

# About one lightweight encryption algorithm ensuring the security of data transmission and communication between internet of things devices

Sabyrzhan Atanov<sup>1</sup>, Yerzhan Seitkulov<sup>1</sup>, Khuralay Moldamurat<sup>1</sup>, Banu Yergaliyeva<sup>1</sup>,  
Abzal Kyzyrkanov<sup>2</sup>, Zhexen Seitbattalov<sup>1</sup>

<sup>1</sup>Faculty of Information Technology, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

<sup>2</sup>Department of Computer Engineering, Astana IT University, Astana, Kazakhstan

## Article Info

### Article history:

Received Mar 14, 2024

Revised Jul 11, 2024

Accepted Jul 17, 2024

### Keywords:

Embedded systems

Fuzzy logic

Internet of things network

Mersenne Twister

Mobile security

Twine encryption algorithm

## ABSTRACT

In this paper, a new encryption algorithm Twine-Mersenne was developed based on the Twine algorithm with the addition of a random number generator for the dynamic generation of S-boxes. Dynamic generation of random numbers based on the Mersenne Twister helps to increase the cryptographic strength of the proposed algorithm. The algorithm we propose solves the issues of optimizing the costs of computing and energy resources of internet of things (IoT) devices, using a combination of lightweight cryptographic principles and fuzzy logic, and also provides reliable security and intelligent authentication of the mobile application user. The paper also considers the practical implementation of the proposed algorithm based on Arduino ESP32, a device with limited computing resources. In addition to this, fuzzy logic has found its practical application in the field of intelligent user authentication in developed mobile applications based on Arduino Studio for mobile cellular applications. As a result, the proposed lightweight encryption algorithm has proven itself to be an effective tool in ensuring the security of data transmission and communication between IoT devices.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Khuralay Moldamurat

Faculty of Information Technology, L.N. Gumilyov Eurasian National University

2 Satpayev str., Astana, Kazakhstan

Email: moldamurat@yandex.kz

## 1. INTRODUCTION

The application of internet of things (IoT) technology can be widely found in various fields such as transportation, agriculture, medicine and smart cities. Mainly, IoT devices collect data from diverse sensors and transmit it to the cloud. The number of IoT devices and their field of application is growing every year, which leads to an increase in the volume of transmitted data. In most cases, those data are confidential and sensitive, and most IoT devices have vulnerabilities. Consequently, they become the target of various fraud schemes and cyberattacks [1]–[3].

To solve security issues during data transmission, various encryption algorithms and authentication methods are used, but nevertheless, the usual traditional approaches are not suitable for IoT devices due to their limitations. Encryption methods such as data encryption standard (DES) and Rivest-Shamir-Adleman (RSA) provide a high level of cryptographic strength. However, they require high computing power and energy from the IoT device. Nevertheless, it should be noted that there are solutions based on the DES algorithm and field-programmable gate array (FPGA) platform for fast decision-making systems. But implementing such a system is expensive. Even a modified advanced encryption standard (AES) encryption algorithm with faster encryption

time and reduced central processing unit (CPU) usage has high requirements. Thus, the shortage of resources forces engineers to develop new methods and models for encrypting data [4]–[11].

One of the proposed authentication methods was a novel signature-based secure authentication mechanism implemented for the Core i3 processor. Since its implementation was executed for the computer processor, and not for the ARM architecture, it would most likely be demanding in terms of computing power and energy. It should be noted that solutions based on fuzzy logic can successfully solve the authentication problem since they do not require powerful computing and are completed in a short period of time. Notably, among decision-making and fault detection systems, systems based on fuzzy logic are popular [12]–[15].

This study we considered an enhanced cryptographic solution combining the characteristics of symmetric, asymmetric encryption algorithms and a public key server to encrypt transmitted data [16]. However, the previously mentioned algorithm AES was used to encrypt and decrypt data and has drawbacks due to the requirement of high computing power and energy resources. One of the works presented LoRa [17]. It was also based on the AES-128 algorithm. There was a work suggesting a hybrid encryption method [18] based on 10 methods, such as PRESENT-80, mCrypton, HIGHT, and others. Limited parameter analysis was presented to select the optimal hybrid encryption method, which might lead to long response times. In the next work [19], a dynamic data encryption algorithm with efficient encryption/decryption time was considered. Unfortunately, that proposed approach had limitations in image encryption. The application of blockchain technology has also found use in authentication and key exchange [20], [21]. However, the further scaling of blockchain technology leads to a decrease in transaction speed. Notably, the relevance of encryption extends beyond traditional fields, as evidenced by the innovative application of building damage assessment technology [22], highlighting the diverse applicability of data protection techniques. As previously noted, high-cost FPGAs are used for high-speed systems to solve complex tasks [23], [24], such as image recognition and video processing. However, each of the encryption methods considered had various disadvantages, such as high resource and power consumption, an expensive platform, moderate cryptographic strength, and long encryption/decryption execution time.

The study aims to develop a lightweight encryption algorithm and authentication method based on fuzzy logic for practical application on Arduino ESP32 with limited computing resources. The proposed lightweight encryption algorithm significantly increases cryptographic strength due to dynamic generations of S-boxes. The rest of the paper is organized as follows. Section 2 shows the process of identification and authentication procedure, then describes the architecture of the proposed Twine-Mersenne algorithm and compares it to other encryption algorithms. In section 3, the practical implementation of Twine-Mersenne algorithm and authentication are presented on the mobile application. Finally, in section 4 we conclude the paper with a brief summary and discuss the future work.

## 2. METHOD

### 2.1. Secure credential authentication

Secure credential (SC) authentication for IoT devices includes a multi-factor authentication mechanism coupled with advanced cryptographic techniques for enhanced security. This process ensures that only legitimate devices can connect to the network, providing strong protection against unauthorized access and attacks. The authentication process algorithm is shown in Figure 1.

