ISSN 1533-9211

DOI: 10.5281/zenodo.14822860

IMPROVING THE SECURITY OF DATA TRANSFER IN IoT NETWORKS BASED ON ASYNCHRONOUS ENCRYPTION MODE

M.M. AL- HIARI

Faculty of Computer Science and Information Technology, Department of Cyber Security, Jerash University, Jerash, Jordan. E-mail: m.hiari@jpu.edu.jo. hiari5355@yahoo.com,
ORCID ID: https://orcid.org/0009-0002-2770-5417

Abstract

The paper discusses the methodology of using cellular automata with active cells to organize effective security in IoT networks. The objective of the research is to increase the reliability of the functioning of the IoT network and the resistance of such networks to cyberattacks. To solve this problem, cellular automata with active cells were used, which allow organizing an asynchronous mode of generating stream ciphers used to encrypt data transmitted from device to device. Cellular automata with active cells make it possible to implement a generator of pseudo-random bit sequences with a large number of outputs. At the outputs of the generator, different key gammas are formed, which participate in the formation of stream ciphers. The initial settings are specified by the bits of the first nine bytes of the message. If there is more than one active cell, then for each additional active cell, six more bytes are added, which define the initial settings of each additional active cell. The organization of streaming encryption of data during their transmission from device to device, as well as between gateways and the central processor is considered. The proposed algorithm for generating an asynchronous stream cipher can be easily implemented in both software and hardware. The use of cellular automata with active cells allows the implementation of encryption mode for a group of devices, the number of which can vary. Constantly changing the key range increases resistance to attacks on IoT network devices and the entire network as a whole.

Keywords: Asynchronous Stream Cipher, Cellular Automata with Active Cells, Sensor, Actuator, Gateway, Key Gamma Generator.

INTRODUCTION

Modern society is characterized by a high level of development of digital computer networks, which provide the opportunity for information interaction between people all over the world. At the same time, modern development of digital technology has made devices “smart”, which allows them to exchange data with each other without human intervention. The use of smart devices has evolved into the creation of IoT-based networks, which are rapidly being introduced into all areas of human activity. The number of devices in the IoT network is constantly growing and is determined by tens of billions of units, and the total benefit, according to MGI estimates, by 2030 will be approximately 5.5 to 10.6 trillion dollars and will reach 29 billion devices in use. At the same time, the average annual growth rate of the IoT devices market is projected to be 23.25%.

Today, there is a wide variety of technologies and devices that are developed and produced by a large number of manufacturers, which leads to the emergence of problems due to the difficulties that arise in interaction. To establish reliable information exchange, there is a need to create specialized standards and protocols that ensure the interaction of various networks

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

and IoT devices implemented using different protocols. At the same time, the security of the IoT network and the confidentiality of the transmitted data must be maintained.

There are many protocols used to organize data exchange between IoT devices. These protocols are characterized by remote device locations, limited bandwidth, and small code size. These protocols include: MQTT, MQTT-SN, NetFlow, Modbus, AMQP, Bluetooth, Cellular communication, CoAP, LoRa, LoRaWan, LWM2M, Wi-Fi, Zigbee, Z-Wave and other protocols. Each of these protocols does not satisfy the full range of requirements of different IoTs and is not universal.

At the moment, the modern organization of IoT leads to a number of problems, the solution of which is the focus of the efforts of modern specialists in this field. One of these problems is the weak counteraction to cyber threats and protection of smart devices. To counter emerging cyber threats, the following counteraction aspects are taken into account:

- Data encryption;
- Authentication and access control;
- Software updates;
- Network traffic monitoring and anomaly detection;
- Network segmentation;
- Secure operation and implementation.

Data encryption is an integral and necessary component of the security of modern data processing and transmission networks. However, research in recent years shows that 98% of IoT devices transmit data in unencrypted form [1, 2]. This has led to a large number of cyberattacks on IoT devices and IoT networks in general over the past few years. Cyberattacks have caused losses to about 90% of organizations using smart devices. Cyberattacks on IoT devices lead to privacy breaches, financial losses, and disruptions. Modern networks use fairly reliable encryption methods (AES, RSA and others). However, without updating keys and other means of implementing encryption, hacking of information transmitted over the network is possible. At the same time, updates are carried out at certain intervals, which is not always effective in IoT operations.

To ensure the security of IoT devices, it is necessary to comply with multi-factor authentication, which is determined by several heterogeneous factors. For example, passwords, biometric data and other characteristics may be used. There are also constant changes to the software. As a rule, software updates and changes are carried out automatically, the dynamics of which can be tracked by an attacker. The implementation of means for continuous monitoring of network traffic and prompt detection of anomalies in network traffic is of great importance for IoT security [3 - 5]. For this purpose, special equipment and software are used that implement machine learning technologies and artificial neural networks of various configurations. Such technologies identify threats with a high degree of accuracy and generate control responses to them in real time. Intrusion detection systems are based on continuous monitoring and analysis of network traffic, and also monitor the behavior of smart devices, which allows them to identify various deviations that indicate the presence of cyberattacks at a given moment in time.

In addition, the security of IoT devices is improved by separating devices according to certain characteristics and combining them into network segments [6, 7]. Each such segment has its own device protection algorithms that distinguish them from devices in other segments. This helps minimize the risk of hacking other devices and subnets in the overall IoT network. At the same time, data exchange between different segments or their devices has different protection rules. Taking into account all the described cybersecurity issues will ensure the smooth functioning of organizations and facilities using IoT. In this paper, a combined encryption method based on the asynchronous behavior of IoT devices is considered, which in the threat response mode changes the encryption process taking into account the constant updating and segmentation of the IoT network, which reduces the risks of hacking.

