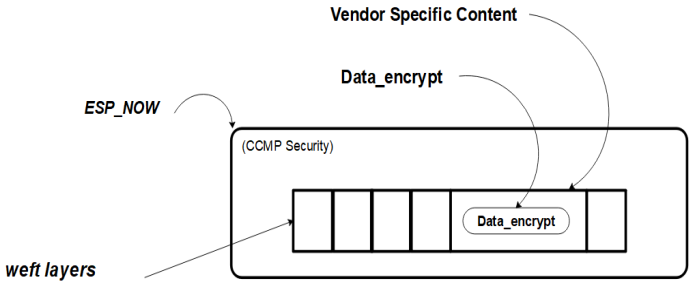


Figure 6. Level of communication security in the designed network.

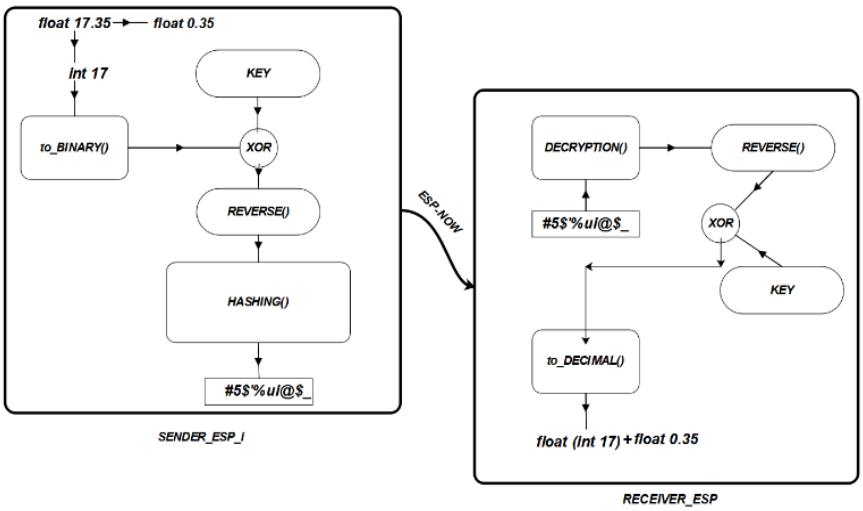


Figure 7. Algorithmic description of encryption and decryption processes in the transmission network from a sensor node to the central node.

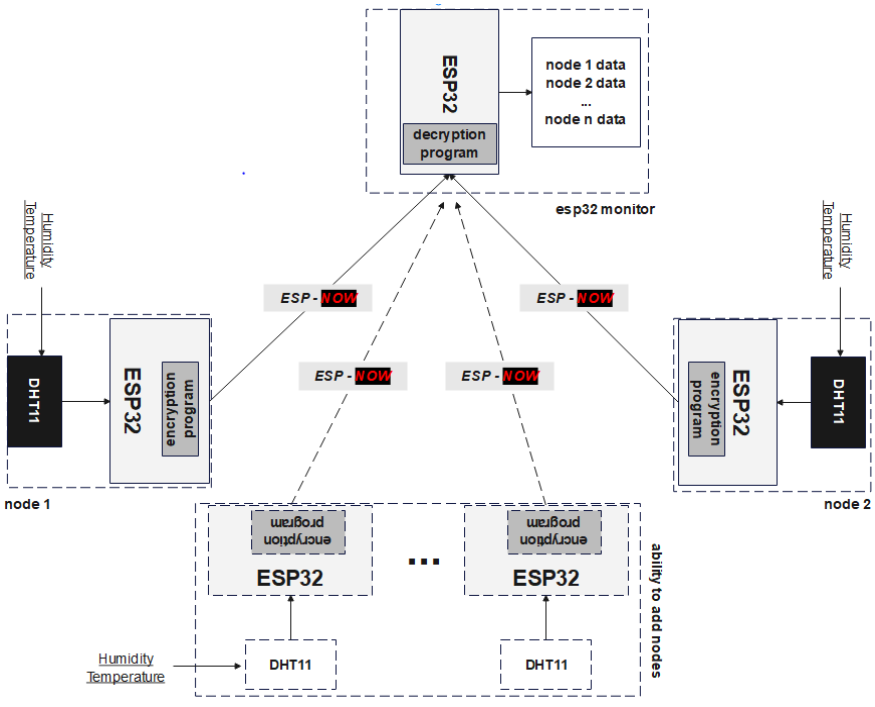


Figure 8. Functional system proposed.

