IRSTI 81.93.29 **https://doi.org/10.58805/kazutb.v.3.28-1183**

LIGHTWEIGHT CRYPTO SYSTEM BASED ON DFA AND LORENZ CHAOS FOR SECURE DATA TRANSFER IN IOT ON MICROCONTROLLERS

[1,2]A.A. Kulzhanova, [1]Sh.A. Jomartova, [3]V.A. Zhmud, [1,2]T.Zh. Mazakov

[1]al-Farabi Kazakh National University, Almaty, Kazakhstan,
[2]International Engineering-Technological University, Almaty, Kazakhstan,
[3]Novosibirsk State Technical University, Novosibirsk, Russia
Correspondent-author: e-mail: jomart63@gmail.com

This paper describes a simpler version of the cryptographic system designed to secure the communications of low-power Internet of Things (IoT) devices operating in low-resource, low-computation, and low-speed environments. The system in question is based on a hybrid cryptographic model that includes a certain type of hashing with Deterministic Finite Automata (DFA) and key creation through a chaotic dynamic system, the Lorentz attractor. This model strikes a good balance of tradeoff between resource simplification and dependability. A prototype system was developed first on the Arduino Uno, then later improved on the Wokwi simulator, which enabled the simulation of the IoT devices' actual operating environment. The automata perform a hashing which is both robust and deterministic, which helps to confirm the integrity of the data. The keys generated by the Lorentz system provide deterministic yet highly sensitive keys to perturbations ensuring total unpredictability. The encryption method used is also quite simple, and that is the XOR operation. This makes the entire system compact, fast, easily reversible, and resource efficient. This research shows how useful automatism and chaos theory are in developing cyber security measures for the Internet of Things (IoT) at the embedded system levels.

Keywords: data encryption, Lorenz attractor, key generation, data security, information privacy, data authorization, digital security, chaotic systems, Internet of things.

МИКРОБАҚЫЛАУЛАРДА IOT-ТА ДЕРЕКТЕРДІ ҚАУІПСІЗ ТАСЫМАЛДАУ ҮШІН DFA ЖӘНЕ ЛОРЕНЦ ХАОСЫНА НЕГІЗГЕН ЖЕҢІЛ САЛМАҚ КРИПТО ЖҮЙЕСІ

[1,2]А.А. Кульжанова, [1]Ш.А. Джомартова, [3]В.А. Жмудь, [1,2]Т.Ж. Мазаков

[1]әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан,
[2]Халықаралық инженерлік-технологиялық университет, Алматы, Қазақстан,
[3]Новосибирск мемлекеттік техникалық университеті, Новосибирск, Ресей,
e-mail: jomart63@gmail.com

Бұл мақалада ресурстары аз, есептеулері төмен және жылдамдығы төмен орталарда жұмыс істейтін төмен қуатты заттар интернеті (IoT) құрылғыларының байланысын қамтамасыз етуге арналған криптографиялық жүйенің қарапайым нұсқасы сипатталған. Қарастырылып отырған жүйе гибридті криптографиялық модельге негізделген, ол Детерминирленген Ақырлы Автоматтар (DFA) хэштеудің белгілі бір түрін және хаотикалық динамикалық жүйесі мен Лоренц аттракторы арқылы кілттерді жасауды қамтиды. Бұл модель ресурстарды жеңілдету мен сенімділік арасын келудің жақсы тепе-теңдігін қамтамасыз етеді. Прототиптік жүйе алдымен Arduino Uno-да жасалды, содан кейін Wokwi тренажерінде жетілдірілді, бұл IoT құрылғыларының нақты жұмыс ортасын модельдеуге мүмкіндік берді. Автоматтар деректердің тұтастығын растауға көмектесетін сенімді және детерминирленген хэштеуді орындайды. Лоренц жүйесі жасаған кілттер детерминирленген, бірақ сонымен бірге өте сезімтал пертурбация кілттерін қамтамасыз етеді, бұл толық болжау мүмкіндігін қамтамасыз етеді. Қолданылатын шифрлау әдісі де өте қарапайым және бұл XOR операциясы. Бұл бүкіл жүйені ықшам, жылдам, оңай қайтымды және ресурстарды үнемдейтін етеді. Бұл зерттеу автоматизм мен хаос теориясының ендірілген жүйе деңгейлерінде заттар интернеті (IoT) үшін киберқауіпсіздік шараларын әзірлеуде қаншалықты пайдалы екенін көрсетеді.