Problem Statement

The current state of the security issue for IoT devices depends, first of all, on the effective encryption of data that smart devices exchange with each other and with the central processor. The most efficient approach to real-time encryption is to use stream ciphers. The best option for encryption in IoT is to implement different stream ciphers for each segment of the IoT network, with the stream cipher constantly changing. The paper solves the problem of efficient encryption in the IoT network based on an asynchronous stream cipher, which is formed using cellular automata with active cells. In addition, the aim of this work is to generate a stream cipher that changes at times when cyberattacks and anomalies are detected in the IoT network. In this case, stream ciphers are formed differently for each IoT segment, which is implemented through the use of cellular automata with active cells (CAAC).

Relative Works

Stream ciphers are most suitable for implementing real-time mode in data transmission systems [8]. Currently, a large number of stream encryption algorithms have been developed, such as A3, A5, A8, MUGI, PIKE, RC4, SEAL. Stream ciphers are divided into synchronous and asynchronous. In this case, for the implementation of data exchange between two or more points, the asynchronous cipher is most suitable, since it allows one point to form several different stream ciphers for different receivers. Most often, stream encryption algorithms use the gamma mode, which is implemented using a pseudo-random bit sequence generator. At the output of such a generator, a pseudo-random bit sequence (key gamma) is formed, the bits of which participate in the encryption of message bits. Encryption is performed using an encryption function whose arguments are the key bits and the message bits. The most commonly used function is the XOR function. A large number of stream ciphers are based on linear feedback shift registers (LSFSR) [9, 10]. Both synchronous and asynchronous stream ciphers are implemented based on LSFSR. However, to form a key range with a long repetition period, it is necessary to use a register with a large number of digits. In addition, this approach does not provide high quality of generated ciphers in the case of using a large number of outputs to form multiple key ranges by one generator. Analysis of several ciphers generated by one

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

generator based on LSFSR allows us to determine the correlation dependence and structure of the register. Chaotic maps are often used to implement self-synchronizing stream ciphers [11 - 13]. One of the problems with such encryption is the small length of the quasi-chaotic sequence. Also, with such stream ciphers, problems arise when generating many different pseudo-random bit sequences.

There are also approaches to constructing stream ciphers based on fuzzy logic [14]. Such generators use a non-linear function, which creates problems when implementing PRNGs with multiple outputs, and therefore such PRNGs are limited in their use for encryption in group systems. The most promising approaches for implementing multiple stream ciphers are the use of cellular automata [9]. The use of elementary cellular automata does not provide a large number of outputs on one CA, which limits it for large groups of data exchange objects.

Two-dimensional cellular automata allow organizing a large number of outputs on one CA. For this purpose, heterogeneous cells or active cells are introduced into the cellular automaton. The principles of organizing such CAs are described in detail in [15]. The output in a CA with heterogeneous cells can be either outputs of homogeneous cells or outputs of heterogeneous cells. This is made possible by the constantly changing environment of the CA. In a CA with active cells, the outputs are the outputs of the active cells. The use of such CAs allows changing the number of homogeneous and active cells during their functioning, which allows them to be used for any number of objects in a data exchange group.

The work [16] presents the organization of data exchange in a group of robots based on the asynchronous principle of forming a stream cipher. The use of CAAC has been shown to be most effective in such groups. This approach is especially effective for a dynamically changing group of robots in real time. It is effective to use such CAACs for IoT systems, since the number of devices used in them is constantly increasing, and the load on the organization of data transmission and their effective protection is also increasing.

Cellular automata with active cells

Cellular automata with active cells (CAAC) are described in detail in [15]. The CA is a two-dimensional cellular environment that is organized as a classical two-dimensional CA. The most commonly used form of coverage is the orthogonal form, as well as the von Neumann and Moore neighborhoods. Moreover, active cells may be present in such a CA. Active cells are active at a given point in time and implement a local state function (LSF) that is typically different from the LSF of inactive cells and other active cells. LSF is implemented by analyzing signals coming from cells organizing the neighborhood of the active cell at each step of the time iteration. At each time step, the active cell transmits an active signal to one of the cells in the neighborhood, which at the next time step goes into the active state. The cell that was active in the previous time step executes LSF and ceases to be active. An active signal from an active cell transmits an active signal to a cell in its own neighborhood according to a given local transition function (LTF). This function is chosen such that the choice is equally probable. At each time step, the active state is transferred from cell to cell using the LTF specified. The process of functioning of such a CAAC in Fig. 1 is shown.