In this illustration Data_encrypt represents the outgoing data of the encryption algorithm. It will be decrypted upon receipt thanks to a secret key in the receiving node (or central node) decryption program which is responsible for monitoring all the nodes. The complete algorithmic flow of encryption and decryption processes between sensor nodes and the central node is detailed in Figure 7, highlighting the key transformations applied to the data.

The general architecture of the communication flow is presented in Figure 8. As shown in Figure 5, it is possible to add nodes in the circuit, the communication process will remain the same as well as the established security.

3. Results and Discussion of Safety Performances

Our security assessment rigorously examined multiple attack vectors, validating the resilience of our architecture. The AES-CCM 128-bit encryption, reinforced by our custom key strategy and multi-stage hashing, forms a strong cryptographic defense. Replay attacks are mitigated through timestamp validation and ESP-NOW's rapid, intermittent transmissions, reducing interception opportunities. The protocol's independence from WiFi access points eliminates exposure to SSID spoofing and rogue AP attacks.

```
Initial value: 17.75C
---------Decimal to binary-----------
extraction of the whole part: 17
The binary value on bits 8 is : 00010001
---------Decimal to binary-----------
extraction of the whole part: 17
The binary value on bits 8 is : 00010001
-----------XOR and Inverse---------
Key value = 10101101
10101101 xor 00010001 = 10111100
reverse value: 01000011
---------Decimal to binary-----------
extraction of the whole part: 17
The binary value on bits 8 is : 00010001
-----------XOR and Inverse---------
Key value = 10101101
10101101 xor 00010001 = 10111100
reverse value: 01000011
---------Hashing-----------
idx0: 01 idx1: 00 idx2: 00 idx3: 11
idx0: @ idx1: $ idx2: $ idx3: %
value after hashing: @$$%
value after hashing and combining: #5$'%ui@$_
---------Hashing-----------
idx0: 01 idx1: 00 idx2: 00 idx3: 11
idx0: @ idx1: $ idx2: $ idx3: %
value after hashing: @$$%
value after hashing and combining: #5$'%ui@$_
--------- Decryption -----------
Received value: #5$'%ui@$_
idx0: @ idx1: $ idx2: $ idx3: %
Valeur inverse: 01000011
Real starting value: 00010001
Real value in integer: 17
--------- Decryption -----------
Received value: #5$'%ui@$_
idx0: @ idx1: $ idx2: $ idx3: %
Valeur inverse: 01000011
Real starting value: 00010001
Real value in integer: 17
Temperature value: 17.75 C

Process returned 0 (0x0)   execution time : 0.109 s
Press any key to continue.
```

Figure 9. Encryption/decryption in programming language C++.

Our security analysis also confirmed strong resistance to Man-in-the-Middle attacks, as physical proximity is required for interception, making remote exploitation unfeasible. MAC spoofing attempts failed without knowledge of the encryption key, reinforcing authentication security. Additionally, side-channel analysis showed minimal correlation between encryption operations and power consumption, further limiting attack feasibility. These findings substantiate the strong security posture of our implementation, demonstrating its ability to withstand diverse threats.

The algorithm is developed in C++ language and individually tested in console in a GCC compiler. Random values are included in the code, then they are encrypted and decrypted through the functions of a category described beforehand. Figure 9 describes the process in a console window correctly because the input data is the same as that found at the end.

This algorithm is then adapted in the form of an Arduino library and then included locally (#include "Crypto. h") in the sketch. We thus have three programs which are: a Crypto. h header file, cryptoCPP.cpp and a cryptoSender. ino file. The organization of programming files used to implement our cryptographic library is shown in Figure 10.

