Jurnal Ilmu Komputer Agri-Informatika

Tersedia secara daring di:
https://jurnal.ipb.ac.id/index.php/jika

Volume 13 Nomor 1 halaman 27–35
eISSN: 2654-9735, pISSN: 2089-6026

Stream Cipher-Based Mutual Authentication for Low-Power LoRa P2P Devices

YOSO ADI SETYOKO[1]*, IMAS SUKAESIH SITANGGANG[1], SHELVIE NIDYA NEYMAN[1], SRI WAHJUNI[1]

Abstract

The Internet of Things (IoT) has been widely implemented in various industrial sectors in Indonesia. Examples of these implementations include devices for monitoring home electricity usage, automation tools in industry, and early warning systems for forest fires. IoT implementations often use resource-constrained devices. This poses a challenge for developers seeking lightweight algorithms to secure IoT data. Therefore, lightweight stream cipher encryption algorithms were selected, including Snow-3G, Snow-V, Rabbit, and ZUC. This study proposes a stream-cipher-based mutual authentication scheme employing a random Initialization Vector (IV) in the mutual authentication process for LoRa Peer-to-Peer (P2P). BAN Logic is used in this study as a tool for mutual authentication validation. This study uses low-power ATmega328P hardware for the protocol. The results of this study demonstrate an improvement in LoRa P2P network security through mutual authentication using a random IV. This study also successfully benchmarked the performance of four stream cipher algorithms. The encryption and decryption process of 256 bytes of data takes 53.89 milliseconds (ms) for Snow-3G, 11.36 ms for Snow-V, 7.78 ms for Rabbit, and 15.04 ms for ZUC.

*Keywords**: ATmega328P, BAN Logic, low-power devices, mutual authentication, random initialization vectors, stream cipher encryption*

Abstrak

Internet of Things (IoT) sudah banyak diimplementasikan di berbagai sektor industri di Indonesia. Implementasi tersebut, contohnya, antara lain alat untuk *monitoring* penggunaan daya listrik di rumah, alat otomasi di industri, dan peringatan dini kebakaran di hutan. Implementasi IoT banyak menggunakan *device* dengan kemampuan komputasi yang rendah. Hal tersebut menjadi tantangan bagi *developer* dalam memilih algoritma yang ringan untuk mengamankan data IoT. Oleh karena itu, empat algoritma stream cipher, yaitu Snow-3G, Snow-V, Rabbit, dan ZUC, dipilih karena karakteristiknya yang sesuai untuk perangkat IoT dengan keterbatasan sumber daya komputasi. Penelitian ini mengusulkan penggunaan algoritma stream cipher dengan random Initialization Vector (IV) dalam proses mutual authentication pada jaringan LoRa Peer-to-Peer (P2P). BAN Logic digunakan pada penelitian ini sebagai alat untuk validasi *mutual authentication*. Penelitian ini menggunakan *hardware low power* ATmega328P yang digunakan pada protokol. Hasil penelitian ini adalah pengembangan keamanan jaringan LoRa P2P dengan penambahan *mutual authentication* yang memanfaatkan *random* IV. Penelitian ini juga berhasil melakukan *benchmarking* performansi empat algoritma *stream cipher*. Proses enkripsi dan dekripsi 256 *byte* data membutuhkan waktu 53.89 milidetik (ms) untuk Snow-3G, 11.36 ms untuk Snow-V, 7.78 ms untuk Rabbit, dan 15.04 ms untuk ZUC.

Kata Kunci**:** ATmega328P, BAN Logic, enkripsi *stream cipher*, *mutual authentication*, *random initialization vector*

INTRODUCTION

The Internet of Things (IoT) has emerged as a key enabling technology in the Industry 4.0 paradigm, facilitating the integration of physical devices with digital systems. In Indonesia, IoT has been widely adopted across various industrial sectors to improve operational efficiency and data-driven decision-making. The implementation of IoT has been shown to enhance Enterprise Resource Planning (ERP) systems by enabling real-time data acquisition and monitoring (Nugroho *et al.* 2020). In addition, industrial facilities benefit from IoT-based

[1] School of Data Science, Mathematics, and Informatics, IPB University, Bogor, Indonesia.
* Penulis Korespondensi: Surel: yosoadisetyoko@apps.ipb.ac.id.