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

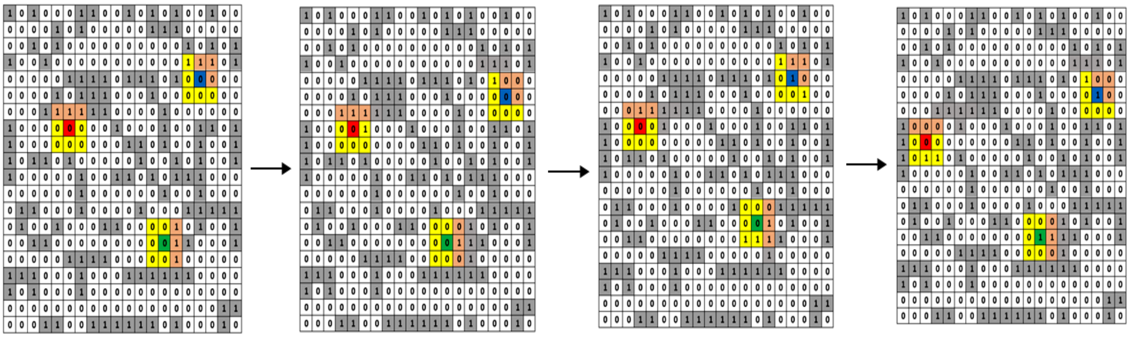

Fig 1: An example of the process of functioning of the CAAC without additional changes in various characteristics

If there are many active cells and the distribution of ones and zero states is approximately uniform, then the CAAC functions in the mode of constant cell movement. Otherwise, closed cycles of active cells' "movement" trajectories may arise. Such situations are described in [15, 17].

Several approaches are used to eliminate loops [15]. The first such approach is to constantly change the LTF or the initial arguments of the LTF. After a certain number of time steps, the set of neighborhood cells that participate in the implementation of LTF changes. For the von Neumann neighborhood are used two neighborhood cells, for the Moore neighborhood I use three neighborhood cells. An example of the functioning of such a CAAC in Fig. 2 is shown.

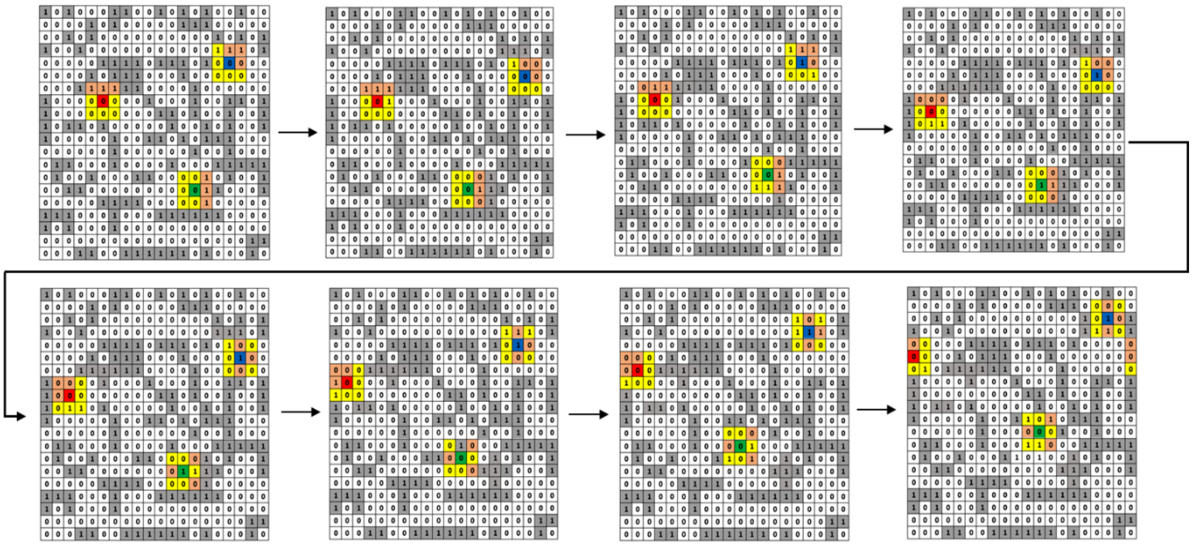

Fig 2: An example of the functioning of the CAAC with a variable structure of neighborhood cells implementing LTF

The top part of Figure 2 shows the initial neighborhood for the LTF implementation for each active cell. After three time steps, the neighborhood for LTF changes for each cell (bottom row in Fig. 2) and the "movement" of active cells changes. Both examples considered implement a

ISSN 1533-9211

DOI: 10.5281/zenodo.14822860

CAAC in which LSF is performed only by active cells. Inactive cells do not change their state, which may not always give the desired result. The solution to the emerging problems is provided by the implementation of the CAAC, in which all cells (active and inactive cells) perform LSF at each time step. An example of the operation of such a CAAC in Fig. 3 is shown.

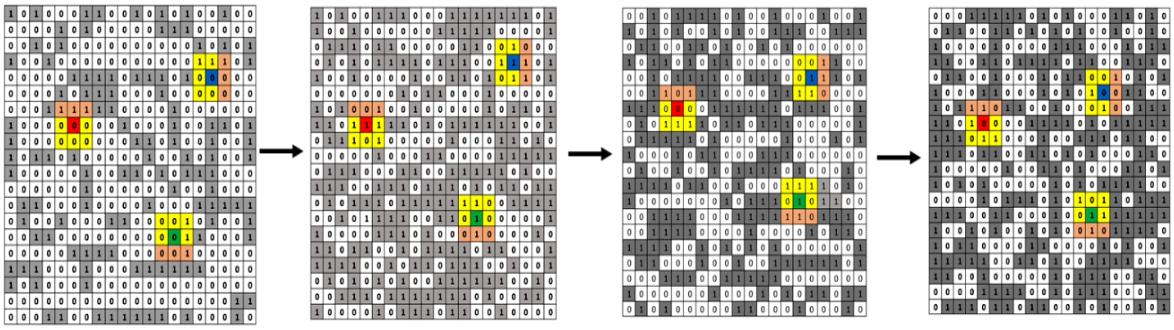

Fig 3: An example of the operation of the CAAC with a variable environment

This mode ensures reliable operation of the CAAC without cycles in the trajectories of the "movement" of active cells. In this mode, there is no need to change the LTF or the structure of its neighborhood.