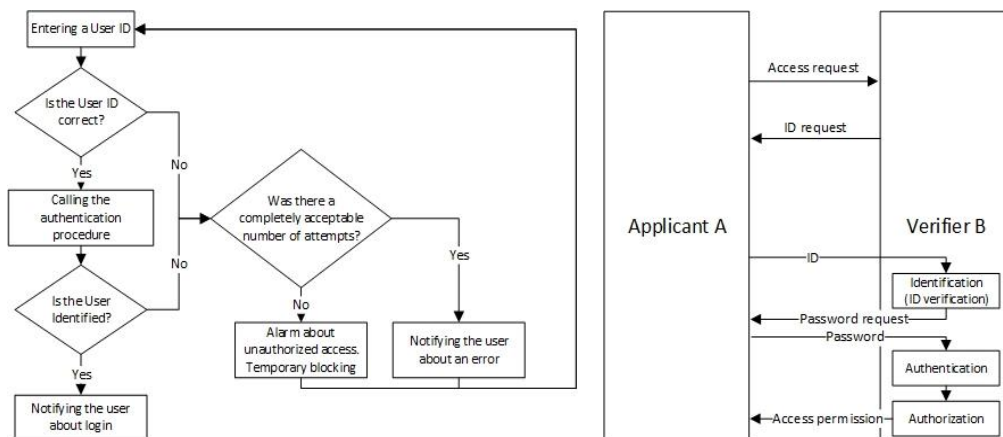


Figure 1. Identification and authentication procedure

## 2.2. Structure of intelligent authentication

Authorization in the digital world is key to ensuring access to information, traditionally relying on binary logic where access is either granted or denied based on strict rules. However, this approach often struggles with the complexities and nuances of modern access control scenarios. Fuzzy logic offers an alternative by introducing degrees of truth, allowing for more nuanced decisions that better reflect the ambiguity and variability of real-world situations. Implementing fuzzy logic in authorization systems enables more flexible and context-aware access decisions, adapting to a wide range of conditions beyond the binary. This adaptability is vital in today's dynamic digital environment, yet it also presents challenges, such as increased complexity and the need for a careful balance between security and usability. The relevance of advanced computational techniques for enhancing security controls is further highlighted by a studies [25]–[34] that examine the fundamental elements of secure systems, demonstrating the complex relationship between hardware-level innovation and the effectiveness of authorization mechanisms. Fuzzy logic operates with so-called "linguistic variables," and the general principle of operation is illustrated in Figure 2. These linguistic variables represent fuzzy sets, where the elements are values of some physical quantity. Each value of this quantity corresponds to a membership function, which takes values from 0 to 1, indicating the degree to which this quantity belongs to a particular set. Fuzzification, an integral part of fuzzy logic, transforms precise input data into fuzzy sets to manage ambiguity, enabling systems to make decisions with human-like reasoning. This approach is particularly useful in password verification, where it evaluates similarities between the attempted and expected passwords. The process relies on membership functions across three input dimensions of password verification: length match, character match, and case match. The implementation of fuzzy logic comprises four main modules:

- Rule base: This is a table of rules for evaluating various levels of criteria compliance and determining corresponding actions or outcomes. It contains all the "if-then" rules and conditions proposed by experts to manage the decision-making system.
- Fuzzification: This step helps convert input data into fuzzy sets and obtain the values of membership functions  $\mu$  in the interval  $[0,1]$ , where 1 represents complete truth and 0 represents absolute falsehood.
- Inference mechanism: This module calculates the membership functions for the output variable based on given logical rules.
- Defuzzification: This process converts the fuzzy output values back into physical values. There are various methods for defuzzification, so it is crucial to choose the optimal method best suited for use with the expert system.

When determining the shapes of membership function curves, it is important to note that they are constructed subjectively based on expert survey results and are, therefore, somewhat subjective. In practice, the shape of the membership function curves is chosen based on the complexity of the calculations. Triangular membership functions are most commonly used due to their versatility and lower computational resource requirements for hardware implementation. Additionally, in the task of intelligent user authentication for system access, it is necessary to have an exact match of parameters, meaning the membership function  $\mu$  should be equal to 1 at only one point for all input dimensions. This is achievable only with the use of triangular membership functions. The triangular membership function  $\mu$  is defined by a triplet of numbers  $(a, b, c)$ , and its value at point  $x$  is calculated according to the expression as shown in Figure 3. If  $(a - b) = (b - c)$ , then we get a symmetric triangular membership function, which can be uniquely defined by two parameters from the triplet  $(a, b, c)$ . The process of intelligent authentication in this implementation relies on membership functions across three input dimensions of password verification: length match, character match, and case match. Let's consider each of these criteria.

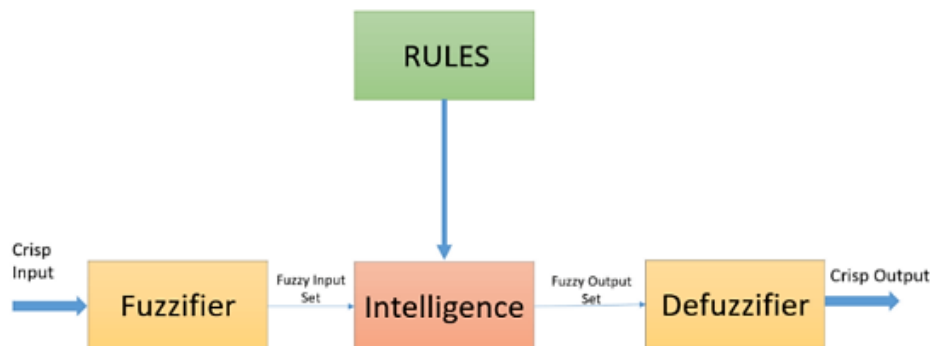


Figure 2. Structure of fuzzy logic operation

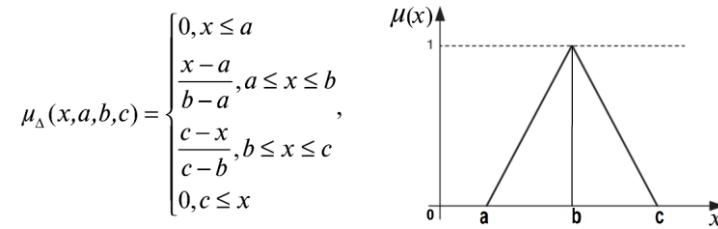


Figure 3. Triangular membership function

### 2.3. Length match criterion

This measurement evaluates how closely the length of the attempted password matches the expected password length. The membership function for this criterion classifies the length match into ranges such as short  $\mu_{Short}$ , adequate  $\mu_{Adequate}$ , or excessive  $\mu_{Excessive}$ , allowing flexible responses depending on the degree of match. In all three equations,  $L_a$  denotes the number of characters in the attempted password, and  $L_e$  denotes the expected number of characters in the password. The membership functions for this criterion are defined as:

- Short: indicates that the attempted password is significantly shorter than the expected one. The membership function  $\mu_{Short}$ , for this literal calculated as:

$$\mu_{Short} = \begin{cases} \frac{L_e - L_a}{L_e}, & \text{if } L_a < L_e \\ 0, & \text{otherwise} \end{cases}$$

- Adequate: represents a close or exact match in length between the attempted password and the expected password length. The membership function  $\mu_{Adequate}$  for this literal calculated as:

$$\mu_{Adequate} = \begin{cases} \frac{L_a}{L_e}, & \text{if } L_a \leq L_e \\ 1 - \frac{L_a - L_e}{L_e}, & \text{if } 2L_e > L_a > L_e \\ 0 & \text{otherwise} \end{cases}$$