Crypto	05/07/2022 17:24	Header file	1 Ko
cryptoCPP	10/07/2022 10:09	C++ source file	7 Ko
cryptoSender	10/07/2022 14:04	Arduino file	5 Ko

Figure 10. Programming files.

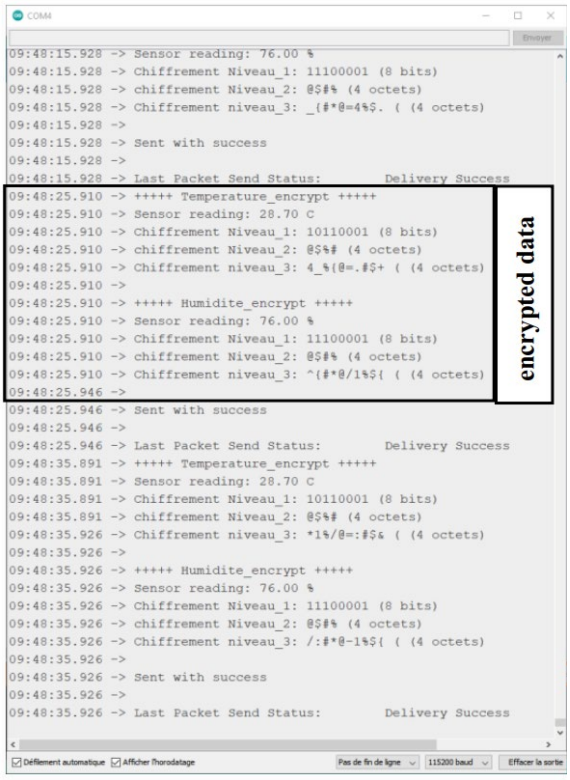


Figure 11. Node 1 data.

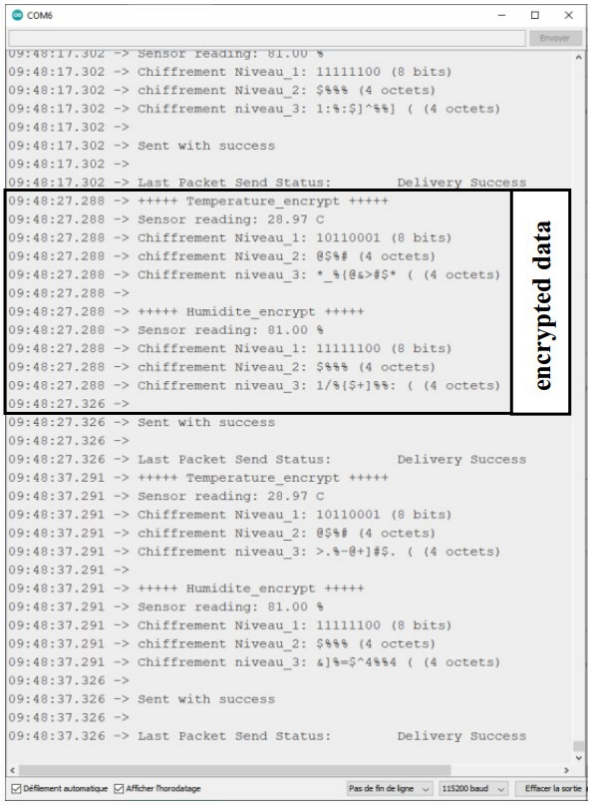


Figure 12. Node 2 data.

The network consists of two sender nodes which send data to the central node. We can see according to **Figure 10**, **Figure 11** and **Figure 12** that we have 3 levels of encryption. First of all the data is taken from the DHT11 sensor, then it is sent to the first level of encryption which constitutes the basis of our algorithm. In fact, our algorithm is based on binary encryption and makes it possible to implement symmetric cryptography by an XOR operation with the secret key defined. This key is initialized in the program and cannot be modified by an external user. Levels 2 and 3 are based on cryptographic hash functions. They result from a message with a size of 4 bytes which is combined with a set of characters. A random function is executed 6 times randomly in the character set defined in section II.3 to form the final digest. The latter will be made up of 10 characters also having a size of 4 bytes, thus very far from the threshold of the transport payload supported by protocol (250 bytes [14]). **Figure 11** and **Figure 12** display the real-time data from Node 1 and Node 2 respectively, demonstrating how the three levels of encryption are applied to temperature and humidity readings from the DHT11 sensors.