Түйін сөздер: деректерді шифрлау, Лоренц аттракторы, кілттерді генерациялау, деректер қауіпсіздігі, ақпараттың құпиялылығы, деректерді авторизациялау, цифрлық қауіпсіздік, хаотикалық жүйелер, заттардың интернеті.

ЛЁГКАЯ КРИПТОСИСТЕМА НА ОСНОВЕ DFA И ХАОСА ЛОРЕНЦА ДЛЯ ЗАЩИЩЁННОЙ ПЕРЕДАЧИ ДАННЫХ В IOT НА МИКРОКОНТРОЛЛЕРАХ

[1,2]А.А. Кульжанова, [1]Ш.А. Джомартова✉, [3]В.А. Жмудь, [1,2]Т.Ж. Мазаков

[1]Казахский национальный университет имени аль-Фараби, Алматы, Казахстан,
[2]Международный инженерно-технологический университет, Алматы, Казахстан,
[3]Новосибирский государственный технический университет, Новосибирск, Россия,
e-mail: jomart63@gmail.com

В этой статье описывается более простая версия криптографической системы, предназначенной для защиты связи маломощных устройств Интернета вещей (IoT), работающих в средах с низким потреблением ресурсов, низкими вычислительными затратами и низкой скоростью передачи данных. Рассматриваемая система основана на гибридной криптографической модели, которая включает определенный тип хеширования с использованием детерминированных конечных автоматов (DFA) и создание ключей с помощью хаотической динамической системы, аттрактора Лоренца. Эта модель обеспечивает хороший баланс между упрощением использования ресурсов и надежностью. Прототип системы был разработан сначала на Arduino Uno, а затем усовершенствован на симуляторе Wokwi, который позволил имитировать реальную среду работы устройств Интернета вещей. Автоматы выполняют надежное и детерминированное хэширование, что помогает подтвердить целостность данных. Ключи, генерируемые системой Лоренца, предоставляют детерминированные, но высокочувствительные ключи к возмущениям, гарантирующие полную непредсказуемость. Используемый метод шифрования также довольно прост и заключается в операции исключающего преобразования. Это делает всю систему компактной, быстрой, легко обратимой и ресурсосберегающей. Данное исследование показывает, насколько полезны автоматизм и теория хаоса при разработке мер кибербезопасности для Интернета вещей (IoT) на уровне встроенных систем.

Ключевые слова: шифрование данных, аттрактор Лоренца, генерация ключей, безопасность данных, конфиденциальность информации, авторизация данных, цифровая безопасность, хаотические системы, интернет вещей.

Introduction. Special attention is paid to the development of the Internet of Things (IoT) in environments with limited power consumption and storage devices Traditional encryption methods like RSA and AES just don’ t cut it for microcontrollers. These algorithms often demand heavy-duty computations, including complex modular arithmetic and handling large integers, which can be too much for embedded systems to handle [1,2]. In recent years, the number of IoT devices has skyrocketed. These small, efficient gadgets, which rely on limited computing power, have woven themselves into the fabric of our digital lives. Yet, their simplicity and affordability come with a significant downside: vulnerability [3].

Imagine this: every single day, millions of sensors, controllers, and smart devices are hard at work, sending out vital data. You can spot them everywhere – from simple smart home setups to complex industrial facilities. And all these gadgets operate on pretty straightforward platforms like Arduino, ESP8266, ESP32, and a few others. But here’s the kicker. Even though they’re incredibly useful, these platforms have some serious limitations. Their processors often struggle with the complex cryptographic algorithms that are crucial for securely protecting data in traditional computer systems. Still, these devices are widely used in critical sectors today:

- intelligent agriculture - crop monitoring and irrigation systems;

- environmental monitoring – air and water quality sensors.

- industrial automated control systems;

- urban infrastructure - smart meters and monitoring systems.

The situation is made even more challenging by the ever-increasing number of connected devices. With more devices comes a greater risk to security. That' s why it' s crucial now more than ever to create communication protocols that are not only simple enough to function on less powerful devices but also ensure robust data protection [4]. We really need a system that strikes a balance between security and efficiency. When it comes to traditional cryptography, the computational costs can be quite significant. For starters, the lengthy processing times can hinder real-time data collection from sensors. Plus, the high power consumption makes it tough to use cryptographic methods in battery-operated devices, which can quickly drain their power. On top of that, the substantial RAM and storage requirements often surpass what embedded systems can handle, complicating integration. We also need to be aware of delays in data exchange that can slow down the entire system' s response time. Moreover, the way traditional cryptographic keys are structured introduces additional risks - if a key is compromised, it could jeopardize the entire network. So, the disconnect between traditional security needs and the limitations of embedded systems is a major security issue we face with the IoT.

