Methods of information systems protection

UDC 621.396.13+004.056.55 **doi:** https://doi.org/10.20998/2522-9052.2021.3.14

Petro Klimushin, Tetiana Solianyk, Tetiana Kolisnyk, Oleksandr Mozhaev

Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

POTENTIAL APPLICATION OF HARDWARE PROTECTED SYMMETRIC AUTHENTICATION MICROCIRCUITS TO ENSURE THE SECURITY OF INTERNET OF THINGS

Abstract. The paper objective is to determine the basic schemes and their characteristics for ensuring the security of Internet of Things nodes using symmetric authentication cryptographic microcircuits. The main results that had been obtained by using method of structural and functional design represent potentially possible options for using symmetric authentication cryptomicrocircuits to ensure the protection of Internet of Things nodes. The analysis of the presented schemes' functioning made it possible to form the following conclusions. The host-side private key storage authentication scheme provides a fast symmetric authentication process, but requires secure storage of the private key on the host side. The simplest authentication scheme without storing a secret key on the host side, which does not imply the use of a cryptographic chip on the host side, provides a fast symmetric authentication process, but has a relatively low cryptographic strength, since the interaction in the system is performed without a random component in cryptographic transformations, which assumes constant the nature of requests in the system, and, consequently, the possibility of cryptanalysis of messages. To increase the cryptographic strength of such a scheme, it is advisable to introduce into the interaction system a random component in cryptographic transformations and use additional hashing procedures with an intermediate key, which leads to the complication of the scheme due to double hashing, but significantly increases the level of information security of IoT nodes. Downloading software in the system is implemented using secret encryption and authentication keys, which are permanently stored in the secure non-volatile memory of cryptographic chips of IoT nodes. In this case, session keys for encrypting the firmware code or decrypting it are generated on the client and host side, respectively. This approach allows creating unique downloads of the original firmware code (application) by preventing cryptanalysts from obtaining its images and algorithms. The peculiarity of the scheme of exchange of symmetric session encryption keys of messages are: use of a secret key stored on the side of the host and the client; the determination of the session key is performed as a result of hashing a random number with a secret key, that is, the exchange of the session key is performed in an encrypted secure form.

Keywords: Internet of Things; cybersecurity; symmetric cryptography; crypto authentication; encryption algorithms; cryptographic chips; microcontrollers; micro accelerators; fog and cloud computing.

Introduction

With the implementation of the Internet of Things (IoT), the solution of universally important social problems is expected: improvement of people's way of life; increasing the quality of medical services; ensuring reliable public safety; improvement of management processes; job creation; creating special opportunities for business, increasing productivity and competitiveness.

The projected number of devices (things) that will be connected to the Internet is constantly growing, so the size of the address space has already changed from 32 bits in IPv4 (4.3 billion unique addresses) to 128 bits in IPv6 ($3.4x10^{38}$ unique addresses). In addition, billions of already installed sensors and devices are still not IP-compatible.

When it comes to protecting such systems, "encryption" is often equated with the term "security", although it is only one element of security. To create a secure environment, you must first identify and identify the elements connected to the network. First, you need to determine who exactly wants to connect to the network, so encryption without prior authentication allows you to protect only those who should not be online.

Network security is provided by various switching and network technologies at different levels, sets of protocols used in the network. However, they should not be considered as a means of ensuring comprehensive end-to-end security of the connection. The level of the entire system's security is determined by the level of security of the weakest link, which is the weakest link of the entire security system.

Obviously, the main problem with the introduction of the Internet of Things is the personalization of devices using unique IDs, MAC addresses, keys and certificates in order to ensure the security of their operation on the network.

Using of a certain protection scheme has particular difficulties that have to be solved by either the equipment manufacturer or the end consumer. Someone must take the cost of solving the problem of personalization and the connection process. This can be a device manufacturer, service provider or customers with their own demand to connect ready-made equipment. The complexity of these processes makes them a weak chain in network security.

The offered article is focused on research of potential possibilities of using the technical decisions of personalization in system of the Internet of things. These solutions provide additional tools to ensure the highest level of security, with minimal additional costs.