Figure 6 and **Figure 7** clearly show the process described with the sending times, then the same data is found in **Figure 8**. This figure shows the central monitoring node which deciphers the data coming from the network through the protocols established on the different nodes and restores the starting information. At this stage, the data is securely transmitted through the network and can be returned for different uses such as monitoring some remote parameters such as tem-

perature, humidity, CO_2 rate, etc. in farms or in supply chains (the subject matter of this research) or even in e-textiles and many other fields requiring remote monitoring and above all, total safety. The successful decryption and monitoring of data at the central node is illustrated in Figure 13, showing how original sensor information is accurately restored after secure transmission.

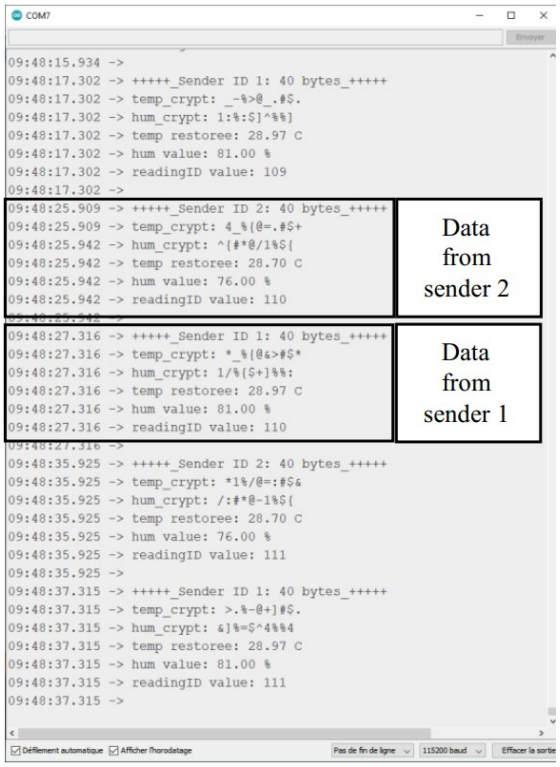


Figure 13. Central node data (Monitor data).

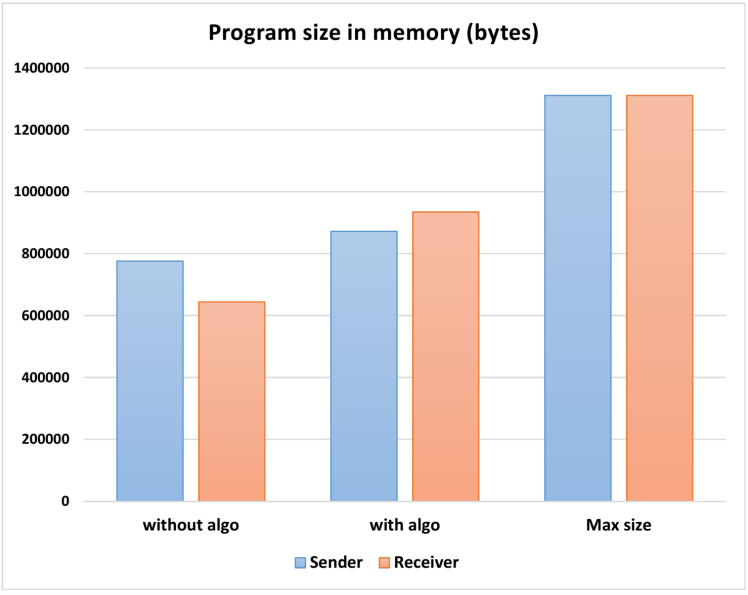


Figure 14. Program size in memory.