solutions for monitoring electrical energy consumption (Mubarok and Ardiansyah 2020), while the Indonesian government has utilized IoT technologies for environmental monitoring, particularly in the prevention of forest fires. Despite these advancements, challenges remain in integrating network technologies capable of connecting low-power devices over wide and rural areas (Kartikasari *et al.* 2020). Long Range (LoRa), as part of the Low Power Wide Area Network (LPWAN) technology, offers a promising solution due to its capability to support long-range communication with low energy consumption (Turcinovic *et al.* 2020). Low-power end devices (EDs) can communicate within LoRa networks over distances of up to approximately 15 kilometers, making it suitable for large-scale and remote IoT deployments (Villarim *et al.* 2019).

However, the implementation of IoT technology is inherently associated with significant information security challenges. In many cases, IoT deployments, particularly in rural environments, involve outdoor installations that are exposed to various security risks. The use of outdoor IoT end devices raises critical concerns regarding data protection and secure communication. Due to the limited computational capabilities of most IoT end devices (Alsharif *et al.* 2024), implementing robust security mechanisms remains a significant challenge. Consequently, IoT systems must be designed by carefully balancing security requirements against the resource constraints of end devices.

This study makes two main contributions. First, it implements four encryption techniques to support mutual authentication among LoRa devices operating in a peer-to-peer (P2P) architecture. Upon successful authentication, the devices are capable of transmitting encrypted data securely. The proposed authentication protocol is formally verified using BAN Logic, which ensures the correctness of mutual authentication and validates the freshness of protocol exchanges, thereby mitigating replay attack threats (Liu *et al.* 2018). Second, this study develops a prototype for data encryption in low-power IoT devices. The experimental evaluation provides insights into the computational performance of various symmetric encryption algorithms implemented on the prototype.

This study evaluates the execution time of four stream cipher encryption algorithms, namely Snow 3G, Snow-V, Rabbit, and ZUC. The selection of Snow 3G and ZUC is based on their standardization by the 3rd Generation Partnership Project (3GPP) for securing data communication in Long Term Evolution (LTE) networks (Nilofer dan Qaddour 2018; Sharaf *et al.* 2020). The Snow-V algorithm is included due to its demonstrated high-speed performance in previous studies, addressing the growing demand for ultra-fast encryption in modern communication systems (Ekdahl *et al.* 2019). Additionally, the Rabbit algorithm is selected as it is recognized for its efficiency and high-speed encryption capabilities (Boesgaard *et al.* 2004) and has been applied in IoT systems for encryption keys (Han and Wang 2018; Benny Gandara and Alaydrus 2019).

The experimental setup utilizes a low-power processing platform. Specifically, the ATmega328P microcontroller is employed to implement all selected stream cipher algorithms. This microcontroller is chosen due to its widespread use and relevance in embedded systems and IoT research prototypes (Firdous *et al.* 2020). All evaluated algorithms belong to the stream cipher category and are implemented following three primary stages: initialization, keystream generation, and data encryption. The prototype is designed to measure the execution time of both encryption and decryption processed on the microcontroller.

METHODS

The research methodology adopted in this study is a prototype-based approach, which enables direct validation of system performance through the integration of hardware and software components. The proposed system is implemented and evaluated using a prototype that facilitates the testing of mutual authentication mechanisms and encrypted data transmission. The LoRa peer-to-peer (P2P) network prototype is deployed over a distance of less than 1 kilometer under line-of-sight conditions, ensuring direct communication without

physical obstructions. In addition, the BAN Logic method is employed to formally validate the proposed authentication protocol, particularly in verifying mutual authentication and ensuring the freshness property of message exchanges.

Scenario for Stream Cipher Encryption Process

This section presents the experimental results of execution time for four stream cipher algorithms implemented on the ATmega328P microcontroller. The benchmarking process evaluates two main components: algorithm initialization and the encryption process. Considering the resource constraints of IoT end devices, relatively small data sizes are utilized in the experiments. The evaluation involves 256 bytes of data for both encryption and decryption processed, which aligns with the maximum payload capacity specified in LoRa communication standards (Casals *et al.* 2017; Laveyne *et al.* 2018; Triantafyllou *et al.* 2022). A simplified LoRa architecture is implemented in this study, eliminating the need for a gateway by employing a direct P2P communication model. The four selected stream cipher algorithms—Snow 3G, Snow-V, Rabbit, and ZUC—are evaluated based on their respective key sizes and keystream generation characteristics, in accordance with their specifications (3GPP 2010).