The implementation of the pseudo-random number generator (PRNG), which forms the key gamma for generating the stream cipher, consists of forming pseudo-random bit sequences (PRBS) at the outputs of active cells. The quality of the generated pseudo-random bit sequences depends on the successful choice of LTF. If the PRBS formed at the output of one of the active cells does not provide high quality, then an additional active cell is used. The bits at the outputs of both active cells are mixed using the XOR function. Research has shown that this approach produces high-quality PRBSs.

The initial states of a PRNG based on CAAC are:

- Dimension of the CAAC;
- Initial states of the CAAC cells;
- Initial coordinates of active cells;
- Neighborhood shape of each cell for the implementation of LSF;
- LSF;
- LTF;
- Neighborhood shape for LTF.

For the normal functioning of a CAAC, which has more than one active cell, it is necessary that all active cells differ in at least one characteristic. Such characteristics can be the neighborhood, the LTF, and the shape of the neighborhood for the LTF. If the conditions are met, then when active cells coincide, the number of active cells is maintained, which is confirmed at the following time steps of the functioning of the CAAC.

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

The number of active CAAC cells may vary. Changes in the number of active cells can be regulated by external influences and by the CAAC itself. The CAAC itself can be organized in such a way that new active cells can be "born" when two active cells coincide (meet) in one CAAC cell. If two active cells with the same characteristics meet, they disappear. This allows the PRNG to adapt automatically to changes in various types of connections.

Asynchronous method for generating stream ciphers based on CAAC

The asynchronous method of generating stream ciphers consists of the fact that the transmitting device generates a stream cipher by superimposing a key gamma generated at the output of one of the active cells of the CAAC. This system is symmetrical, but on the receiving side the PRNG based on the CAAC is not set to the initial state in the same way as the PRNG is set on the transmitting side. Since the PRNG on the transmitting side is pre-set, it forms the first Z bits in the transmitted message, which carry information about the initial settings of the PRNG on the receiving side. The first two bytes of the message contain information about the dimension of the CAAC (N×M). The third byte contains information about the number of active cells. The fourth and fifth two bytes transmit numbers (X_{add}, Y_{add}) that help the receiving side calculate the initial coordinates of the active cell, at the output of which the key gamma for decryption is formed. The next (sixth) byte defines the shape of the neighborhood. Typically, it is a number. For example, 01 indicates the von Neumann neighborhood. The seventh byte also contains a number that specifies the LSF, and the eighth byte specifies the LTF. The ninth byte defines the neighborhood structure for the LTF. For example, if the sequence "10; 10; 1; 3; 5; 1; 3; 1; 2" is transmitted, this means that it is necessary to form a 10×10 CAAC, one active cell is formed with coordinates X_{act}=f (10,3) and Y_{act}=f (10,5). The von Neumann neighborhood is used, LSF implements the XOR function, LTF selects a neighborhood cell, which is determined by the code of the second and third cells of the neighborhood in a clockwise direction. The process of implementing asynchronous formation of a stream cipher based on CAAC in Fig. 4 is shown.

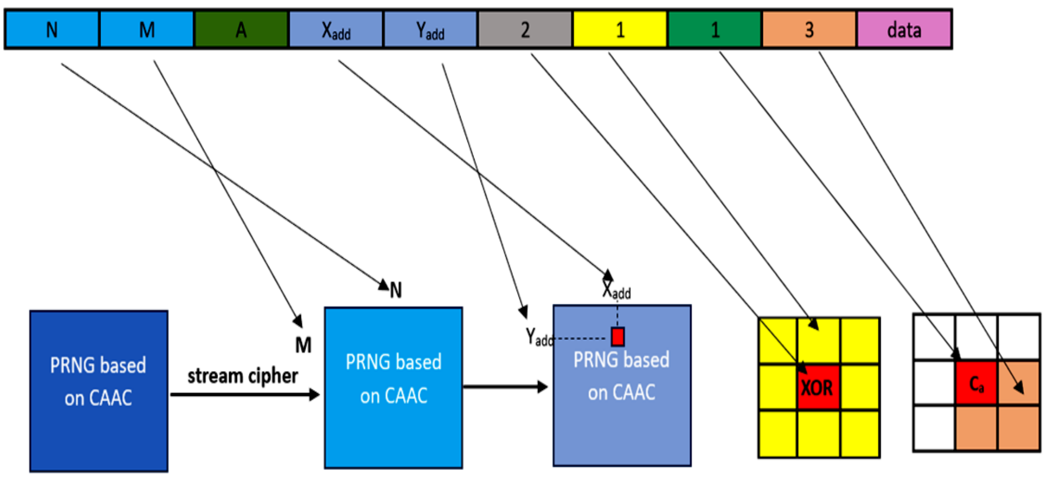

Fig 4: Implementation of asynchronous message transmission mode using stream ciphers based on CAAC

The Seybold REPORT

ISSN 1533-9211

DOI: 10.5281/zenodo.14822860

Figure 4 shows the sequence of initial settings. At the end of the ninth byte, data encrypted using a PRNG based on CAAC is received. During the passage of the first nine bytes, the PRNG manages to establish the initial states and begins to form the key gamma. In such PRNGs, the neighborhood shape is selected from a pre-formed table, and LSF and LTF are also specified in the table. If the PRNG does not have time to move to the desired initial state, then a time delay is introduced between the ninth byte and the data, allowing the PRNG to be set to the initial state.