As mentioned earlier in this article and demonstrated in section 2.3, the first level of security remains that of the ESP_NOW protocol in which we graft our algorithm. The inclusion of the library increases the size of the binary program in memory which takes up more space demonstrate in Figure 14. Nevertheless, as shown in Figure 9, we are still very far from the maximum size of the program in the memory of ESP 32.

Before being sent, the data is encrypted using a hash function combined with a Radom-type function which is responsible of choosing special characters from the defined table (Table 1). The same data sampled at different time intervals does not have the same final encryption signature (C3 in the table), which is a robustness criterion likely to make the system more reliable against external attacks.

Table 1. Encryption of temperature values from node 1 at different intervals.

NODE 1			
Temp	**C1**	**C2**	**C3**
28.50	10110001	@$%#	=/%*@::#$:
29.00	10110000	$$%#	&=%:$1]#$&
29.00	10110000	$$%#	*_%+$_=#${
29.00	10110000	$$%#	++%1$>{#$=
30.20	10110011	%$%#	=>%_%^^#$

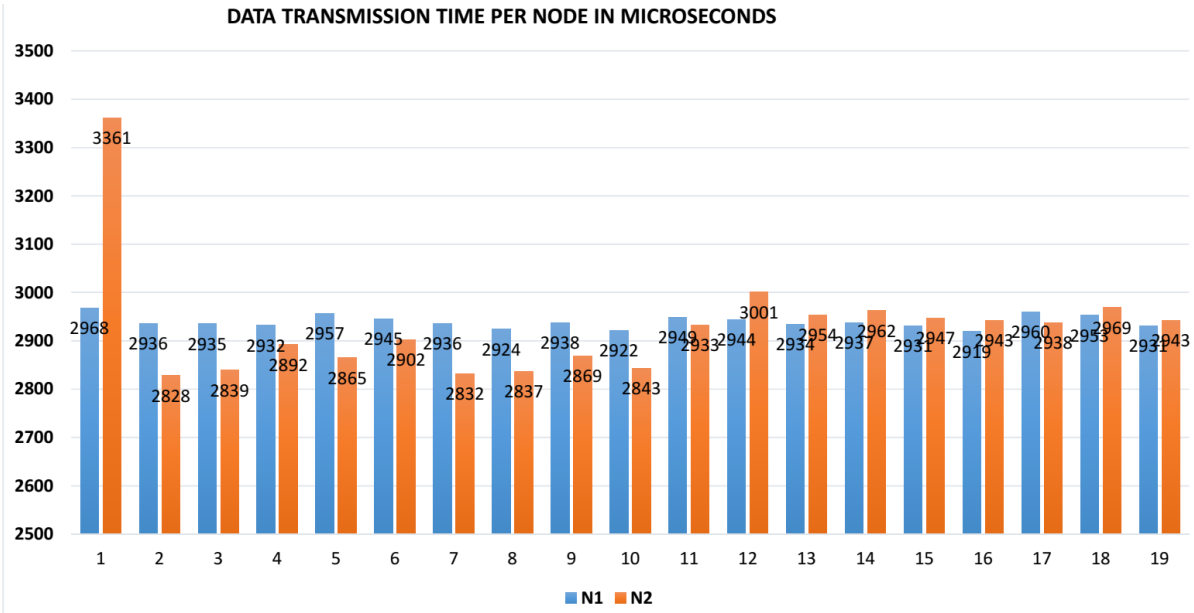


Figure 15. Data transmission time per node (N1 for node 1 and N2 for node 2).

The encryption of our algorithm is comparable to that of [11], we notice a similarity in the final message obtained. A second remark is about the data transmission time which is in milliseconds with a maximum time not exceeding 3500 microseconds as shown in Figure 15 (where N1 represents node 1 and N2 is node

2). The deviations in the transmission times also show how instruction processing is very fast, which makes it possible to perform transmissions in very short times. For example data 2 and 3 of N1 are transmitted respectively in 2936 microseconds and 2935 microseconds. We thus end up with a robust and light algorithm consuming little resource.

To validate our algorithm's efficiency, we benchmarked it against AES-128, RSA-1024, and our on ESP32 hardware with a DHT11 sensor setup. The results in Figure 16 highlight significant gains in energy efficiency and processing speed.