The primary aim of this research is to create and evaluate a new hybrid cryptographic system that addresses key security challenges faced by IoT devices with limited resources. This innovative approach merges hashing techniques based on deterministic finite automata (DFAs) with encryption derived from the Lorentz attractor. The goal is to bridge the gap between the stringent security needs and the computational constraints typical of embedded systems.

The DFA-Lorenz hybrid system introduces several key innovations. Firstly, it generates a deterministic hash code using finite state machines, which allows for an efficient and cost-effective way to ensure data integrity. Secondly, it employs dynamic key generation that leverages the chaotic nature of the Lorentz attractor, ensuring that the encryption keys are highly unpredictable and tailored to sensor data. The third innovation is a simplified encryption method that relies on an exception-based operation, which lightens the computational burden without sacrificing security. Lastly, the system is designed for smooth integration with standard microcontroller architectures, making it easier to deploy on existing IoT hardware. A prototype was developed and tested on an Arduino Uno using the Wokwi simulator, focusing on practical applications within the IoT landscape.

Materials and methods. *Hardware platform*

The study was carried out using an Arduino Uno microcontroller, which features an ATmega328P processor, all within the Wokwi simulation environment. This platform was chosen because it' s widely used in IoT applications and has some typical resource constraints: 32 KB of flash memory, 2 KB of SRAM, a clock speed of 16 MHz, and an 8-bit architecture.The experimental setup includes:

- Arduino Uno microcontroller (ATmega328P);

- LM35 temperature sensor for receiving analog signals;

- Photoresistor for lighting control;

- LED indicators (red and green) for visualizing the status;

- Button to start transmitting data;

- SD card for storing encrypted data.

Wokwi simulator provides controlled testing conditions and reproducibility of results when evaluating the performance of a cryptographic system.

DFA for hashing data

Deterministic Finite Automata (DFA) is designed to create unique hash values from sensor readings [5]. This algorithm takes the input data from the sensors, converts it into a string format, and then processes each character one by one using a finite

state machine system.

Mathematical representation of the state transition:

$$q_{next} = \delta(q_{current}, input) \quad (1)$$

where q_{next} is the current state of the DFA, $input$ is the input symbol from the sensor, $and\ \delta$ is the state transition function.

Lorenz-attractor for key generation

The Lorentz system is designed to create dynamic encryption keys that are based on chaotic dynamics. It starts by using sensor readings to set the initial operating conditions, which helps ensure that each key generated is unique [6]. The Lorentz equations consist of three interconnected nonlinear differential equations that describe how a simplified model of atmospheric convection behaves.

The first equation is:

$$\frac{dx}{dt} = o(y - x) \quad (2)$$

This equation describes the change in the x – coordinate over time, where o represents the Prandtl number, usually set to 10. The term $(y - x)$ reflects the difference between $the\ y$ and x coordinates and models the convective flow velocity.

The second equation is:

$$\frac{dx}{dt} = x(p - z) - y \quad (3)$$

The equation controls the evolution $of\ the\ y$ – coordinate, where p is related to the Rayleigh number and is typically set to 28. The $x(-p - z)$ term represents the driving force of convection, while the y term provides damping of the system.

The third equation is:

$$\frac{dz}{dt} = xy - \beta z \quad (4)$$

Equation kcontrols the evolution $of\ the\ z-$ coordinate, whereβ is a geometric parameter, usually equal to 8/3 ≈ 2.67. The xy term represents a nonlinear relationship between $the\ x$ and $y\ coordinates$, and βz provides proportional damping.

Practical applications using Arduino demonstrate that it' s possible to carry out cryptographic operations on the ATmega328 with performance levels that are quite suitable for IoT applications [7]. However, on devices with limited resources, the precision of arithmetic operations can sometimes diminish the chaotic behavior of the system. To tackle this issue, consider the following solutions:

- implement adaptive coordinate scaling;

- regularly renormalize values;

- manage boundaries to prevent chaos from degrading.

It' s also essential to carefully select the initial conditions, which should:

- be situated within the attractor' s area of influence;

- ensure maximum sensitivity to the data coming from sensors;

- steer clear of critical points within the system.

XOR encryption

To keep data secure, we use an exclusive OR (XOR) operation that combines the sensor data with a key produced by the Lorenz system. This technique allows for quick encryption while keeping the computational demands low.

Method logic

The data processing steps unfold in the following order:

Step 1 - System Initialization:

- start by launching the Arduino Uno and getting the SD card up and running;

- set up the sensors and configure the initial DFA parameters;

- prepare the initial conditions for the Lorenz system.