Literature analysis. Nowadays, IoT devices are present in many environments, such as agriculture, housing automation, transportation, industry, defense and public safety, energy efficiency programs, health care, where security issues can pose a risk to human

security and privacy [1-6]. Many authors who analyze these unsecured systems pay attention to the problem of IoT authentication, in particular: development of mutual authentication protocols [3], blockchain-based IoT authentication [7, 9], multifactorial and continuous IoT authentication [8], etc.

Analysis of works in this field shows that the applying of authentication methods requires further research to determine the most secure (reliable) technologies for access to IoT nodes.

Currently, great attention in the IoT system is paid to the security of stored, processed and transmitted data, protection against copying of intellectual property and digital content, as well as protection against cloning of the final IoT [2, 14]. It should be noted that receiving confirmation from a node to access it (authentication) becomes a decisive factor in ensuring the security of the Internet of Things. In addition, IoT devices bring a new paradigm to network interaction, a distinctive feature of which is almost no interaction with humans, because they are very compact, limited in computing and energy resources. It is advisable to consider the following parameters for a quality choice of tools to provide access to IoT objects: a new paradigm of IoT network interaction; the possibility of using modern microcontrollers and additional cryptographic accelerators (crypto accelerators); a variety of built-in interfaces, protocols and cryptographic algorithms with hardware support.

The aim of the article is to study approaches to use cryptographic chips to ensure secure authentication of Internet of Things nodes using symmetric cryptography procedures.

1. Technology for Internet of things security. The Internet of Things is a network consisting of interconnected physical objects (devices). These devices have built-in microcontrollers, sensors, and software that allows transferring and exchanging data between them over communication protocols. It means the Internet of Things is a network of physical objects that have built-in technologies for interaction with the local or global computing environment [13, 18].

In general, from the information and communication point of view Internet of Things means a set of the following components, which is described as a symbolic formula [17]:

IoT = Microcontrollers (sensors) + Data + Network + Services.

IoT is a network of various IoT nodes connected to the Internet, which implement different models of interaction:

1) "Thing-Thing";

2) "Thing-User";

3) "Thing-Web Object".

IoT nodes are connected by wired and wireless communication lines and interact under the control of microcontrollers, which are built into physical objects. These interconnected IoT objects have a programming and identification function that provides the IoT network with data and executes commands received from data centers or from a user who interacts with them through a computer, cell phone, car system, smart device or another platform.

IoT network is based on the concept of "vague computing» in order to obtain greater efficiency and high performance. Fog Computing – is a model of computing and storing data between terminals (nodes IoT) and traditional cloud computing centers, and any device having computing resources can be a fog node.

In this case, the centers of computing will be called "hosts″. A host can be understood as:

1) Computer system or device that has a connection to a local network or to a network and the Internet and is an integral part of this network (network node);

2) Computer server that contains a resource and provides access in a client-server format;

3) Computer program that provides services to other programs and applications. Hosts interact with IoT endpoints (called "clients") to solve their functional tasks.

Integrated security level on a route "client-host" has the following objectives: IoT node authentication; creation and exchange of session encryption keys; data encryption; secure storage of keys and data; key and certificate management; data integrity and confidentiality; software download protection and password protection from copying and reading when logging in.

It is known that security measures involve three main elements (in English literature they are denoted by the abbreviation CIA – Confidentiality, Integrity, Authenticity) [2]:

– Authenticity – the sender of the message must be identified, it means be who he claims to be;

– Integrity – the message one have sent should not change during transportation to the destination;

– Confidentiality – data stored or transmitted should be available only to authorized persons or intended facilities.

In general, main task of CIA is to ensure the security of IoT nodes, while remaining within the available resources in terms of processing power, memory and power supply.

The negative impact on the security of the IoT node is possible in four ways:

1) Over a network through the following results:

– Network scanning using web tools (such as Shodan) to detect unprotected sites;

– Poor generation of random numbers in cryptoalgorithms;

– Use of malicious software;

– Updating the firmware of the original software with code written by an attacker;

– Violation of security at the transport layer of cryptographic protocols SSL/TLS (Secure Sockets Layer/Transport Layer Security);

2) Through external ports including through the unused port and providing access to the site IoT;

3) With the help of proximity attacks one can extract information about cryptographic keys by measuring the level of radiated interference or vibration on an unprotected device;

4) By physically penetrating the device, trying to investigate the internal circuits of the device or the contents of the internal memory.