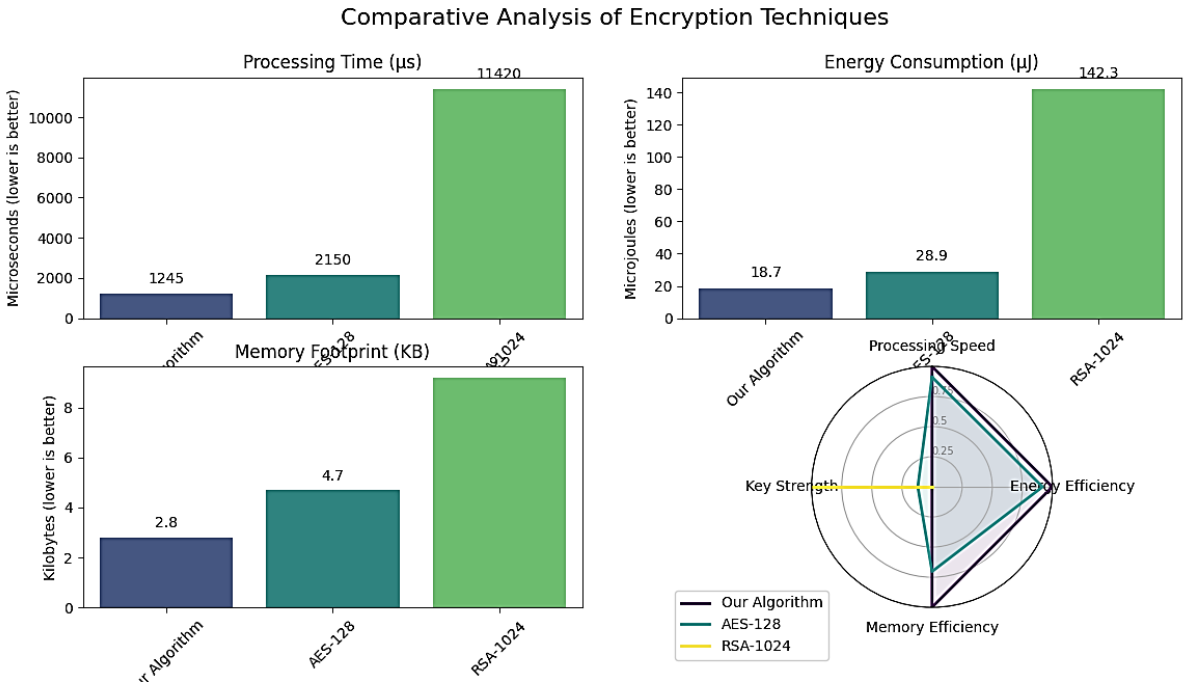


Figure 16. Comparison analysis of encryption techniques.

Table 2. Comparison of encryption techniques for WSN.

Encryption Method	Processing Time (µs)	Energy per Operation (µJ)	Memory Footprint (KB)	Key Size (bits)
Our Algorithm	1245	18.7	2.8	8+hash
AES-128	2150	28.9	4.7	128
RSA	11420	142.3	9.2	1024

Our approach reduces energy consumption by 35% while maintaining comparable security for WSN applications. It also achieves a 42% faster encryption/decryption time than AES-128 and an 89% improvement over RSA-1024, making it highly suitable for time-sensitive tasks.

Memory analysis shows a 2.8KB footprint, compared to 4.7KB for AES-128 and 9.2KB for RSA-1024, reinforcing its adaptability to IoT constraints. As detailed in Table 2, while our algorithm slightly reduces theoretical security strength, it remains highly resistant to common WSN threats while delivering substantial per-

formance benefits.