In the implementation, Snow 3G utilizes a 128-bit key and generates a 32-bit keystream, while Snow-V utilizes a 256-bit key and produces a 128-bit keystream. The Rabbit algorithm uses a 128-bit key and performs encryption directly without an explicit keystream generation phase. The ZUC algorithm adopts a 128-bit key and generates a 128-bit keystream. Experimental results indicate that the keystream generation process of Snow-V is faster compared to ZUC (Wei *et al.* 2021). The prototype processed 256 bytes (2048 bits) of data, requiring multiple iterations of keystream generation to complete the encryption process, as keystream generation is a fundamental component of stream cipher operations.

Performance evaluation results show that all four algorithms exhibit high-speed XOR operations, requiring approximately 4 to 8 microseconds (μs) to process 256 bytes of data. However, the keystream generation phase contributes significantly to the overall execution time. The measured results include both encryption and decryption processed, encompassing key initialization and keystream generation stages.

Figure 1 illustrates the flowchart of the encryption process implemented on the microcontroller. The process begins with inputs consisting of the encryption key, Initialization Vector (IV), and plaintext. Subsequently, the stream cipher generates a keystream corresponding to the size of the plaintext, which is then combined using XOR operations to produce the ciphertext. This mechanism implies that the duration of stream cipher encryption is strongly influenced by the keystream generation process.

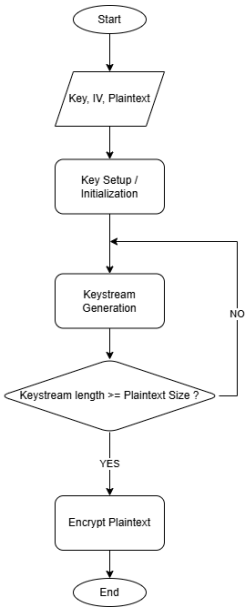

Figure 1 Stream cipher encryption process for testing scenario

Mutual Authentication Process

Figure 2 illustrates the procedures involved in the mutual authentication process. The LoRa receiver and transmitter execute authentication procedures before data transmission. Device IDs are very important to the mutual authentication process. This study uses stream cipher encryption algorithms for the mutual authentication process.

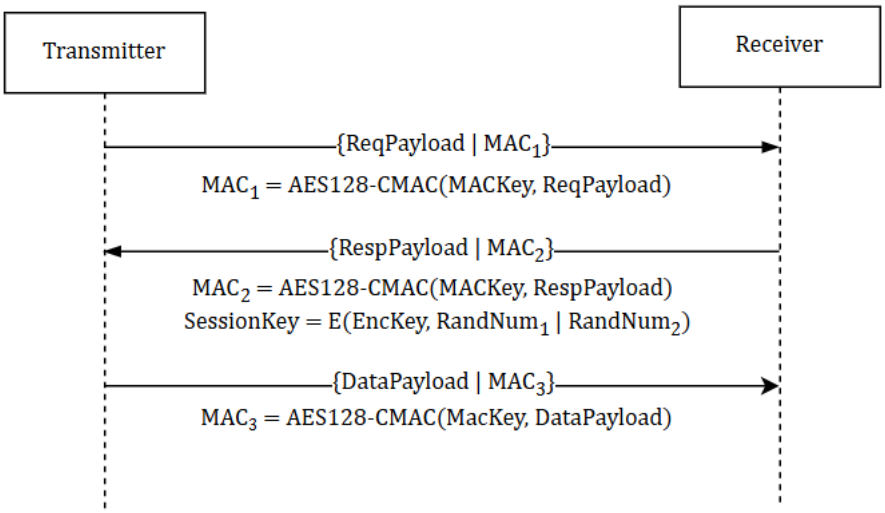

Figure 2 Mutual authentication process

1. LoRa transmitter has 8 bytes $TransID$ for the transmitter ID, and then the transmitter sends a communication request to the LoRa receiver. First, the transmitter generates a 16-byte random number, denoted as $RandNum_1$ (Equation 1). The transmitter then encrypts the 16-byte random value *RandNum* using *EncKey* to generate a 16-byte *TransNonce*, as defined in Equation (2). $SeqNum_1$ is 4 bytes sequence number, and Ts_1 is 4 bytes transmitter timestamp. This scheme uses 8 bytes of random $Initialization\ Vector\ (IV)$.Concatenation of $TransID$, $TransNonce$, $SeqNum_1$, Ts_1 (Equation 3) will be used to create 8 bytes MAC_1. MAC_1 has been calculated by $AES128-CMAC$ algorithm (Equation 4).