- Excessive: indicates that the attempted password is significantly longer than the expected length. The membership function  $\mu_{Excessive}$  for this literal calculated as:

$$\mu_{Excessive} = \begin{cases} 1 & \text{if } L_a > 2L_e \\ \frac{L_a - L_e}{L_e}, & \text{if } 2L_e > L_a > L_e \\ 0 & \text{otherwise} \end{cases}$$

- Character match criterion: this measures the similarity of characters between the attempted password and the expected password. The evaluation determines the degree of character match, facilitating the next stage of security control. It is based on calculating the percentage of matching characters in the attempted password compared to the total number of characters in the password. The probability of character match ( $P_{symbol}$ ) is determined using the classic formula:

$$P_S = P_{symbol} = m/n$$

where  $m$  is the number of matching characters in the password attempt, and  $n$  is the total number of characters in the correct password.

- Mismatch: Indicates a significant difference in characters between the attempted and the expected password. The membership functions for this criterion  $\mu_{Mismatch}$  are defined as:

$$\mu_{Mismatch} = 1 - \frac{P_S}{100}$$

- Partial match: Reflects a moderate level of similarity in characters between the attempted and the expected password. The membership functions for this criterion  $\mu_{PartialMatch}$  are defined as:

$$\mu_{PartialMatch} = \begin{cases} 1 - \frac{P_s - 50}{50}, & \text{if } P_s > 50 \\ \frac{P_s}{50}, & \text{otherwise} \end{cases}$$

- Complete match: Indicates a high degree of similarity in characters between the attempted password and the expected password. The membership functions for this criterion  $\mu_{FullMatch}$  are defined as:

$$\mu_{FullMatch} = \frac{P_s}{100}$$

- Case match criterion: This evaluates the case sensitivity of the password attempt compared to the expected password, analyzing matches in terms of lowercase and uppercase letters, digits, or special characters. The literals and membership functions for the "Case match" input are identical to those for the "Character match" input, with the only difference being that they consider the percentage of matching cases ( $P_{case}$ ) instead of matching characters when the user inputs the password.

Let's consider the procedure for forming the rule base. Fuzzy rules are the foundation for decision-making in fuzzy logic-based systems. They are formulated as conditional expressions that link input variables to outputs, allowing the system to interpret and process ambiguous or partially true data to make decisions. Fuzzy rules serve to model complex decision-making processes, mimicking human reasoning and providing adaptability and flexibility under uncertainty. These rules are presented in Table 1, providing a clear structure (rule base) for evaluating various levels of criteria compliance and determining corresponding actions or outcomes.

Table 1. Rule base for the intelligent decision-making system

Length match	Character match	Case match	Output parameter
Adequate	Complete	Complete	Full access
Adequate	Partial	Complete	Conditional access
Adequate	Partial	Partial	Conditional access
Adequate	Partial	Mismatch	Conditional access
Adequate	Mismatch	Mismatch	Conditional access
Adequate	Partial	Partial	Access denied
Short	Partial	Partial	Access denied
Short	Partial	Mismatch	Access denied
Excessive	Partial	Mismatch	Access denied
Excessive	Partial	Mismatch	Temporary lock

Fuzzy inference, determined by the aggregation of various input factors through fuzzy logic rules, can be classified into four distinct outcomes:

- Full access: This outcome indicates complete approval of access without additional verification. It is applied when the system's evaluation strictly meets the security criteria.
- Conditional access: This indicates granting access but with the requirement of additional verification steps. This may involve re-entering the password or biometric confirmation to enhance security confidence.
- Access denied: Represents a scenario where access is not granted due to insufficient compliance with security criteria. This indicates a moderate deviation from expected parameters and may include a temporary restriction on access attempts.
- Temporary lock: Implies a significant deviation from access criteria or suspicious activity, leading to a prolonged suspension of access attempts and blacklisting the source.

Each of these outcomes allows the system to tailor responses to various levels of verification success, enhancing both security and flexibility in the authorization system. After application of fuzzy rules to the fuzzified inputs, defuzzification aims to obtain a quantitative value (crisp value) for each output linguistic variable. Formally, this is calculated by the formula:

$$y = \frac{\sum \mu_i * x_i}{\sum \mu_i}$$

Here,  $\mu_i$  represents the membership degree of the  $i$ -th output fuzzy set, and  $x_i$  represents the corresponding output value. The use of fuzzy logic in password verification broadens the scope for authorization systems by accounting for the complexity of human-entered data. This improves both security and user experience, enabling effective decision-making in access control. In a fuzzy logic-based authorization system, decision-making goes beyond binary outcomes, providing a more nuanced, intelligent approach to access control. This not only allows for immediate rejection of service but also gives the user the opportunity to correct input errors. In cases of radical non-compliance, such as in a brute-force attack on the authorization system, the intelligent system can blacklist the source.

Data encryption algorithm Twine-Mersenne. Traditional algorithms such as RSA and elliptic curve cryptography (ECC) are not suitable for most IoT environments due to the complexity of encryption and decryption operations. The proposed method includes various security mechanisms to protect data and uses clustering based on an intelligent algorithm, namely fuzzy logic, as well as data encryption in the form of a lightweight Twine-Mersenne algorithm, the feature of which is the dynamic generation of S-boxes.

Algorithm for dynamic generation of S-boxes. Mersenne Twister (MT) is a widely used pseudorandom number generator, and it is named after the Mersenne primes, a class of prime numbers named after the French mathematician Marin Mersenne. Mersenne Twister is known for providing a very long period (the length of the cycle after which the sequence of numbers repeats) and for being fairly efficient in computation. Another version of the Mersenne Twister algorithm that is widely used in scientific literature is MT19937, which indicates that the period is  $2^{19937}-1$ , which is the prime Mersenne number. There is also a 64-bit variant known as MT19937-64.

Using a shared initial key, the sender and receiver can independently create a matching series of S-boxes, which is important for maintaining consistency in cryptographic protocols as shown in Figure 4. This method allows for secure, efficient communication without transmitting actual S-boxes, minimizing the risk of interception. It combines the benefits of randomness for security with the determinism needed for strong encryption and decryption-called a diffusion layer. Each S-box is unique and generated from different seeds derived from the Mersenne Twister, enhancing the encryption strength. Furthermore, this scalable technique allows for synchronized encryption states and can adapt to various security levels, making it highly practical for real-world applications.