It should be remembered that the consequences of a successful attack can put at risk the network as a whole, it means, everything connected to it. Therefore, the comprehensive security of IoT nodes should protect against all these methods of attacks. There are a number of ways to support important elements of the CIA [10]:

– Authenticity – by identifying the identity of any user or any additional device that tries to connect to the node;

– Integrity – by using the message authentication code (MAC) to confirm the invariability of the message along the route;

– Confidentiality – achieved by encrypting the message.

In addition, measures can be taken to protect against contactless attacks, which are practical in nature and can be implemented throughout the system or only in a single IoT node [10]:

– Storage of keys in the protected Flash - memory without possibility of electric access to keys;

– Shielding system to limit electromagnetic radiation;

– Introduction of special schemes to prevent attempts to control the supply voltage or other signals;

– Encryption of key information in the repository so that it is not possible to open it with physical access to the built-in Flash - memory;

– Reducing the number of external ports.

It is also extremely important to protect cryptographic keys throughout their life cycle – from generation, use, storage to destruction. The tested methodology is the use of hardware security modules (HSM), which store keys in encrypted form and in secure equipment. All transactions encrypt and decrypt data that comes from outside, occurring within the device. Thus, cryptographic keys never leave a secure perimeter inside the device in which they were created.

Authentication can be performed in two main ways: symmetric and asymmetric. The main difference between them is how secret keys are used. If the same key is used on both the host side and the client side, the authentication is symmetric. If a mathematically related pair of public and secret keys is used, the authentication is asymmetric.

The Advanced Encryption Standard (AES) symmetric block encryption algorithm was adopted by the US government as a standard in a 2002 through the competition between technology institutes (128-bit block size, 128/192/256-bit key). It is an international block encryption standard ISO/IEC 18033-3:2010 and has replaced the Data Encryption Standard (DES), which no longer meets network security requirements.

The main advantages of the AES algorithm include:

1) Scattering – changing any key character or plaintext affects a large number of ciphertext characters;

2) Mixing – the transformations that are used make it difficult to obtain statistical relationships between open and closed text;

3) Byte-oriented structure, which gives good prospects for the implementation of the algorithm in future processors;

4) High speed on different platforms;

5) Not prone to many types of cryptanalytic attacks, such as: differential cryptanalysis, linear cryptanalysis, cryptanalysis based on related keys (there are no weak keys in the algorithm).

There are three ways to implement encryption according to the AES algorithm [19]:

1) Software implementation – focused on the bit rate of the platform and uses mathematical optimizations and calculated substitution tables;

2) Hardware implementation – used extensions of the command system and special instructions for modern microprocessors, which allow performing some hardware operations, which significantly speeds up the process;

3) Implementation with the use of a video card - the resources of graphics accelerators are used to perform parallel encryption or decryption.

Asymmetric cryptography based on the cryptographic algorithm with RSA (Rivest Shamir Adleman) public key, ECC (Elliptic Curve Cryptography) is a powerful tool, but it requires much more processing power for encryption and decryption compared to symmetric algorithms DES, AES block encryption. Therefore, the use of algorithms with public key was found as not very effective to transfer each encrypted data packet through the Internet. As a result, using of asymmetric cryptography was limited to the exchange and calculation of symmetric session keys, which are used to encrypt and decrypt data streams in the Internet [15].

The security of the RSA cryptosystem is based on the problem of integer factorization, and ECCs are based on elliptic curves over finite fields. To ensure the required level of cryptocurrency, the ECC system uses keys of much shorter length than the RSA system. This emphasizes the effectiveness of their use in the field of Internet security as devices with very limited resources.

The basis for building cryptographic systems are random sequences, which can be obtained using random number generators. The cryptographic stability of the data encryption tool significantly depends on the quality of randomness of these sequences, which are used to generate cryptographic keys, key information and system parameters of the cryptographic system.

Random number generators as well as sequences obtained with their use can be divided into two classes:

1) Random (physical, hardware) – random sequences with high-quality randomness;

2) Pseudo-random (algorithmic, software) – pseudo-random sequences that have a period of repetition.