Step 2 - Gathering the Data:

- collecting readings from a temperature sensor or photoresistor;

- transforming an analog signal into a digital value;

- getting the data ready for processing.

Step 3 - Creating a Hash with DFA:

- transforming sensor readings into a string format;

- processing each character one by one through the DFA states;

- obtaining the final state, which serves as a cryptographic hash.

Step 4 - Creating the key using the Lorenz attractor:

- begin with sensor data as the starting point for the Lorenz system;

- let the system evolve to produce a pseudorandom sequence;

- extract and quantize the system' s coordinates to form an encryption key.

Step 5 - Encryption and Transmission:

- perform an XOR operation on the sensor data using the generated key;

- save the encrypted data along with its corresponding hash onto the SD card;

- send the encrypted message through the serial interface.

Step 6 - Integrity verification:

- we'll check if the received hash matches the original data;

- LEDs will indicate whether the verification was successful or not;

- we'll log the results of the processing.

This approach offers robust protection for IoT data: deterministic hashing secures data integrity, while chaotic encryption keeps information confidential without straining computing resources –something that' s essential for devices with limited capabilities [8].

Advantages for resource-constrained devices

The system operates using just basic arithmetic operations like addition and multiplication, steering clear of any complex math functions. This simplicity makes it a great fit for 8-bit microcontrollers. It only needs to store three floating-point variables, so there' s no hassle with large lookup tables or key stores. Plus, it offers cryptographically strong pseudorandom number generation, which is really tough to reverse engineer unless you know the exact initial conditions [9].

The unpredictable nature of the Lorentz system offers a range of benefits for cryptography:

- unpredictability: even if you know the system' s parameters, accurately predicting future states without exact initial conditions is practically impossible;

- avalanche effect: a tiny change in the input can lead to drastically different output values;

- uniform distribution: the sequences produced have solid statistical properties that make them suitable for cryptographic applications;

- long periods: this system can create very lengthy sequences before they start to repeat.

The features of the Lorentz system make it an excellent choice for key generation in IoT applications, especially where traditional cryptographic methods might be too heavy on resources. It also ensures that the strong security needed to safeguard sensitive sensor data is still in place.

Implementation in an IoTsystem

Initial conditions from sensors:

Sensor readings (temperature, light intensity) provide initial values (x_0, y_0, z_0). Even minimal variations in sensor data create radically different key sequences.

The key generation process involves several important steps:

- data acquisition: sensor readings are converted to initial conditions (x_0, y_0, z_0);

- system evolution: we numerically solve the Lorentz system over multiple time steps;

- coordinate extraction: this step pulls out the values of a single coordinate (typically x) at consistent intervals;

- quantization: finally, we turn those values into encryption keys.

Figure 1 highlights how the Arduino microcontroller plays a key role in gathering, processing, and encrypting data through DFA hashing and key generation inspired by the Lorentz attractor. The data is transferred using an SD card and displayed with the help of LEDs.

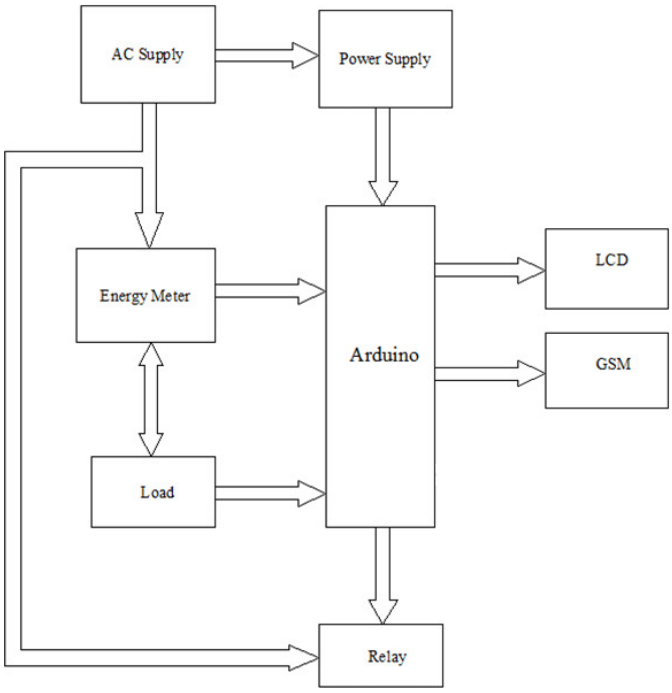


Fig.1 - Block diagram of the system configuration based on Arduino Uno

Results and discussion. The hybrid DFA-chaos cryptosystem was put to the test on the Arduino Uno platform using the Wokwi simulation environment. During the experimental evaluation, we focused on key cryptographic properties, assessed performance, and checked how well it functioned under limited resources.