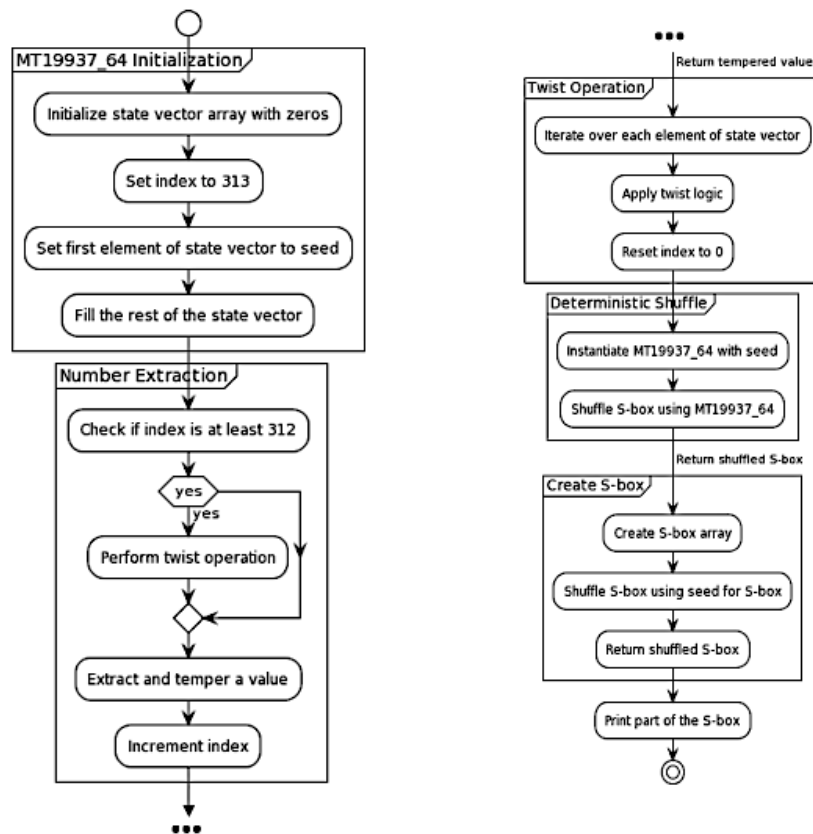


Figure 4. The flowchart of generating S-boxes using the Mersenne Twister algorithm



Building the Mersenne Twister MT19937-64 algorithm implementation, let's extend the usage to generate S-boxes. Each S-box generation will be based on a new pseudo-random number extracted from the MT instance, which is seeded initially with a shared key. To create an S-box, we use an input counter from 0 to 255. This approach ensures both the sender and receiver, who know the initial shared key, can generate the same sequence of S-boxes independently. This algorithm defines the Mersenne Twister class (MT19937\_64) as a deterministic shuffle function that uses Mersenne Twister to shuffle an array and a function to create an S-box (*create\_sbox*) by generating a seed for the shuffle using the shared key. The *create\_sbox* function initializes a list representing an 8-bit S-box, then shuffles it deterministically using a pseudo-random number as the seed, ensuring both sender and receiver can generate the same S-box independently as long as they share the initial key. The Tiny Mersenne Twister (TinyMT) is a variant of the original Mersenne Twister designed for applications that require smaller state size and memory footprint. It was developed to retain the original MT's strong pseudo-random properties while being better suited for use in environments with limited resources, such as embedded systems or when handling multiple streams of random numbers is needed in Figure 5. This algorithm defines the TinyMT64 class, which implements a simplified version of the TinyMT algorithm for generating 64-bit pseudorandom numbers. It includes an initialization method that sets up an internal state with a given seed and predefined parameters.

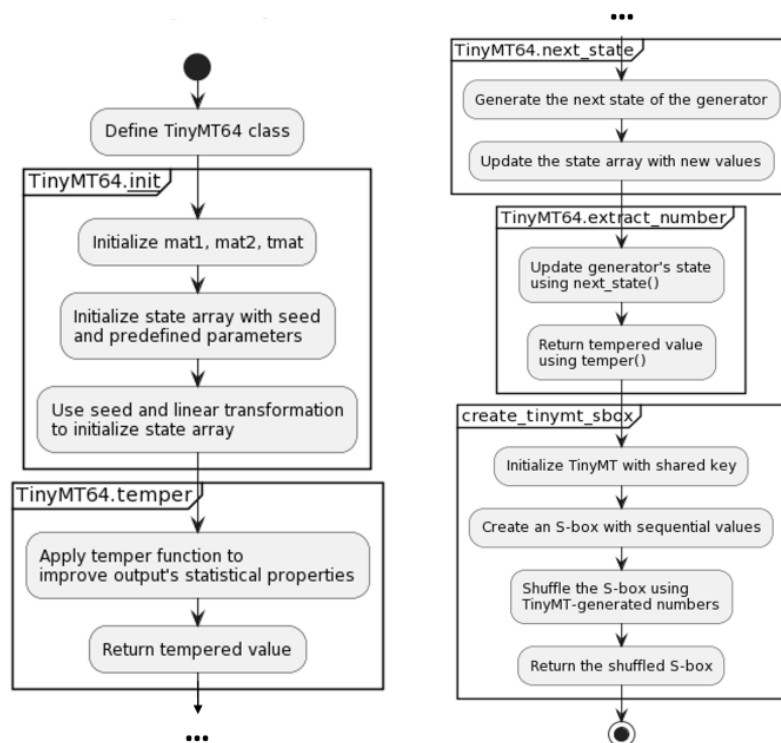


Figure 5. The flowchart of generating S-boxes using the TinyMT algorithm

The *next\_state* method updates the generator's internal state, and *extract\_number* produces a tempered pseudorandom number from the current state. The *create\_tinynt\_sbox* function demonstrates the use of TinyMT to create a shuffled S-box, which is a key component in many cryptographic systems, ensuring that a sender and receiver with a shared key can independently generate the same pseudorandom sequence for secure communication or data encryption. In simulations with many instances of random number generators (e.g., Monte Carlo simulations), TinyMT's smaller state size allows for more efficient use of memory. This can be particularly beneficial in parallel computing environments or applications running on GPUs where memory resources are shared among many threads.

Despite its smaller state size, TinyMT is designed to have good statistical properties for its outputs, making it suitable for various applications, including simulations and games. However, it is important to note that while TinyMT is designed for quality outputs, the specific requirements of an application (e.g., cryptographic security) should guide the choice of a random number generator. The proposed Twine-Mersenne algorithm is a lightweight 64-bit block encryption algorithm based on the Twine algorithm with the addition of a random number generator based on the Mersenne Twister for the (dynamic) generation of

Sboxes. It is also capable of generating keys from 80 to 128 bits like the Twine algorithm but differs in that Sboxes are generated dynamically for each session and this significantly increases the cryptographic strength of the algorithm. It is worth noting that based on the generated keys of different sizes (80 and 128 bits), there are two versions of the lightweight encryption algorithm, Twine-80 and Twine-128. In accordance with the longest key length, Twine-128 has the optimal degree of data encryption security. However, unlike the proposed Twine-Mersenne algorithm, the process of generating S-boxes in the Twine algorithm is static.