$$IV_1 = RandIV \quad (1)$$

$$TransNonce = E(EncKey, RandNum_1, IV_1) \quad (2)$$

$$ReqPayload = TransID \mid TransNonce \mid SeqNum_1 \mid Ts_1 \mid IV_1 \quad (3)$$

$$MAC_1 = AES128-CMAC(MACKey, ReqPayload) \quad (4)$$

$$ReqMessage = ReqPayload \mid MAC_1$$

2. The LoRa receiver reads *ReqPayload* and *MAC* from the transmitter. It then verifies whether *MAC* corresponds to *ReqPayload*. The receiver generates 8 bytes $MACVerification_1$ to check the validity of MAC_1 with $AES128-CMAC(ReqPayload)$ calculation. If $MACVerification_1$ value is equal to MAC_1 so the receiver accepts this communication request. The receiver prepares to create a response message and session key. And then the receiver sends a response message with MAC_2. The receiver not only checks MAC_1 but also check $SeqNum_1$ and Ts_1. $SeqNum_1$ must be $LastSeqNum + 1$ for a valid request. And then timestamps $CurrentTs - Ts_1 \leq 10\ seconds$ for a valid request. Sequence number and timestamps validation applies to the transmitter when reading the response message.

$$MACVerification_1 = AES128-CMAC(MACKey, ReqPayload)$$

$$IV_2 = RandIV$$

$$RcvNonce = E(EncKey, RandNum_2)$$

$$RespPayload = RcvID \mid RcvNonce \mid SeqNum_2 \mid Ts_2 \mid IV_2$$

$$MAC_2 = AES128-CMAC(MACKey, RespPayload)$$

$$RespMessage = RespPayload \mid MAC_2$$
$$ClientNonce = D(EncKey, RcvNonce)$$
$$SessionKey = E(EncKey, RandNum_1 \mid RandNum_2)$$

3. LoRa transmitter receives and verifies $RespPayload$ from the receiver. LoRa transmitter calculate $MACVerification_2$ (Equation 5) to verify $RespPayload$. If $MACVerification_2$ value equals with MAC_2 so $RespPayload$ is valid. If $RespPayload$ is valid so transmitter prepare to create $SessionKey$ (Equation 6). The data message can be delivered from transmitter to receiver securely. Data message is encrypted by $SessionKey$, and data message is equipped with $SeqNum_3$ and Ts_3 for freshness (Equation 7).

$$MACVerification_2 = AES128 - CMAC(MACKey, RespPayload) \quad (5)$$
$$RandNum_2 = D(EncKey, RcvNonce)$$
$$SessionKey = E(EncKey, RandNum_1 \mid RandNum_2) \quad (6)$$
$$DataPayload = TransID \mid E(SessionKey, Data) \mid SeqNum_3 \mid Ts_3 \mid IV_3 \quad (7)$$
$$MAC_3 = AES128 - CMAC(MacKey, DataPayload)$$
$$DataMessage = DataPayload \mid MAC_3$$

The LoRa transmitter and receiver provide the same session keys. LoRa transmitters and receivers encrypt payloads or information using session keys before the message transmission process. The mutual authentication process between the transmitter and receiver is successful. The encryption method generates $TransNonce$, $RcvNonce$, and $SessionKey$ using stream cipher encryption techniques. This study uses sequence numbers and timestamps to maintain the currency of this approach. Upon successful mutual authentication, the $SessionKey$ can be utilized to encrypt data flow from the transmitter to the receiver.