In case of using several active cells, then for each active cell an additional six bytes are added, since the dimension of the CAAC does not change.

For more reliable operation of the PRNG, a byte (the third byte) is always added, which contains information about the number of active cells. If the third byte encodes a triple, then twelve setup bytes are added to the first nine bytes.

The use of CAAC on the receiving and transmitting sides allows for continuous stream encryption of transmitted data in both directions in asynchronous mode, which significantly increases the system's resistance to attacks on the cipher.

IoT Encryption Technology Based on Asynchronous Stream Cipher

The structure of a simple IoT contains two parts: the external part and the part for processing data and controlling smart devices. The outer part contains sensors, actuators and devices for initial analysis of data (gateways) coming from the sensors. Gateways can be processors or microcontrollers, which enables these devices to make simple decisions on controlling sensors and actuators, as well as generating and transmitting data to another part of the IoT. The external network can be organized as a complete monitoring system.

The second part of the IoT processes and stores data received from the external part of the IoT network. To transfer data between the two parts, special protocols are used, which have different organization to achieve different goals depending on the specifics of the IoT.

As stated above, data transmitted from device to device is poorly protected. This is especially true for the organization of encryption. Unauthorized access to data in the IoT network can be carried out at various points in the network. Therefore, effective encryption must be implemented in all areas of the network where transmission from device to device occurs. If we consider the two devices separately, then at the initial moment both devices contain key gamma generators, implemented on the basis of a CAAC with one active cell. The initial settings are the same for both devices. During normal operation of this section of the network, in each PRNG of both devices, active cells “move” along the CAAC field along the same trajectory and at their outputs at each moment in time form one pseudo-random bit, which for the entire data transmission session make up a pseudo-random bit sequence (key gamma). The key and data bits from the smart device form a stream cipher that is transmitted to another device. In case of detection of unauthorized influence on one of the two devices or on both devices, the coordinates of the active cell in the PRNGs of both devices are changed. To do this, the device that first detected unauthorized interference transmits a signal to the other

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

device about the need to change the coordinates of the active cell, and also transmits two numbers to calculate the new coordinates of the active cell at the next time step.

The new coordinates of the active cell are calculated according to the following systems of equations and inequalities.

$$\begin{cases} X_{new} = f(X_{old}, X_{add}) \\ Y_{new} = f(Y_{old}, Y_{add}) \end{cases},$$

$$\begin{cases} X_{new} \leq N \\ Y_{new} \leq M' \end{cases}$$

where N×M – dimension of CAAC.

The function for calculating X_{new} and Y_{new} can be a function for calculating the remainder of division by the value of the dimension of the CAAC [16].

$$\begin{cases} X_{new} = (X_{old} + X_{add}) mod N \\ Y_{new} = (Y_{old} + Y_{add}) mod M \end{cases}.$$

The arguments of the function for calculating the coordinates of new active cells can be the addresses of the transmitting device or gateway. For example, coordinates can be calculated by reducing (cutting off high-order bits) or increasing (adding low-order bits) the number of binary digits that limit the number of dimensions of the CAAC.

After calculating the new coordinates of the active cell, a new key gamma is formed at the next time step.

Detection of unauthorized intervention is carried out through constant monitoring of data transfer traffic between devices, and also analysis of the states of both devices at fixed points in time. Analysis of device states is carried out by monitoring their energy parameters, signal processing time and other characteristics.

The asynchronous encryption mode for a group of IoT devices is as follows.

There are a large number of sensors, actuators and gateways in the IoT network. All of them can interact with each other. Even if they are connected to different gateways.

As is known, connections of sensors and actuators to gateways can be either wired or wireless. Gateways are connected to the central processor both with and without wires. The most common way to use Wi-Fi is through routers. Information is transferred between devices using standard and specialized protocols. Typically, packets that contain the addresses of the transmitting and receiving devices, as well as basic data, are transmitted. An attacker may know the addresses, but the underlying data in the packets should not be known.

In order to transmit encrypted data, the initial settings of the CAAC are transmitted in the first generated and transmitted packet. In these settings, the dimension of the CAAC and the initial data for calculating the initial coordinates of the active cell are specified. If several devices are used to which data is transferred, then data is transferred to calculate the coordinates of the corresponding number of active cells. A local transition function is also specified. In this case,

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

a PRNG based on CAAC with a set of active cells is implemented. For the gateway, the organization of asynchronous encryption in Fig. 5 is shown.