Comparative analysis of encryption algorithms. The analysis was conducted to compare the performance of different data encryption algorithms on a desktop personal computer (PC) with a 3.10 GHz Intel® Core™ i5 processor, 8 GB of random-access memory (RAM), and Windows 11 Professional. MATLAB R2020b was used to model and evaluate encryption performance. Table 2 shows the comparative effectiveness of several encryption algorithms in relation to Twine-80, Twine-128 and the proposed Twine-Mersenne algorithm. Observations indicate that Twine's use of a 64-bit block is relatively standard and efficient, compared to the block size of many other algorithms. On the contrary, unlike Twine-80, the key size of Twine-128 is 128 bits, which is noticeably larger than its counterparts in the table. The things that stand out about the Twine-80 and Twine-128 are the smaller code size of just 414 and 216 bytes, coupled with modest RAM usage of 6.75 and 5.93 bytes. These attributes greatly highlight Twine's minimalist approach. To quickly assess the cryptographic strength of various algorithms, a test was carried out with the generation of short keys of no more than 5 bits in size. Subsequently, various attack scenarios were simulated, such as saturation attack, impossible differential attack, key schedule-based attacks and others. The result of cryptographic strength for various encryption algorithms is given in the last column of Table 2. The Twine-128 algorithm runs for 36 and 24 rounds for Twine-80, which meets the need for robust security measures. Minimal code and memory requirements make Twine a viable option for mobile IoT devices, radio-frequency identification (RFID) tags, and other devices with limited computational resources, making it the most optimal encryption solution in the field. However, the proposed Twine-Mersenne algorithm also runs for 24 rounds, and has a compact code size and minimal RAM usage, in addition to this, it has an increased cryptographic strength.

Table 2. Comparative analysis between the Twine-Mersenne, Twine and other encryption algorithms

Encryption algorithm	Block size, bit	Key size, bit	Number of rounds	Code size, bytes	RAM, byte	Level of strong cryptography
PRESENT	64	80	32	1738	274	Moderate
Simon	64	96	42	1370	188	Moderate
Speck	64	96	26	2552	124	Moderate
SIT	64	64	5	826	22	Moderate
AES	128	128	10	23090	720	High level
LEA	128	128	24	3700	432	High level
RC5	64	128	20	20044	360	Moderate
HIGHT	64	128	32	13476	288	Moderate
Lightweight encryption algorithm	64	128	9	823	144	High level
Twine-128	64	128	36	414	6.75	High level
Twine-80	64	80	24	216	5.93	Moderate
Twine-Mersenne-80	64	80	24	473	11.48	High level

This feature is also beneficial for devices with limited computing capabilities and the presence of dynamic generation of S-boxes makes it an ideal candidate for deployment in the most vulnerable node of the IoT network. In addition, it is worth noting the prospect of developing the Twine-Mersenne encryption algorithm aimed at reducing energy consumption and increasing the efficiency of encryption/decryption processes in real time. The proposed Twine-Mersenne algorithm has 16 subblocks of 4 bits each in Figure 6. The pseudocode describes the encryption algorithm in Figure 7 Twine-Mersenne, which generates a secret key. As previously noted, the proposed lightweight Twine-Mersenne encryption algorithm is ideal for its implementation on embedded systems with limited computing resources.

The proposed Twine-Mersenne algorithm is easily implemented on IoT devices, in particular on platforms such as FPGA, single-board computer Raspberry Pi and Arduino microcontroller. The choice of device type depends on the task being implemented, its complexity, the required speed of the solution, the speed of data encryption and financial capabilities. It is recommended to use FPGAs to solve complex problems that require high processing speed. In turn, it is recommended to use Raspberry Pi to solve problems using artificial intelligence. The Arduino microcontroller is quite suitable for simple data encryption tasks. The implementation diagram of the general encryption process based on Twine-Mersenne is presented in Figure 8.



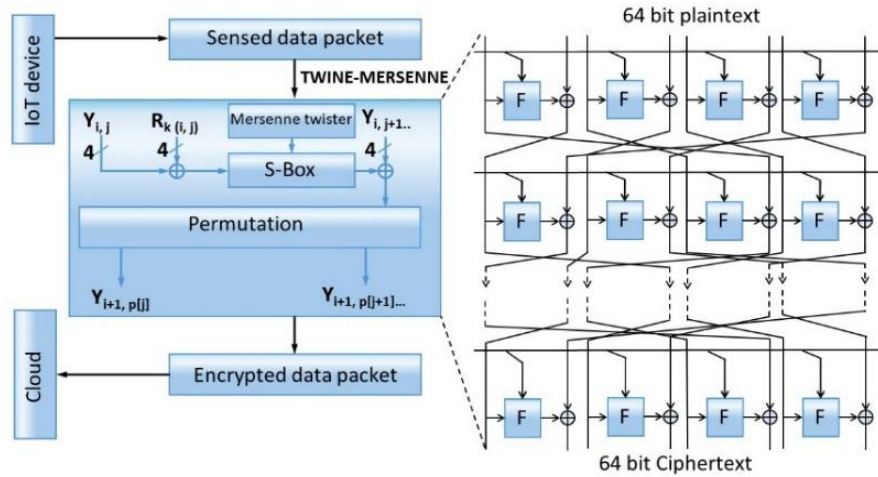


Figure 6. The architecture of the lightweight Twine-Mersenne encryption algorithm