BAN Logic Analysis

BAN Logic is a formal method used to analyze and verify the security properties of network and communication protocols (Liu *et al.* 2018). It employs a set of logical rules and notations to evaluate whether a protocol satisfies specific security requirements. In this study, two primary rules are utilized, namely the Message Meaning Rule (MMR), which is calculated using Equation 8, and the Nonce Verification Rule (NVR), which is calculated using Equation 9. These rules are applied to assess the correctness of the authentication process and to ensure that both communicating parties achieve mutual authentication. BAN Logic rules use BAN Logic notation. The notation $P| \equiv Q$ means P believes Q. Notation $P| \equiv \#(X)$ means P believes X message is fresh. The notation $P \overset{K}{\leftrightarrow} Q$ means P and Q agree to use key K. Notation $Q \lhd \{X\}_K$ means Q understands X, which is encrypted with K. Notation $Q|{\sim}X$ means Q said X.

$$Message\ Meaning\ Rule\ (MMR) = \frac{P| \equiv P \overset{K}{\leftrightarrow} Q, Q \lhd \{X\}_K}{P| \equiv Q| {\sim} X} \quad (8)$$

If P believes P and Q use key K and Q sees X, then it can be concluded that P believes Q said X. P and Q can only communicate encrypted messages using key K. If both parties comply with the MMR rules, then P and Q can authenticate using key K.

$$Nonce\ Verification\ Rule\ (NVR) = \frac{P| \equiv \#(X), P| \equiv Q| {\sim} X}{P| \equiv Q| \equiv X} \quad (9)$$

NVR means that if P believes X is fresh and if P believes Q said X, then P believes Q believes X. In other words, if P believes X is fresh and if the protocol complies with MMR, then P believes Q believes the X message. NVR helps ensure that the message sent from P to Q is fresh. Random Initial Vector (IV) in this study is used so that the message sent from the transmitter to the receiver is fresh. There are also other variables, such as timestamps are also used in this protocol to ensure the message sent is fresh. So, it can be concluded that $ReqPayload$ and $RespPayload$ in this study are fresh.

The process of verifying protocol security using BAN Logic has steps including determining assumptions, determining goals, and making conclusions through BAN Logic rules (Liu *et al.* 2018). The BAN Logic initial assumptions used in this study are as follows:

1. $Rcv| \equiv \#(ReqPayload)$The receiver believes $RandNum_1$ is fresh. Freshness means that the $ReqPayload$ is valid only once during the authentication process. This ensures that the authentication process cannot be duplicated by other parties. The timestamp variable Ts and random IV helps ensure that the authentication process is fresh and free of redundancies.
2. $Trans| \equiv \#(RespPayload)$, the transmitter believes $RandNum_2$ is fresh.
3. $Rcv| \equiv Rcv \overset{MACKey}{\longleftrightarrow} Trans$, the receiver believes $MACKey$ is the encryption key which is used by the transmitter and receiver.
4. $Trans| \equiv Trans \overset{MACKey}{\longleftrightarrow} Rcv$, the transmitter believes $MACKey$ is the encryption key between the transmitter and receiver.

Furthermore, the objectives of the proposed protocol scheme in this study are defined using BAN Logic notation where $Rcv| \equiv Trans| \equiv ReqPayload$ and $Trans| \equiv Rcv| \equiv RespPayload$. The meaning of the first goal above is receiver believes the transmitter believes $ReqPayload$. And then the second goal means the transmitter believes the receiver believes $RespPayload$. We will start with the message meaning rule in BAN logic as follows. First, the message comes from the transmitter to the receiver with an idealized format as follows $\{ReqPayload\}_{MacKey}$. So it means $Rcv \lhd \{ReqPayload\}_{MacKey}$ with Message Meaning Rules (MMR). Equation 10 is the equation to calculate MMR.

$$MMR = \frac{Rcv| \equiv Rcv \overset{MACKey}{\longleftrightarrow} Trans, Rcv \lhd \{ReqPayload\}_{MACKey}}{Rcv| \equiv Trans| \sim ReqPayload} \quad (10)$$

Then we get $Rcv| \equiv Trans| \sim ReqPayload$. The calculation of NVR is derived using Equation 11.

$$NVR = \frac{Rcv| \equiv \#(ReqPayload), Rcv| \equiv Trans| \sim ReqPayload}{Rcv| \equiv Trans| \equiv ReqPayload} \quad (11)$$

Finally, we get the first goal, which is $Rcv| \equiv Trans| \equiv ReqPayload$. And then we will achieve the second goal with the message meaning rule and nonce verification rule as follows, as in Equations 12 and 13.

$$MMR = \frac{Trans| \equiv Trans \overset{MACKey}{\longleftrightarrow} Rcv, Trans \lhd \{RespPayload\}_{MACKey}}{Trans| \equiv Rcv| \sim RespPayload} \quad (12)$$

$$NVR = \frac{Trans| \equiv \#(RespPayload), Trans| \equiv Rcv| \sim RespPayload}{Trans| \equiv Rcv| \equiv RespPayload} \quad (13)$$

We achieved the second goal $Trans| \equiv Rcv| \equiv RespPayload$. The BAN Logic method helps verify that both the transmitter and receiver have successfully verified authentication using the private key. Therefore, this research scheme successfully performs mutual authentication.