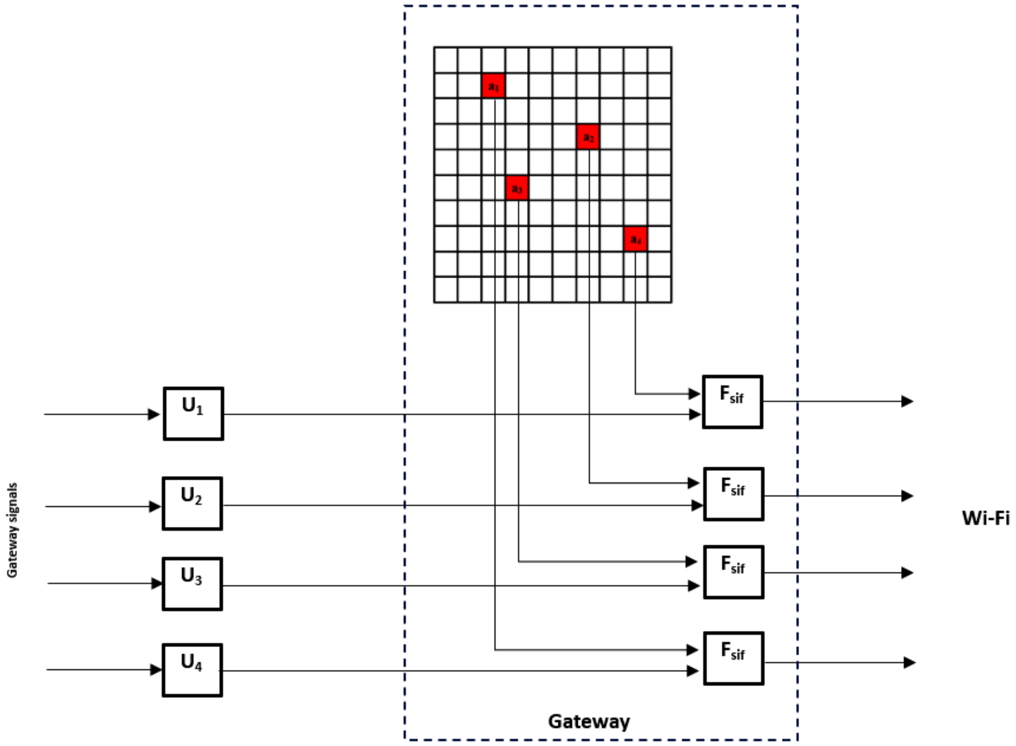

Fig 5: Scheme of encryption organization based on CAAC for a group of primary devices (sensors, actuators)

Figure 5 shows a diagram of the formation of asynchronous stream ciphers using a CAAC with four active cells. The encrypted data from each U_i device is fed to the corresponding inputs of the gateway, where, using the key gammas formed at the outputs of the corresponding active cells, four bit sequences are generated as stream ciphers through the F_{sif} encryption devices. The gateway can have four outputs, allowing all four stream ciphers to be sent to the router simultaneously. There may be one gateway output. Then all four ciphers are transmitted in time-sharing mode or in other modes. The gateway can receive incoming information using Wi-Fi or other transmission means.

A situation is possible when several end devices exchange information. In this case, these devices are connected to different gateways (Fig. 6). For example, the network has motion sensors and a video camera that turns towards the motion sensor that has recorded the movement of an object in its observation sector. In such a situation, the reacting sensor ($S_{i,j}$) forms a packet with encrypted data, arriving at one of the inputs of its gateway G_j. Here index *i* indicates the number of the sensor in the network of one gateway with number *j* in the general IoT network.

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

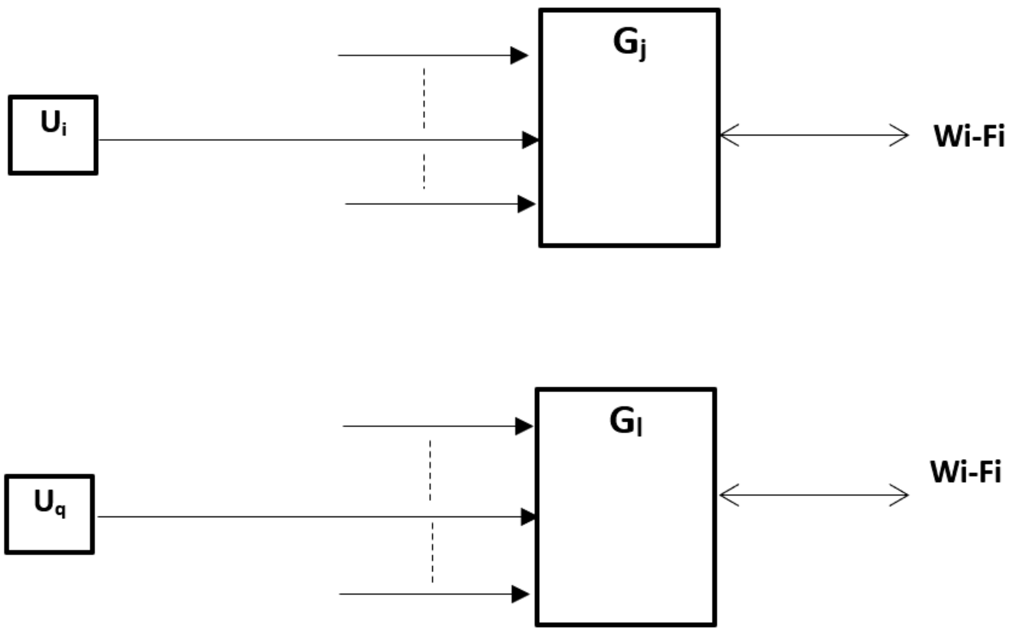

Fig 6: Layout of two devices in an IoT network exchanging data

Video camera $C_{q,l}$ has number q in the network of the l-th gateway. Then the transmitted packet from $S_{i,j}$ contains information about the address (i,j) of the transmitting sensor and the address (q,l) of the receiving actuator sensor (video camera). This package also contains information about the presence of movement in the control sector of the $S_{i,j}$ sensor. In addition, the package contains information about numbers that enable the $C_{q,l}$ device to calculate the dimension of the CAAC, the initial coordinates of the active cell and the local transition function. These installation data are received by the $C_{q,l}$ device and it forms the initial settings of the CAAC. After this, a key gamma is formed and data about the device in the direction of which the video camera should be turned is read. If the video camera rotation needs to be performed by all video cameras in the IoT, then the recipient address is not specified. In this case, all video camera devices will generate the same PRNG for the duration of the communication session.