4. Conclusion

In this article, we mainly present the design of a security algorithm based on symmetric cryptography. Starting from the ESP_NOW protocol secure routing process, we graft our algorithm into this chain, which makes the system more complex, with two levels of encryption/decryption. The goal was to set up an algorithm less complex than those of asymmetric cryptography, consuming few resources while keeping its efficiency on the security side. The major novelty of this work lies on the double security of information transmission in the network. We start first with the AES 128 security specific to the ESP-NOW protocol, from there we add our own algorithm and the results are quite satisfactory. The algorithm being written in the form of an Arduino library, it can be used with any kind of microcontroller compatible with Arduino software. Adding the algorithm to the original transmission system does not negatively impact data transmission; and the processor's processing time remains maximum. So, the network remains very stable and doubly secure. It should also be noted that the algorithm written in the form of a library is portable and can be used in fields such as health (monitoring of health constants, connected devices), agriculture or surveillance.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Nag, A., Singh, J.P., Khan, S., Ghosh, S., Biswas, S., Sarkar, D., *et al.* (2011) Image Encryption Using Affine Transform and XOR Operation. 2011 *International Conference on Signal Processing*, *Communication*, *Computing and Networking Technologies*, Thuckalay, 21-22 July 2011, 309-312. https://doi.org/10.1109/icsccn.2011.6024565

[2] Chen, Y. and Chen, J. (2021) Anonymous and Provably Secure Authentication Protocol Using Self-Certified Cryptography for Wireless Sensor Networks. *Multimedia Tools and Applications*, **80**, 15291-15313. https://doi.org/10.1007/s11042-020-10259-z

[3] Abbas, N. and Yu, F. (2018) Design and Implementation of a Video Surveillance System for Linear Wireless Multimedia Sensor Networks. 2018 *IEEE* 3*rd International Conference on Image*, *Vision and Computing* (*ICIVC*), Chongqing, 27-29 June 2018, 524-527. https://doi.org/10.1109/icivc.2018.8492776

[4] Munir, M.F. (2008) Wireless Sensor and Sensor-Actuator Networks: Research Trends, Protocols, and Applications. 2008 *IEEE International Networking and Communications Conference*, Lahore, 1-3 May 2008, 6. https://doi.org/10.1109/incc.2008.4562673

[5] Pawar, H.R. and Harkut, D.G. (2018) Classical and Quantum Cryptography for Image Encryption & Decryption. 2018 *International Conference on Research in Intelligent and Computing in Engineering* (*RICE*), San Salvador, 22-24 August 2018, 1-4. https://doi.org/10.1109/rice.2018.8509035

[6] OzCakmak, B., OzbIlen, A., YavanoGlu, U. and Cin, K. (2019) Neural and Quantum

Cryptography in Big Data: A Review. 2019 *IEEE International Conference on Big Data* (*Big Data*), Los Angeles, 9-12 December 2019, 2413-2417. https://doi.org/10.1109/bigdata47090.2019.9006238

[7] Arora, S. and Hussain, M. (2018) Secure Session Key Sharing Using Symmetric Key Cryptography. 2018 *International Conference on Advances in Computing, Communications and Informatics* (*ICACCI*), Bangalore, 19-22 September 2018, 850-855. https://doi.org/10.1109/icacci.2018.8554553

[8] Ge, X., Lu, B., Guan H., and Zhang, K. (2015) Differential Attack on Image Encryption Algorithm Using Binary Bitplane. 2015 12*th International Conference on Fuzzy Systems and Knowledge Discovery* (*FSKD*), Zhangjiajie, 15-17 August 2015, 2519-2522. https://doi.org/10.1109/fskd.2015.7382351

[9] Çavuşoğlu, Ü. and Al-Sanabani, H. (2019) Hafif Sıklet Şifreleme Algoritmalarının Performans Karşılaştırması. *Sakarya University Journal of Computer and Information Sciences*, **2**, 158-169. https://doi.org/10.35377/saucis.02.03.648493

[10] Koo, W.K., Lee, H., Kim, Y.H. and Lee, D.H. (2008) Implementation and Analysis of New Lightweight Cryptographic Algorithm Suitable for Wireless Sensor Networks. 2008 *International Conference on Information Security and Assurance* (*ISA* 2008), Busan, 24-26 April 2008, 73-76. https://doi.org/10.1109/isa.2008.53