RESULTS AND DISCUSSIONS

Benchmarking Results of Stream Cipher Process

Table 1 shows that the Rabbit encryption algorithm achieves the fastest performance in keystream generation. The results suggest that stream cipher algorithms are well-suited for implementation in low-power devices, particularly for non-real-time communication scenarios. Furthermore, Figure 3 presents the total time required to encrypt 256 bytes (or 256 characters) on the ATmega328P microcontroller. Among the evaluated methods, Rabbit demonstrates the best overall performance, while Snow-3G exhibits the slowest execution time compared to the other algorithms.

Table 1 Algorithm test results in ATmega328P

Algorithms	Average Execution Time		*Key Size (bits)*
	Key Setup (ms)	*Encryption / Decryption 2048 bits or 256 bytes (ms)*	
Snow-3G	26.5	53.89	128
Snow-V	11.68	11.36	256
Rabbit	1.5	7.78	128
ZUC	7.23	15.04	128

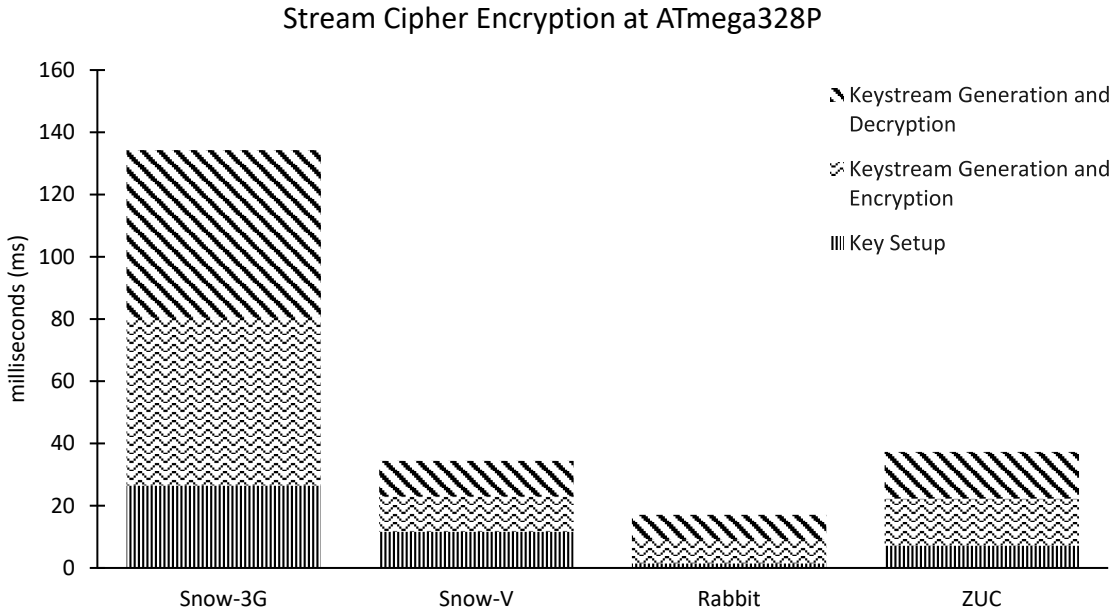

Figure 3 Total time for encryption process

Protocol Test with Prototype

The protocol prototype is developed based on specific hardware architectural requirements. This prototype demonstrates that stream cipher algorithms can be effectively implemented to support mutual authentication and ensure data security. The system is equipped with a LoRa SX1278 module and a Real-Time Clock (RTC) DS3231 module. Table 2 presents the results of the mutual authentication process and data transmission between the transmitter and receiver. Meanwhile, Table 3 illustrates that identical plaintext inputs produce different ciphertext outputs due to the randomization of the Initialization Vector (IV).