Each gateway provides additional encryption of data coming from each smart device associated with that gateway (Figure 5). According to the addresses of the devices to which the data is sent, the gateway encrypts them using a PRNG based on the CAAC with P active cells (where P is the number of smart devices connected to the corresponding gateway). The generated codes at the gateway outputs are sent via Wi-Fi to those gateways whose addresses were indicated by the transmitting device. The transmitted information indicates the recipient's gateway number, the address of the device connected to this gateway, the initial settings of the PRNG gateway that transmits the information, as well as the coordinates of the corresponding active cell that forms the key range for decryption. The initial settings determine the dimension of the CAAC, the number of active cells and their initial coordinates, the local transition functions of all active cells and the initial state of the CAAC. After the receiving gateway has set all initial states,

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

decryption begins on the specified active cell. The decrypted information is sent to the appropriate device, where it is also decrypted using a PRNG based on a CAAC with one active cell. In fact, the information undergoes a double encryption process, which increases resistance to various types of attacks on the stream cipher. Fig. 7 shows the structure of one gateway and its external connections.

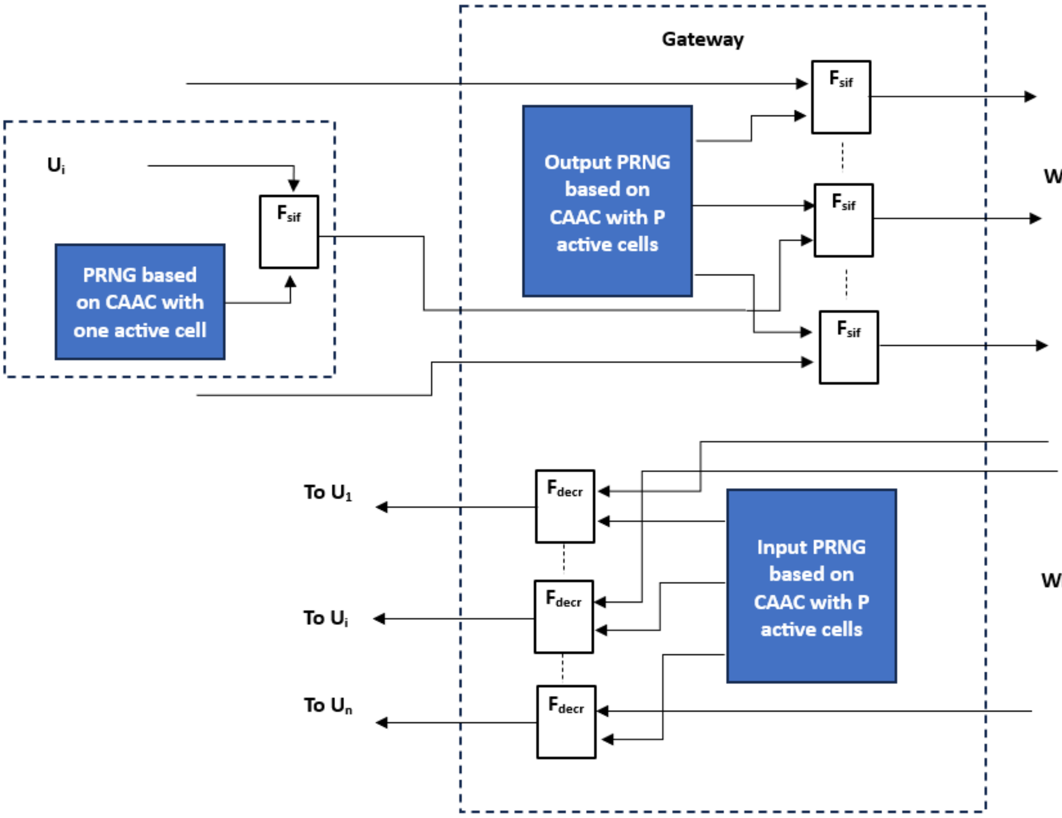

Fig 7: The structure of one gateway and its external connections

The transmission of encrypted information can be carried out in both serial and parallel modes. In this case, data reception can be carried out in sequential mode if the data comes from different gateways. If data comes from one gateway from several devices, then their reception can be carried out in parallel.

The F encryption functions can implement various operations with data and key gamma. The simplest and most common is the XOR function, since bitwise encryption is performed.

Data exchange with the central processor is carried out via Wi-Fi. In this case, a separate PRNG or a PRNG based on the CAAC with P active cells, which are used for data exchange between gateways, can be used to encrypt data. Here the highest priority remains with the central processor.