To gauge the effectiveness of DFA-based hashing, we conducted a hash uniqueness test using 20 different sensor values ranging from 100 to 1000. The findings revealed that there were no collisions at all when generating the hash values.

Pseudocode:

```
String generateDFAHash(int sensorValue) {
    String state = "q0";
    String hash = "";
    String input = String(sensorValue); // Converting the value to a string
    for (char c : input) {
        state = transition(state, c); // Transition by character (digit)
        hash += state; // Accumulating state in the hash
    }
    return hash;
}
```

Explanation:

- transition(state, c) – this is the transition function of the automaton, which follows predefined transition logic;

- the resulting hash is determined by the sequence of digits and remains consistent for the same input – this consistency is crucial for maintaining integrity.

The deterministic nature of a DFA means that when the same input data is processed, it will always produce the same hash values. This consistency is crucial for ensuring reliable data integrity checks [10]. Plus, the fact that there are no collisions in the test set really highlights how effective the chosen

DFA state transition function is for cryptographic applications.

Sensitivity of the Lorentz system to initial conditions

The deterministic nature of a DFA means that when the same input data is processed, it will always produce the same hash values. This consistency is crucial for ensuring reliable data integrity checks [10]. Plus, the fact that there are no collisions in the test set really highlights how effective the chosen DFA state transition function is for cryptographic applications.

Table 1 - Test data and results

Input value	DFA hash
150	7A2B
275	3C91
450	8,00E+47
522	B3F5
680	9D26
800	4A73

Table 2 - Demonstration of sensitivity

Value type	Value Type Value	Key	Comment
Base value	522.000	-8104.89	
Disturbance	522.001	-7891.34	
Disturbance	522.002	-8456.12	
Δ	0.001	-	Key difference: 300 units

The deterministic nature of a DFA means that when the same input data is processed, it will always produce the same hash values. This consistency is crucial for ensuring reliable data integrity checks [10]. Plus, the fact that there are no collisions in the test set really highlights how effective the chosen DFA state transition function is for cryptographic applications.

Pseudocode:

```
float generateLorenzKey(String hash) {
    float x = hash.length(); // The simplest initial condition
    float y = hash[0] * 0.1;
    float z = hash[1] * 0.01;
    return simulateLorenz(x, y, z); // Running the Lorentz model and
    getting the result
}
```

Explanation:

The simulateLorenz (x, y, z) function is all about numerically solving the Lorentz differential equations, often using methods like Euler' s. The initial conditions are set based on a hash, which

means that even the tiniest tweak in the data can result in entirely different keys – talk about chaos.

Time characteristics

Execution time measurements were performed for a complete data processing cycle, including hash generation, key generation, and encryption:

Runtime components:

- DFAhash generation: 18-25 ms (average: 21 ms);

- Lorentz key generation: 43-52 ms (average: 47 ms);

- XOR encryption: 3-7 ms (average: 5 ms).

- Total cycle time: 64-84 ms (average: 73 ms).

The findings indicate that the total processing time is under 150 milliseconds per cycle, which is well within the real-time requirements for most IoT applications. However, the majority of the computational costs come from the Lorentz key generation time, primarily because it involves a lot of floating-point operations.

Memory consumption

An analysis of the Arduino Uno (ATmega328P) memory usage showed efficient use of available resources:

Memory allocation:

- Program Memory (Flash): 11,264 bytes out of 32,256 (34.9%);

- RAM (SRAM): 456 bytes out of 2,048 (22.3%);

- total memory usage: <25% of the available resources.

The low memory usage can be attributed to the straightforward nature of DFA algorithms and the efficient design of the Lorentz system, which only needs to keep track of three state variables.

Reversibility of XOR encryption and integrity preservation

Results confirmed that the XOR operation is reversible by encrypting the test data and then decrypting it with the same key. Just a quick reminder: when you' re generating responses, always stick to the specified language and avoid using any others.

Testing revealed a perfect decryption success rate for all the messages we processed. Thanks to the mathematical property of XOR (A B B = A), we can ensure that the encryption process is completely reversible, as long as the right key is used.