Table 2 Mutual authentication and data transmission results with prototype

Messages	Authentication Steps		*Size (bytes)*
	Source→Destination	*Messages(hex)*	
ReqMessage	Trans→Rcv	0 A C B 9 F 6 D 7D E1 E9 E2 5F 86 82 12 CC 6B C0 23 8E C6 93 79 0 0 0 1 0 0 0 A 1 0 0 0 0 0 0 0 F D5 B3 7E F2 73 8F A8	48
RespMessage	Rcv→Trans	1 B D C A 1 7 E 7E E2 EA E1 5C 85 81 11 CF 68 C3 20 8D C5 90 7A 0 0 0 2 0 0 0 1E 1 0 0 0 0 0 0 0 C2 BB 96 89 BE 1A 26 C1	48
DataTransmission	Trans→Rcv	0 A C B 9 F 6 D 65 EF FB EC 1E 86 87 1A 8D 79 A1 42 EF A7 F2 18 0 0 0 A 0 0 0 B 1 0 0 0 0 0 0 0 26 35 6C CD DF F2 EB 6	48

Table 3 Static and random IV keystream result in Rabbit

Keystream with static Initial Vector (IV)	Keystream with dynamic Initial Vector (IV)
7D E1 E9 E2 5F 86 82 12 CC 6B C0 23 8E C6 93 79	C7 8 D8 56 7F C7 66 2F AE 22 A9 B3 4F FC BF C2
7D E1 E9 E2 5F 86 82 12 CC 6B C0 23 8E C6 93 79	4B 71 3B B5 74 5C 8D 9E 6A B8 15 43 54 24 E2 F9
7D E1 E9 E2 5F 86 82 12 CC 6B C0 23 8E C6 93 79	4D F6 67 E1 FA ED 4B 6C 84 94 8F E8 93 E6 E2 1B
7D E1 E9 E2 5F 86 82 12 CC 6B C0 23 8E C6 93 79	A4 3C E7 7B 9D 3B 63 3F 18 1C FA 8F 72 F2 76 2E
7D E1 E9 E2 5F 86 82 12 CC 6B C0 23 8E C6 93 79	D4 22 DA E1 F3 A5 24 7D 56 D9 4 8D 7 A3 38 D4

CONCLUSION

The LoRa security protocol proposed in this study has been successfully designed and evaluated using four stream cipher algorithms: Snow-3G, Snow-V, Rabbit, and ZUC. This research developed a LoRa P2P communication model enhanced with mutual authentication capabilities and validated the proposed scheme through a hardware-based prototype. The BAN Logic method was effectively applied to verify the correctness of the mutual authentication process within the protocol. In addition, this study ensured the use of consistently randomized Initialization Vectors (IVs) in stream cipher encryption, strengthening data security. The experimental results demonstrate that all evaluated stream cipher algorithms can be efficiently implemented on resource-constrained devices such as the ATmega328P microcontroller. Among the tested algorithms, Rabbit achieved the best performance, requiring only 7.78 milliseconds to encrypt 256 characters of data. The benchmarking results confirm that Rabbit is the fastest algorithm for message encryption in this scenario. These findings represent the initial stage of our research on LoRa security protocols. Future work will focus on improving the prototype by developing a more advanced LoRa security architecture. Additionally, this study has not yet evaluated protocol performance in environments with significant physical obstructions, which remains an important area for further investigation. This research also opens opportunities for optimizing energy efficiency in LoRa security and highlights the challenge of integrating LoRa security mechanisms with other IoT protocols.

REFERENCES

3GPP. 2010. ETSI / SAGE Date : 18 th June 2010 Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3 Document 3 : Implementors ’ Test Data. *Test*. :1–20.

Alsharif MH, Kelechi AH, Jahid A, Kannadasan R, Singla MK, Gupta J, Geem ZW. 2024. A comprehensive survey of energy-efficient computing to enable sustainable massive IoT networks. *Alexandria Eng. J.* 91:12–29. https://doi.org/10.1016/j.aej.2024.01.067.

Benny Gandara R, Alaydrus M. 2019. Analysis of the IEEE 802.15.4 protocol with rabbit encryption algorithm for industrial applications in oil and gas sector. *2019 16th Int. Conf. Qual. Res. QIR 2019 - Int. Symp. Electr. Comput. Eng.*:1–5. https://doi.org/10.1109/QIR.2019.8898287.