ISSN 1533-9211
DOI: 10.5281/zenodo.14822860

CONCLUSION

The paper examines and studies an approach to increasing the resistance of IoT networks and devices to cyberattacks based on asynchronous stream encryption. A cellular automaton with active cells is used as a key scale generator. The use of such a cellular automaton allows increasing the number of outputs on one pseudo-random sequence generator. This made it possible to transfer data from one device to other devices simultaneously, which are encrypted using different key gammas. This organization makes it possible to respond to unauthorized access to data and constantly change key gammas, which prevents an attacker from finding the encryption key. Research has shown that the most effective approach to implementing flexible data encryption, in the context of constantly changing number of devices in the IoT network, is the use of pseudo-random bit sequence generators with the ability to change the number of outputs without significantly reconfiguring it. Since the main parameters (dimension, local function of states of all cells) of the generator do not change, and only the parameters of active cells change, then the most suitable is to use a cellular automaton with active cells. The generator, built on the basis of a cellular automaton with active cells, is simple to implement and can be implemented both in software and hardware. In addition, a cellular automaton with active cells is easily reconfigured and, in the implementation of an asynchronous cipher, has the highest speed in the process of setting initial states. Due to the possibility of constantly changing the dimensionality of the cellular automaton at different stages of the formation of an asynchronous cipher, the reliability of such a cipher increases, which practically ensures high resistance to various types of attacks. With the help of a cellular automaton with active cells, the IoT network has the ability to quickly respond to cyber attacks, non-standard changes in network traffic and the emergence of anomalies, and promptly change the stream cipher for data encryption. Also, different stream ciphers are generated for different segments of the IoT network, which further increases the network's resistance to attacks.

In further research, the author plans to implement an asynchronous stream cipher based on two-dimensional cellular automata with heterogeneous and active cells and their combinations.

References

1) https://www.paloaltonetworks.com/cybersecurity-perspectives/expanding-iot-visibility.

2) https://www.iot-now.com/2023/10/18/137178-iot-security-survey-reveals-alarming-challenges-and-costs/

3) WeiBao Zhang 12 and Joan P. Lazaro. A Survey on Network Security Traffic Analysis and Anomaly Detection Techniques. International Journal of Emerging Technologies and Advanced Applications April, 2024, Volume1, Issue4, 1-9.

4) Faeiz Alserhani. Analysis of Encrypted Network Traffic for Enhancing Cyber-security in Dynamic Environments. Applied Artificial Intelligence 2024, Vol. 38, No. 1, P. 42, https://doi.org/10.1080/08839514.2024.2381882

5) Manish R. Joshi, Theyazn Hassn Hadi. A Review of Network Traffic Analysis and Prediction Techniques. March 2020. https://arxiv.org/pdf/1507.05722

6) Louma Chaddad, Ali Chehab and Ayman Kayssi. OPriv: Optimizing Privacy Protection for Network Traffic. J. Sens. Actuator Netw. 2021, 10, 38: 1-15

ISSN 1533-9211

DOI: 10.5281/zenodo.14822860

7) C. Zanasi et al. Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures, Ad Hoc Networks 156 (2024) 103414, 1-15.

8) A. A. Kuznetsov et al. Stream Ciphers in Modern Real-time IT Systems: Analysis, Design and Comparative Studies, Springer (November 19, 2021), 928 pages.

9) Stepan Bilan. Formation Methods, Models, and Hardware Implementation of Pseudorandom Number Generators: Emerging Research and Opportunities. - (2017). - IGI Global, USA. - P. 301.

10) Maria George and Peter Alfke. Linear Feedback Shift Registers in Virtex Devices. XAPP210 (v1.3) April 30, 2007. Application Note: Virtex Series and Virtex-II Series. https://docs.xilinx.com/v/u/en-US/xapp210.

11) Baoju Chen. Simin Yu. David D-U Li. Jinhu Lü. Cryptanalysis of Some Self-Synchronous Chaotic Stream Ciphers and Their Improved Schemes. June 2021. International Journal of Bifurcation and Chaos 31(08): 2150142

12) Chunlei Fan ID and Qun Ding. A Novel Image Encryption Scheme Based on Self-Synchronous Chaotic Stream Cipher and Wavelet Transform. Entropy 2018, 20, 445: 1-13.

13) D.N. Butusov, A.V. Tutueva, A.I. Karimov, T.I. Karimov. Models of random systems with controlled symmetry in stream encryption. Bulletin of Bryansk State Technical University. – 2019. – No. 11. – P. 46 – 54. – DOI: 10.30987/1999-8775-2019-2019-11-46-54

14) Anikin I.V., Alnajjar K. An Approach to Stream Cipher Design Using a Fuzzy Logic-Based Pseudo-Random Sequence Generator. Engineering Bulletin of the Don, No. 6 (2023). ivdon.ru/ru/magazine/archive/n6y2023/8455

15) S. M. Bilan, M. M. Bilan, R.L. Motornyuk. (2002). New Methods and Paradigms for Modeling Dynamic Processes Based on Cellular Automata, IGI-Global. 2020, P. 200.

16) Volodymyr Mokhor, Stepan Bilan, Volodymyr Samburskyi. Asynchronous Method of Generating Stream Ciphers in a Group of Robots Based on Cellular Automata with Active Cells. LNCS 14978. 16th International Conference on Cellular Automata for Research and Industry, ACRI 2024 Florence, Italy, September 9–11, 2024 Proceedings, P. 177-188

17) Stepan Bilan, Mykola Bilan, Sergii Bilan. Research of the method of pseudo-random number generation based on asynchronous cellular automata with several active cells. - MATEC Web of Conferences, - Vol. 125, 02018, (2017), pp. 1-6.