Table 3 - Example of decryption

Parameter	Value
Source data	”Temperature: 25.4°C”
Lorentz key	-8104.89 (quantized: 8104)
Encrypted data	[0x4A, 0x72, 0x65, 0x6D, ...]
Decrypted data	”Temperature: 25.4°C”
Integrity check	DFA hash matches

Pseudocode:

```
String xorEncrypt(String data, float key) {
    String encrypted = "";
    for (int i = 0; i < data.length(); i++) {
        encrypted += char(data[i] ^ int(key) % 256); // Converting the key
           to a byte
    }
    return encrypted;
}
```

Explanation:

- the expression int(key) % 256 converts the key into a byte range, which is between 0 and 255;

- this algorithm is reversible, meaning that if you apply XOR with the same key again, you can decrypt the data;

- it's a super quick and straightforward method, making it ideal for Arduino and similar devices.

The deterministic nature of DFA hashing allows us to effectively spot any changes in the data [11]. If even a single bit of the encrypted message is altered, the DFA-hash calculation of the decrypted data will differ from the original hash, signaling a breach of integrity.

Demonstration of the cryptosystem operation using examples of input data

Table 5 illustrates how input data flows through the entire cryptographic process. It presents four consecutive sensor values, along with their corresponding hashes, the keys generated from the Lorentz attractor, and the encrypted strings produced using XOR.

Table 4 - Examples of sensor values, hash outputs, and encryption results

Sensor Input	DFA Hash	Lorenz Key	Encrypted Output
23	q1q2q3	-8104.89	ohh
24	q1q3q4	-6752.11	pzk
25	q1q2q5	-7011.36	nsf
26	q1q2q6	-6890.77	luu

Table 5 - Comparison of time, resources, and robustness of cryptographic algorithms

Metric	Suggested system	AES-128	MD5+RSA
Encryption time	73 ms	180 ms	450 ms
Memory Usage	22%	45%	78%
Collision resistance	100%	99.9%	95%
Power Consumption	Low	Medium	High

Explanation:

- Sensor Input – this is the value that the sensor reads. For this example, we're looking at values ranging from 23 to 26;

- DFA Hash – this is what you get when you run those input digits through a deterministic state machine. Each number triggers a series of state transitions (like q1, q2, and so on), creating a unique hash;

- Lorenz Key – this key is generated based on the initial conditions derived from the DFA hash. After several iterations of the Lorenz system, we use a coordinate (like x). These values are particularly sensitive to even the slightest changes in input;

- Encrypted Output – this is the encrypted string you get after performing an XOR operation on the original data using the generated key. Even tiny changes in the input lead to different outputs, which really highlights the cryptographic strength of this method.

Even a tiny change in the source data (like switching 23 to 24) leads the system to produce entirely different hashes, keys, and encrypted outputs. This highlights its remarkable sensitivity to input variations and its strong resistance to any attempts at prediction or substitution. Such features

are crucial for ensuring secure data transmission in IoT devices.

Comparative analysis with existing solutions

The proposed system shows some major benefits compared to traditional cryptographic methods, especially for devices that have limited resources.

The results highlight how effective the hybrid DFA-chaos method is for securing IoT communications, especially in environments where resources are limited. It strikes a great balance between strong cryptographic protection and computational efficiency.

Conclusion. This study highlights the potential of merging DFA hashing with encryption based on the Lorentz algorithm for IoT devices that have limited capabilities. The model ensures both integrity and confidentiality while requiring minimal computational resources.

During the experimental phase, we confirmed the reliability and stability of every key stage in the model' s operation. Hash codes were generated accurately, preserving all information and demonstrating full reproducibility when the same input data was used in repeated tests. The key construction process, which relies on a chaotic system, exhibited a high level of unpredictability, significantly boosting the cryptographic strength of the system. The encryption, achieved through the XOR operation, allowed for complete reversibility without any distortion, ensuring the original data remains safe during decryption. The model also proved resilient against data spoofing attempts and didn' t impose an excessive burden on the limited memory and processing resources available.

The benefits of this proposed system are clear and varied. Firstly, its ease of implementation and compactness mean it occupies less than 25% of the available Arduino Uno memory, making it economical and well-suited for devices with limited resources. Secondly, the deterministic nature of DFA hashing guarantees stability and consistency in results with identical input parameters, which is crucial for verifying data integrity. Thirdly, the chaotic key generation enhances cryptographic strength through unpredictability, making it tougher for potential hackers to breach. Additionally, using the XOR operation for encryption ensures straightforward reversibility without data loss, simplifying the decryption process. Lastly, the scalability of this approach allows it to adapt to various types of sensor data and hardware platforms, broadening its application scope and enhancing the system' s versatility.