Boesgaard M, Pedersen T, Vesterager M, Zenner E. 2004. *The Rabbit Stream Cipher-Design and Security Analysis*.

Casals L, Mir B, Vidal R, Gomez C. 2017. Modeling the energy performance of LoRaWAN. *Sensors (Switzerland)*. 17(10). https://doi.org/10.3390/s17102364.

Ekdahl P, Johansson T, Maximov A, Yang J. 2019. A new SNOW stream cipher called SNOW-V. *IACR Trans. Symmetric Cryptol.* 2019(3):1–42. https://doi.org/10.13154/tosc.v2019.i3.1-42.

Firdous A, Indu, Niranjan V. 2020. Smart Density Based Traffic Light System. *ICRITO 2020 - IEEE 8th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir.*:497–500. https://doi.org/10.1109/ICRITO48877.2020.9197940.

Han J, Wang J. 2018. An enhanced key management scheme for lorawan. *Cryptography*. 2(4):1–12. https://doi.org/10.3390/cryptography2040034.

Kartikasari V, Afroni MJ, Basuki BM. 2020. Model Sistem Monitoring Kebakaran Hutan Berbasis Lora Dengan Menggunakan Arduino. *Sci. Electro*. :39–43.

Laveyne JI, Van Eetvelde G, Vandevelde L. 2018. Application of LoRaWAN for Smart Metering: An Experimental Verification. *Int. J. Contemp. ENERGY*. 4(1):61–67.

Liu B, Yang B, Su X. 2018. An improved two-way security authentication protocol for RFID system. *Inf.* 9(4). https://doi.org/10.3390/info9040086.

Mubarok H, Ardiansyah A. 2020. Prototype Design of IoT (Internet of Things)-based Load Monitoring System. *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2020*. :377–382. https://doi.org/10.1109/ISRITI51436.2020.9315454.

Nilofer F, Qaddour J. 2018. Comparative Study of Vulnerabilities in LTE Cryptographic Algorithm. *Int. J. Comput. Appl.* 180(25):19–25. https://doi.org/10.5120/ijca2018916587.

Nugroho A, Rizaludin D, Soebandhi S, Junaedi L, Winardi S, Al-Azam MN. 2020. Automatic Sign of Commencement of Work from Enterprise Resource Planning. *Proceeding - ICoSTA 2020 Int. Conf. Smart Technol. Appl. Empower. Ind. IoT by Implement. Green Technol. Sustain. Dev.* :0–5. https://doi.org/10.1109/ICoSTA48221.2020.1570590248.

Sharaf MA, Abdelbary E, Mostafa H, Hussein A, Nassar AM. 2020. Efficient ASIC Implementation of a NB-IoT Security Co-processor. *Midwest Symp. Circuits Syst.* :695–698. https://doi.org/10.1109/MWSCAS48704.2020.9184519.

Sharif SO, Mansoor SP. 2010. Performance analysis of stream and block cipher algorithms. *ICACTE 2010 - 2010 3rd Int. Conf. Adv. Comput. Theory Eng. Proc.* 1:522–525. https://doi.org/10.1109/ICACTE.2010.5578961.

Triantafyllou A, Zorbas D, Sarigiannidis P. 2022. Time-slotted LoRa MAC with variable payload support. *Comput. Commun.* 193:146–154. https://doi.org/10.1016/j.comcom.2022.06.043.

Turcinovic F, Vukovic J, Bozo S, Sisul G. 2020. Analysis of LoRa Parameters in Real-World Communication. *Proc. Elmar - Int. Symp. Electron. Mar.* 2020. :87–90. https://doi.org/10.1109/ELMAR49956.2020.9219028.

Villarim MR, De Luna JVH, De Farias Medeiros D, Pereira RIS, De Souza CP. 2019. LoRa performance assessment in dense urban and forest areas for environmental monitoring. *INSCIT 2019 - 4th Int. Symp. Instrum. Syst. Circuits Transducers*. :1–5. https://doi.org/10.1109/INSCIT.2019.8868567.

Wei M, Yang G, Kong F. 2021. Software implementation and comparison of ZUC-256, SNOW-V, and AES-256 on RISC-V platform. *2021 IEEE Int. Conf. Inf. Commun. Softw. Eng. ICICSE 2021*. :56–60. https://doi.org/10.1109/ICICSE52190.2021.9404134.