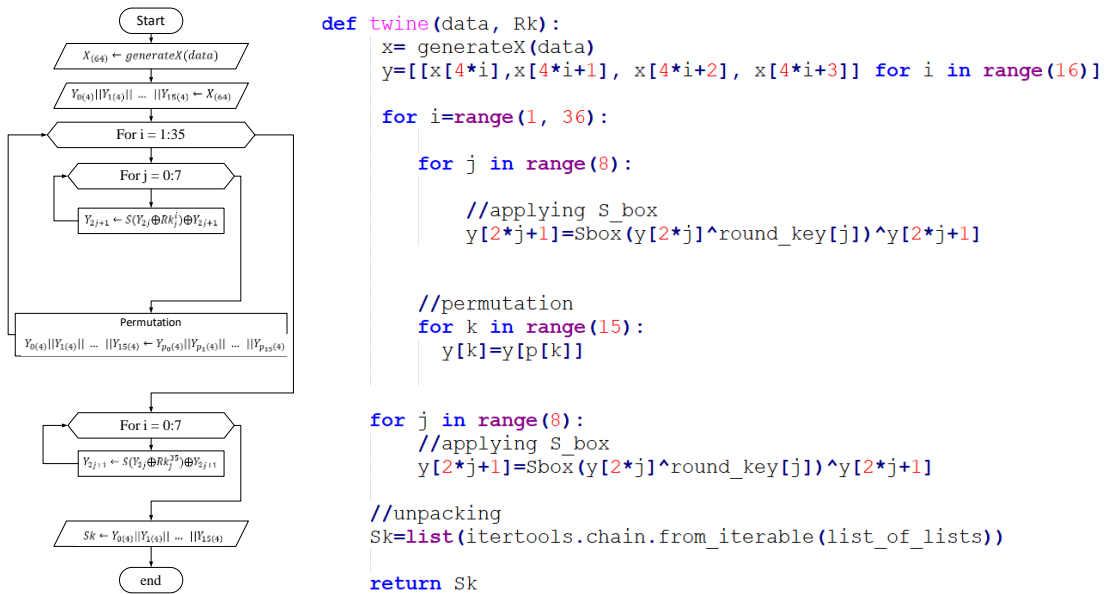


Figure 7. Flowchart and pseudocode for the Twine-Mersenne encryption algorithm

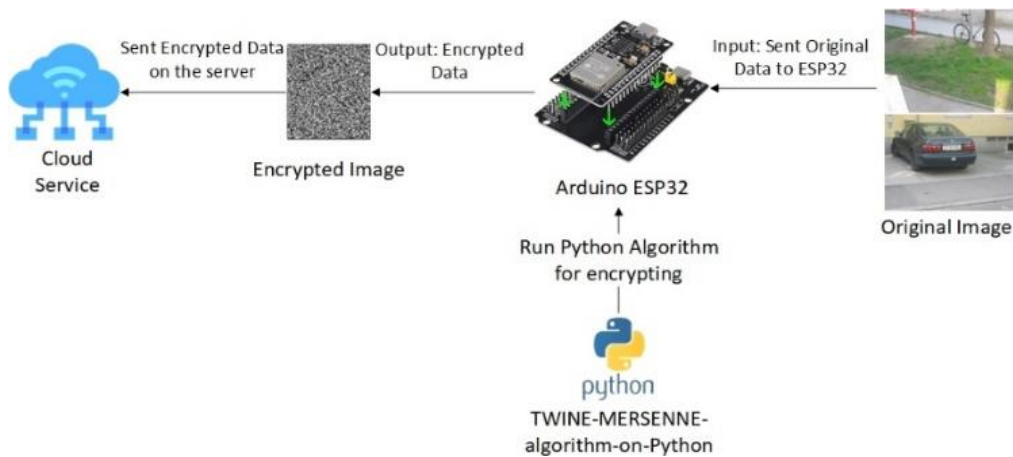


Figure 8. General diagram of the process of encrypting input data based on the Twine-Mersenne algorithm

The Android Studio environment was used to implement this algorithm for cellular communications. In our case, a microcontroller based on Arduino ESP32, the simplest and most common on the internet of things environment, was used as an IoT device for the simplest visualization of the data encryption process. The Python programming language was used to implement the lightweight Twine-Mersenne encryption algorithm. In this implementation, the input data to the Arduino ESP32 microcontroller is supplied in the form of graphic images from a smartphone or digital video recorder (DVR). Next, the previously prepared Twine-Mersenne data encryption algorithm implemented in the Python programming language is launched on the Arduino ESP32. The algorithm performs a series of operations using XOR, dynamic substitution (S) and permutation (p). As a result, encrypted data is transmitted over a secure communication channel to the cloud, where end users already have the key to decrypt the data.

### 3. RESULT AND DISCUSSION

Figure 9 shows the result of encrypting input data in the form of images received from a smartphone and a DVR using the lightweight Twine-Mersenne encryption algorithm. The development of a fuzzy logic-based authorization algorithm is a significant advance in the field of Android application security. The algorithm, created for the convenience of users, demonstrates the seamless integration of complex security measures in the application. Figure 10 shows a diagram of the operation of fuzzy logic that implements decision-making algorithms. These elements together form a robust structure that strengthens the application and ensures a smooth user experience with security features.



Figure 9. Result of image processing using the Twine-Mersenne encryption algorithm

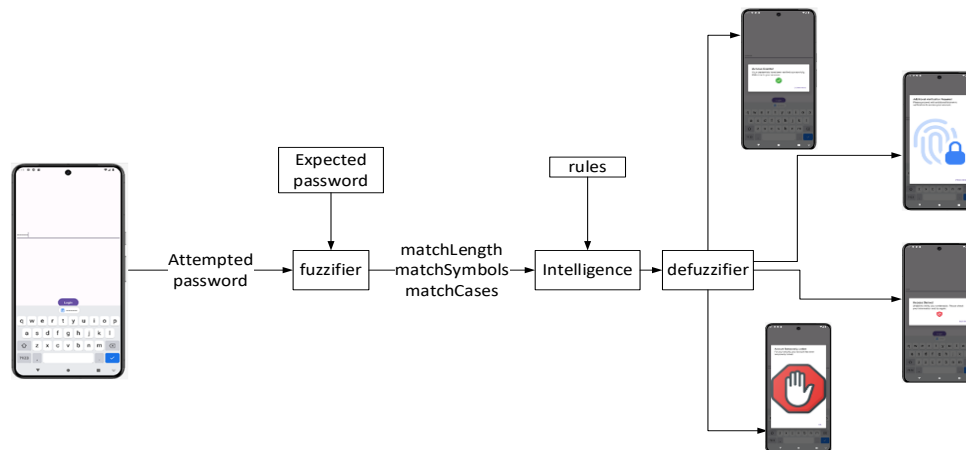


Figure 10. Operating principle of fuzzy logic

The key features and advantages of the proposed model of cryptographic protection and authentication for IoT devices are:

- Ease of implementation and cost-effectiveness. The model promises to be cost-effective due to the use of algorithms with small key size and low power consumption. This is important for IoT devices as they reduce hardware and security costs.
- Applicability in various fields. The proposed model is not limited to specific IoT application areas such as smart homes or medical wearables. It can be adapted for a wide range of applications, including cellular security, making it versatile and attractive to developers.

- Use of dynamically generated S-boxes. Dynamic S-boxes strengthen encryption security and make the algorithm less vulnerable to various types of attacks. This significantly increases the reliability of the proposed model in comparison with other lightweight ciphers.
- Practical implementation on Arduino ESP32. The successful practical implementation of the algorithm on the Arduino ESP32 platform confirms its applicability in real-world conditions with limited computing resources and power consumption. This is significant for IoT projects, where not only cryptographic security is important, but also the economic efficiency of the solution.