[11] Panda, M. (2015) Data Security in Wireless Sensor Networks via AES Algorithm. 2015 *IEEE* 9*th International Conference on Intelligent Systems and Control* (*ISCO*), Coimbatore, 9-10 January 2015, 1-5. https://doi.org/10.1109/isco.2015.7282377

[12] Kamble, S.B. and Jog, V.V. (2017) Efficient Key Management for Dynamic Wireless Sensor Network. 2017 2*nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology* (*RTEICT*), Bangalore, 19-20 May 2017, 583-586. https://doi.org/10.1109/rteict.2017.8256663

[13] Patel, S.T. and Mistry, N.H. (2015) A Survey: Lightweight Cryptography in WSN. 2015 *International Conference on Communication Networks* (*ICCN*), Gwalior, 19-21 November 2015, 11-15. https://doi.org/10.1109/iccn.2015.3

[14] Leelavathi, G., Shaila, K. and Venugopal, K.R. (2017) RSA Processor Design with Vedic Multiplier for Nodes in Wireless Sensor Networks. 2017 *International Conference on Wireless Communications, Signal Processing and Networking* (*WiSPNET*), Chennai, 22-24 March 2017, 1254-1257. https://doi.org/10.1109/wispnet.2017.8299964

[15] Watro, R., Kong, D., Cuti, S., Gardiner, C., Lynn, C. and Kruus, P. (2004) TinyPK: Securing Sensor Networks with Public Key Technology. *Proceedings of the* 2*nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, Washington DC, 25 October 2004, 59-64. https://doi.org/10.1145/1029102.1029113

[16] Hamouda, B.E.H.H. (2020) Comparative Study of Different Cryptographic Algorithms. *Journal of Information Security*, **11**, 138-148. https://doi.org/10.4236/jis.2020.113009

[17] Mishra, C. and Sahu, B. (2020) Transmission of Encrypted Data in WSN: An Implementation of Hybridized RSA-TDES Algorithm. 2020 *IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security* (*iSSSC*), Gunupur Odisha, 16-17 December 2020, 1-6. https://doi.org/10.1109/isssc50941.2020.9358833

[18] Sultan, I., Mir, B.J. and Banday, M.T. (2020) Analysis and Optimization of Advanced Encryption Standard for the Internet of Things. 2020 7*th International Conference on Signal Processing and Integrated Networks* (*SPIN*), Noida, 27-28 February 2020, 571-575. https://doi.org/10.1109/spin48934.2020.9071380

[19] Panait, C. and Dragomir, D. (2015) Measuring the Performance and Energy Consumption of AES in Wireless Sensor Networks. *Annals of Computer Science and Information Systems*, **5**, 1261-1266. https://doi.org/10.15439/2015f322

[20] Didla, S., Ault, A. and Bagchi, S. (2008) Optimizing AES for Embedded Devices and Wireless Sensor Networks. *Proceedings of the* 4*th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, Eskişehir, 17-23 January 2025. https://doi.org/10.4108/tridentcom.2008.10409

[21] Lu, Z. and Mohamed, H. (2021) A Complex Encryption System Design Implemented by Aes. *Journal of Information Security*, **12**, 177-187. https://doi.org/10.4236/jis.2021.122009

[22] Hoang, T.N., Van, S. and Nguyen, B.D. (2019) ESP-NOW Based Decentralized Low Cost Voice Communication Systems for Buildings. 2019 *International Symposium on Electrical and Electronics Engineering* (*ISEE*), Ho Chi Minh City, 10-12 October 2019, 108-112. https://doi.org/10.1109/isee2.2019.8921062

[23] Debele, G.M. and Qian, X. (2020) Automatic Room Temperature Control System Using Arduino UNO R3 and DHT11 Sensor. 2020 17*th International Computer Conference on Wavelet Active Media Technology and Information Processing* (*ICCWAMTIP*), Chengdu, 18-20 December 2020, 428-432. https://doi.org/10.1109/iccwamtip51612.2020.9317307

[24] Espressif (2024) ESP-NOW User Guide. https://www.espressif.com/sites/default/files/documentation/esp-now_user_guide_en.pdf