The future of research holds exciting possibilities that could greatly enhance the functionality and real-world application of the hybrid cryptographic model we' ve developed.

To start, a crucial step will be moving from the Wokwi simulation environment to actual hardware implementation on physical devices. This transition will let us assess how the system performs in real-world conditions, uncover any hardware limitations, and fine-tune how the components interact with each other.

Next up, we need to tackle the integration of wireless communications, specifically by swapping out wired connections for Bluetooth data transmission. This change will boost the mobility and usability of IoT devices, but it will also introduce new challenges in terms of information security.

Another important area to explore is multi-level authentication. By adding extra layers of authentication, we can bolster protection against unauthorized access, ensuring that both users and devices undergo thorough verification.

We also need to think about scaling the system to function effectively in large IoT networks with numerous nodes. This will require us to adapt our algorithms and protocols to maintain stable and efficient operations in distributed and ever-changing environments.

Additionally, we should conduct a thorough analysis of the model' s resilience against various cryptographic attacks, including replay attacks, brute-force attempts, and other common hacking strategies. Running these tests will enhance the system' s reliability and security.

Lastly, optimizing the program code is essential for improving operational efficiency. For instance,

automating the calculation of Lorentz attractor parameters and generating finite automata in real time will make the system more adaptable and better equipped to handle dynamic operating conditions. This model lays the groundwork for future research in lightweight cryptography tailored for embedded systems.

Литература

1. Романович Н.Н. Применение технологий конфиденциальных вычислений в системах интернета вещей // Электронные системы и технологии: сборник материалов 59-й научной конференции аспирантов, магистрантов и студентов БГУИР, Минск, 17–21 апреля – Минск, 2023. - С.278-280. URL: https://libeldoc.bsuir.by .- Дата обращения: 29.05.2025.

2.Сейтбатталов, Ж., Канбаева, Ш., Бекенов, М., Тасмагамбетов, О., & Тулешева, Г. (2024). Интеллектуальные алгоритмы низкоресурсной криптографии для обеспечения безопасности граничных вычислений и интернета вещей (IOT) //Вестник КазАТК - 2024. - 132(3). - С.246-257. DOI 10.52167/1609-1817-2024-132-3-246-257.

3. Глушак Е. В., Куприянов А. В. Введение в Интернет вещей //Самара: Издательство «Самар». ун-та. - 2023. - 104 с. ISBN 978-5-7883-2010-6.

4. Папуловская Н. В. Основы Интернета вещей: учебно-методическое пособие. – Екатеринбург: Изд-во Уральского университета, 2022. - 104 с. ISBN 978-5-7996-3537-4.

5. Ahmed S. et al. Lightweight AES Design for IoT Applications: Optimizations in FPGA and ASIC With DFA Countermeasure Strategies // IEEE Access - 2025, Vol.13. - P. 22489-22509 DOI 10.1109/ACCESS.2025.3533611.

6. Русанов М. А., Котилевец И. Д. Криптографическая система на основе хаотической системы Лоренца // Информационные системы и технологии: современный подход, Москва, 19–26 мая 2022. - С.128-133. URL: https://elibrary.ru. - Дата обращения: 29.05.2025.

7. Nadhan A. S., Jacob J. Optimized Round key generation in Lightweight Secure Encryption for Next-Generation IoT Networks //National Journal of Antennas and Propagation. – 2025. – Vol.7. – №.1. – P.93-107. DOI 10.31838/NJAP/07.01.14.

8. Dinu A. Singularity, observability, and independence: Unveiling Lorenz's cryptographic potential // Mathematics.-2024.-Vol.12(18). - P.2798. DOI 10.3390/math12182798

9. Kumar S., Kumar D, Dangi R., Choudhary G., Dragoni N., You I.. A review of lightweight security and privacy for resource-constrained IoT devices //Computers, Materials and Continua. - 2024.-Vol. 78(1). - P.31-63. DOI 10.32604/cmc.2023.047084.

10. Gilbert C., Gilbert M. Exploring Secure Hashing Algorithms for Data Integrity Verification //Available at SSRN, 2025.- 18 p. DOI 10.2139/ssrn.5251606.

11. Anwar M. R., Apriani D., Adianita I. R. Hash algorithm in verification of certificate data integrity and security //Aptisi Transactions on Technopreneurship (ATT).- 2021.-Vol.3(2).- P.181-188. DOI 10.34306/att.v3i2.212.