#### 4. CONCLUSION

Since the interconnected nature of IoT devices and cloud server systems makes them vulnerable to various types of cyber-attacks, a scheme for providing cryptographic data transmission security and intelligent authentication for IoT devices has been proposed. The use of the Twine-Mersenne algorithm with dynamically generated S-blocks for encryption and the implementation of mutual authentication and confidentiality will provide a high level of protection with inexpensive hardware implementation. The proposed scheme is just one aspect of providing security for a wide range of applications in areas such as smart homes, intelligent lighting systems, home climate control systems, security applications or wearable sensors for healthcare applications.

In this paper, we considered a lightweight cryptographic algorithm that is used to ensure secure data transmission and secure communication between the smart IoT devices and the cloud server. Considering the small key size, low computing power, and low energy consumption, a simplified authentication and encryption mechanism was proposed without compromising the quality of protection. For future works, we will provide proof of concepts for protection against various attacks, the effectiveness of protection for various hardware platforms, and the feasibility of using blockchain technology.

#### ACKNOWLEDGEMENTS

This work is financed from the republican budget of the Science Committee of the Ministry of Science and Higher Education within the framework of program-targeted funding (Program No. BR18574045).





#### REFERENCES

- [1] S. Villamil, C. Hernandez, and G. Tarazona, "An overview of internet of things," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2320–2327, Oct. 2020, doi: 10.12928/telkomnika.v18i5.15911.
- [2] M. Arief *et al.*, "A novel framework for analyzing internet of things datasets for machine learning and deep learning-based intrusion detection systems," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 2, pp. 1574–1584, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1574-1584.
- [3] A. Tuscano and S. Joshi, "Significance of cyber security of IoT devices in the healthcare sector," in *2023 Somaiya International Conference on Technology and Information Management (SICTIM)*, 2023, pp. 12–16, doi: 10.1109/SICTIM56495.2023.10104657.
- [4] I. Hussain, M. C. Negi, and N. Pandey, "A secure IoT-based power plant control using RSA and DES encryption techniques in data link layer," in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, Dec. 2017, pp. 464–470, doi: 10.1109/ICTUS.2017.8286054.
- [5] S. R. M. Zeebaree, "DES encryption and decryption algorithm implementation based on FPGA," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 774–781, May 2020, doi: 10.11591/ijeecs.v18.i2.pp774-781.
- [6] H. V. Gamido, A. M. Sison, and R. P. Medina, "Modified AES for text and image encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 942–948, Sep. 2018, doi: 10.11591/ijeecs.v11.i3.pp942-948.
- [7] M. Saleh, N. Jhanjhi, A. Abdullah, and R. Saher, "Proposing encryption selection model for IoT devices based on IoT device design," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2022, pp. 210–219, doi: 10.23919/ICACT53585.2022.9728914.
- [8] M. Imdad, D. W. Jacob, H. Mahdin, Z. Baharum, S. M. Shaharudin, and M. S. Azmi, "Internet of things: security requirements, attacks and counter measures," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1520–1530, Jun. 2020, doi: 10.11591/ijeecs.v18.i3.pp1520-1530.
- [9] F. Abdou Vadhil, M. Lemine Salihi, and M. Farouk Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 711–721, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp711-721.
- [10] R. Sudarmani, K. Venusamy, S. Sivaraman, P. Jayaraman, K. Suriyan, and M. Alagarsamy, "Machine to machine communication enabled internet of things: a review," *International Journal of Reconfigurable and Embedded Systems*, vol. 11, no. 2, pp. 126–134, Jul. 2022, doi: 10.11591/ijres.v11.i2.pp126-134.
- [11] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 110–120, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp110-120.
- [12] N. Fathima, R. Banu, and G. F. A. Ahammed, "A signature-based data security and authentication framework for internet of things applications," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 3298–3308, Jun. 2022, doi: 10.11591/ijece.v12i3.pp3298-3308.
- [13] R. Asati, D. S. Bankar, A. Apte, A. L. Nehete, and Y. Mandake, "Fuzzy controlled modified reduced switch converter for switched reluctance motor under dynamic loading," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 50–58, Apr. 2024, doi: 10.11591/ijeecs.v34.i1.pp50-58.




- [14] B. A. Dapshima, R. Mishra, and P. Tyagi, "Transformer faults identification via fuzzy logic approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 3, pp. 1327–1335, Mar. 2024, doi: 10.11591/ijeecs.v33.i3.pp1327-1335.
- [15] Z. Y. Seitbattalov, S. K. Atanov, and Z. S. Moldabayeva, "An intelligent decision support system for aircraft landing based on the runway surface," in *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, Apr. 2021, pp. 1–5, doi: 10.1109/SIST50301.2021.9466000.
- [16] R. Bhandari and K. V B, "Enhanced encryption technique for secure iot data transmission," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 3732–3738, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3732-3738.
- [17] G. Mao *et al.*, "REALISE-IoT: RISC-V-based efficient and lightweight public-key system for IoT applications," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 3044–3055, Jan. 2024, doi: 10.1109/JIOT.2023.3296135.
- [18] L. Ning, Y. Ali, H. Ke, S. Nazir, and Z. Huanli, "A hybrid MCDM approach of selecting lightweight cryptographic cipher based on ISO and NIST lightweight cryptography security requirements for internet of health things," *IEEE Access*, vol. 8, pp. 220165–220187, 2020, doi: 10.1109/ACCESS.2020.3041327.
- [19] D. Khwailleh and F. Al-balas, "A dynamic data encryption method based on addressing the data importance on the internet of things," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 2139–2146, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2139-2146.
- [20] F. Kasfa Ali and S. Mathew, "An efficient lightweight key exchange algorithm for internet of things applications," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 5, pp. 5609–5618, Oct. 2022, doi: 10.11591/ijece.v12i5.pp5609-5618.
- [21] M. Parmar and P. Shah, "Internet of things-blockchain lightweight cryptography to data security and integrity for intelligent application," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 4, pp. 4422–4431, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4422-4431.
- [22] Y. Nihad Hatif, Y. Amer Abbas, and M. Hussein Ali, "Lightweight ANU-II block cipher on field programmable gate array," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 2194–2205, Jun. 2022, doi: 10.11591/ijece.v12i3.pp2194-2205.
- [23] W. A. Al-Musawi, M. A. Ali Al-Ibadi, and W. A. Wali, "Artificial intelligence techniques for encrypt images based on the chaotic system implemented on field-programmable gate array," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, pp. 347–356, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp347-356.
- [24] A. Akhmediya, N. Nabiyev, K. Moldamurat, K. Dyussekeyev, and S. Atanov, "Use of sentinel-1 dual polarization multi-temporal data with gray level co-occurrence matrix textural parameters for building damage assessment," *Pattern Recognition and Image Analysis*, vol. 31, no. 2, pp. 240–250, Apr. 2021, doi: 10.1134/S1054661821020036.
- [25] S. Brimzhanova, S. Atanov, K. Moldamurat, B. Baymuhambetova, K. Brimzhanova, and A. Seitmetova, "An intelligent testing system development based on the shingle algorithm for assessing humanities students' academic achievements," *Education and Information Technologies*, vol. 27, no. 8, pp. 10785–10807, Sep. 2022, doi: 10.1007/s10639-022-11057-w.
- [26] Y. N. Seitkulov, S. N. Boranbayev, G. B. Ulyukova, B. B. Yergaliyeva, and D. Satybaldina, "Methods for secure cloud processing of big data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 3, pp. 1650–1658, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1650-1658.
- [27] B. Yergaliyeva, Y. Seitkulov, D. Satybaldina, and R. Ospanov, "On some methods of storing data in the cloud for a given time," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 2, pp. 366–372, Apr. 2022, doi: 10.12928/telkomnika.v20i2.21887.
- [28] A. Ullah, I. Laassar, C. B. Şahin, O. B. Dinle, and H. Aznaoui, "Cloud and internet-of-things secure integration along with security concerns," *International Journal of Informatics and Communication Technology*, vol. 12, no. 1, pp. 62–71, Apr. 2023, doi: 10.11591/ijict.v12i1.pp62-71.
- [29] M. Lindsay and M. Emimal, "Fuzzy logic-based approach for optimal allocation of distributed generation in a restructured power system," *International Journal of Applied Power Engineering*, vol. 13, no. 1, pp. 123–129, Mar. 2024, doi: 10.11591/ijape.v13.i1.pp123-129.
- [30] M. A. Jasim and T. S. Atia, "An IoT-fuzzy based password checker system for wireless video surveillance system," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 6, pp. 3441–3449, Dec. 2022, doi: 10.11591/eei.v11i6.4375.
- [31] L. A. Saddik, B. A. Khalifa, and B. Fateh, "Evaluation quality of service for internet of things based on fuzzy logic: a smart home case study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 825–839, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp825-839.
- [32] M. Gassara, M. Elleuchi, and M. Abid, "Cloud-based platforms for LoRa internet of things: a survey," *International Journal of Informatics and Communication Technology*, vol. 10, no. 1, pp. 54–64, Apr. 2021, doi: 10.11591/ijict.v10i1.pp54-64.
- [33] A. A. Talib and A. D. Salman, "Development of an electronic payment system using the Internet of things," *IAES International Journal of Robotics and Automation*, vol. 11, no. 3, pp. 213–222, Sep. 2022, doi: 10.11591/ijra.v11i3.pp213-222.
- [34] M. A. Al-Shareeda, M. A. Saare, and S. Manickam, "The blockchain internet of things: review, opportunities, challenges, and recommendations," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 3, pp. 1673–1683, Sep. 2023, doi: 10.11591/ijeecs.v31.i3.pp1673-1683.

## BIOGRAPHIES OF AUTHORS






**Sabyrzhan Atanov**     he is a doctor of technical sciences and a professor at the Faculty of Information Technologies at L.N. Gumilyov ENU. He received several inventions, for which he has a number of patents and certificates. In particular, he received patents for inventions in the field of mobile communications. Also, he has published many original scientific articles in leading world journals and conference proceedings. He has spoken at many international conferences. He specializes in computer security. Over a 40-year scientific career, he led a number of scientific projects that were approved by national scientific councils. He was awarded special prizes for the best university teacher. He is always available at the following email: Atanov5@mail.ru.






**Yerzhan Seitkulov**    has a PhD in physics and mathematics and is a renowned specialist in the field of mathematical problems in the information security area. Professor of the IT Department at L.N. Gumilyov ENU. In 2006 he became a laureate of the Daryn Prize from the government of the RK. In 2008-2012 he received a special scientific scholarship for young scientists from the government. In 2012, he completed a postdoctoral internship at the University of New York. He has published more than 75 original scientific works in leading publications around the world. Currently, he combines scientific work with practical activities in government agencies. Over the past 12 years, he has managed several projects and programs through the science committee and other government bodies. He is always available at the following email: [yerzhan.seitkulov@gmail.com](mailto:yerzhan.seitkulov@gmail.com).






**Khuralay Moldamurat**    she is a candidate of technical sciences and defended her dissertation at the Institute of Mathematics of Kazakhstan. Now she is a docent at the IT Department at the L.N. Gumilyov ENU. She has published more than 40 scientific articles in international publications. She has 15 years of successful experience in scientific and teaching activities. She took part in various scientific projects funded by government agencies. She is known in the scientific community as a specialist in the field of cryptography and cloud computing. She is always available at the following email: [moldamurat@yandex.kz](mailto:moldamurat@yandex.kz).






**Banu Yergaliyeva**    she was born in the Kostanay region in 1985, then went to college in the Arkalyk. In 2000, she entered the Department of Mathematics and Computer Science. She is a specialist in the field of cryptography, currently a doctoral student at L.N. Gumilyov ENU in the information security and cryptology area. Her research fields cloud computing, parallel computing, and cryptology. She is always available at the following email: [banu.yergaliyeva@gmail.com](mailto:banu.yergaliyeva@gmail.com).



**Abzal Kyzyrkanov**    doctoral student in the Department of Computer and Software Engineering, L.N. Gumilyov ENU. senior lecturer, Department of Computer Engineering, Astana IT University. He has a number of publications in the fields of coding theory, information theory, cryptography, cloud computing, computer security, parallel computing, as well as in the field of circuit design and robotics. He is always available at the following email: [kyzyrkanov.abzal@astanait.edu.kz](mailto:kyzyrkanov.abzal@astanait.edu.kz), [abzzall@gmail.com](mailto:abzzall@gmail.com).



**Zhexen Seitbattalov**    he received MSc degree in computing and software engineering in 2018 from Agro University in Astana city. Now he is a teacher and a PhD student in computer and software engineering of the L.N. Gumilyov ENU. He has a number of publications in the fields of coding theory, information theory, cryptography, cloud computing, computer security, parallel computing, as well as in the field of circuit design and robotics, edge computing, computer vision, deep learning. He is always available at the following email: [sbtl.jeks@gmail.com](mailto:sbtl.jeks@gmail.com).