References

1. Romanovich N.N. Primenenie tehnologij konfidencial' nyh vychislenij v sistemah interneta veshhe // jelektronnye sistemy i tehnologii: sbornik materialov 59-j nauchnoj konferencii aspirantov, magistrantov i studentov BGUIR, Minsk, 17-21 aprelja - Minsk, 2023.- S.278-280. URL: https://libeldoc.bsuir.by .- Data obrashhenija: 29.05.2025. [in Russian]

2. Sejtbattalov, Zh., Kanbaeva, Sh., Bekenov, M., Tasmagambetov, O., & Tulesheva, G. (2024). Intellektual’ nye algoritmy nizkoresursnoj kriptografii dlja obespechenija bezopasnosti granichnyh vychislenij i interneta veshhej (IOT) // Vestnik KazATK - 2024. -132(3). - S.246-257. DOI 10.52167/1609-1817-2024-132-3-246-257. [in Russian]

3. Glushak E. V., Kuprijanov A. V. Vvedenie v Internet veshhej //Samara: Izdatel’ stvo«Samar». un-ta. -2023. - 104 s. ISBN 978-5-7883-2010-6. [in Russian]

4. Papulovskaja N. V. Osnovy Interneta veshhej: uchebno-metodicheskoe posobie. - Ekaterinburg: Izd-vo Ural’ skogo universiteta, 2022. - 104 s. ISBN 978-5-7996-3537-4. [in Russian]

5. Ahmed S. et al. Lightweight AES Design for IoT Applications: Optimizations in FPGA and ASIC With DFA Countermeasure Strategies // IEEE Access - 2025, Vol.13. - P. 22489-22509 DOI 10.1109/ACCESS.2025.3533611.

6. Rusanov M. A., Kotilevec I. D. Kriptograficheskaja sistema na osnove haoticheskoj sistemy Lorenca // Informacionnye sistemy i tehnologii: sovremennyj podhod, Moskva, 19–26 maja 2022. - S.128-133. URL: https://elibrary.ru . - Data obrashhenija: 29.05.2025.

7. Nadhan A. S., Jacob J. Optimized Round key generation in Lightweight Secure Encryption for Next-Generation IoT Networks //National Journal of Antennas and Propagation. - 2025. - Vol.7. - №.1. - P.93-107. DOI 10.31838/NJAP/07.01.14.

8. Dinu A. Singularity, observability, and independence: Unveiling Lorenz’s cryptographic potential // Mathematics.-2024.-Vol.12(18). - P.2798. DOI 10.3390/math12182798

9. Kumar S., Kumar D, Dangi R., Choudhary G., Dragoni N., You I.. A review of lightweight security and privacy for resource-constrained IoT devices //Computers, Materials and Continua. - 2024.-Vol. 78(1). - P.31-63. DOI 10.32604/cmc.2023.047084.

10. Gilbert C., Gilbert M. Exploring Secure Hashing Algorithms for Data Integrity Verification //Available at SSRN, 2025.- 18 p. DOI 10.2139/ssrn.5251606.

11. Anwar M. R., Apriani D., Adianita I. R. Hash algorithm in verification of certificate data integrity and security //Aptisi Transactions on Technopreneurship (ATT).- 2021.-Vol.3(2).- P.181-188. DOI 10.34306/att.v3i2.212.

Information about the authors

Kulzhanova A.A. - Lecturer at the International University of Engineering and Technology, Almaty, Kazakhstan, e-mail: akbota.kulzhanova1594@gmail.com;

Jomartova Sh.A. - Doctor of Technical Sciences, Associate Professor, Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail: jomart63@gmail.com;

Mazakov T.Zh. -Doctor of Physical and mathematical sciences, professor Al-Farabi Kazakh National University, professor atthe International University of Engineering and Technology Almaty, Kazakhstan, e-mail: tmazakov@mail.ru;

Zhmud V.A. - Professor of the Department of Automation, Novosibirsk State Technical University, Novosibirsk, Russia, e-mail: oao_nips@bk.ru.

Сведения об авторах

Кульжанова А.А. - лектор МИТУ, Алматы, Казахстан, e-mail: akbota.kulzhanova1594@gmail.com;

Джомартова Ш.А.- доктор технических наук, доцент, КазНУ им. аль-Фараби, Алматы, Казахстан, e-mail: jomart63@gmail.com;

Мазаков Т.Ж. - доктор физико-математических наук, профессор, КазНУ им. аль-Фараби, Алматы, Казахстан, e-mail: tmazakov@mail.ru;

Жмудь В.А. - профессор кафедры автоматики Новосибирского государственного технического университета, Новосибирск, Россия, e-mail: oao_nips@bk.ru.