The creation of effective random sequence generators, as well as the evaluation of their statistical properties is one of the important and separate areas of cryptography research [12].

Today, hardware information security is becoming more widely used, despite the fact that they are much more expensive compared to similar software. The main advantages of such security include [16]:

1) The presence of a hardware built-in random number generator, which provides the generation of more reliable keys of crypt algorithms;

2) Cryptographic keys are stored and cryptographic procedures are performed on the data in a secure cryptographic chip, not in the computer's RAM;

3) The ability to distinguish between the processes of protection of information from unauthorized access and access to a computer;

4) Use a specialized microprocessor function to perform cryptographic transformations, which reduces the load on the computer's CPU;

5) Guaranteeing the integrity of the cryptographic transformation algorithms implementation, because cryptographic software by modifying the program code can change the structure of the cryptoalgorithm;

6) Increase the speed of data encryption / decryption.

Due to certain advantages of hardware protection of information and the requirements of the development of security technology for the use of the Internet of Things, there is a need to create cryptographic accelerators (cryptoaccelerators) – specialized modules in general purpose microprocessor kits.

A feature of cryptoaccelerators is that they operate separately from the core. Thus, the microcontroller core can use its resources to perform other tasks. In addition, cryptoaccelerators, unlike standard interfaces (SPI, I2C, UART, USB, etc.), have specific implementations depending on the model and family of microcontrollers, which complicates the optimal choice for a particular task [18]. The use of cryptographic accelerators increases the speed of encryption by the AES algorithm compared to its software implementation in 8/16-bit microcontrollers by 10-20 times, and in 32-bit microcontrollers – up to 150 times. At the same time, the speed of hash algorithms SHA-256 with hardware implementation in 32-bit microcontrollers increases 100 times, and for hash algorithms HMAC – up to 500 times [11].

It should be noted that the trend towards a comprehensive security solution is observed in 32-bit microcontrollers. They also provide the ability to securely generate and store keys, support electronic certificates and signatures, secure download of firmware (applications).

Let us analyze the potential applications of cryptographic chips of the CryptoAuthentication family for the security of the Internet of Things using symmetric cryptography procedures. Such procedures should include:

– Symmetric authentication with and without storage of the secret key on the host side;

– Symmetric authentication according to the scheme with an intermediate key and without storing the secret key on the host side;

– Creation and exchange of session keys of data encryption;

– Secure storage and transmission of data using symmetric encryption;

– Password protection from copying and reading when logging in;

– Protection of downloading the original firmware code (application) using secret encryption and authentication keys.

2. Symmetric authentication with storage of the secret key on the host side. For symmetric authentication tasks with mutual authorization between the host and the device, both sides use the same secret key encryption. This secret key is pre-programmed into the protected memory of cryptographic chips of symmetric authentication on the host and client side (Fig. 1). The host sends the client a random request number from the Random number generator (RNG), which is created by the built-in random number generator of the chip.

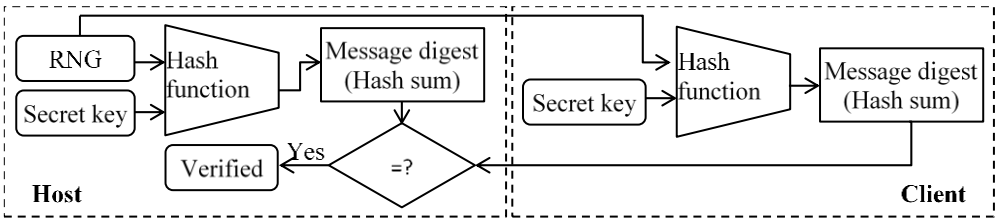

Fig. 1. Symmetric authentication with storage of the secret key on the host side

Upon receiving this request, the client submits it to the input of the hash algorithm (Hash function) together with the secret key, which is stored in the protected memory of the chip. The result is a message digest (hash sum) of the message, which is also called the message authentication code, or MAC. The message is forwarded to the host, where it is compared to a message received in the same way on the host side. If the client and host digests match, the client is considered verified, i.e. the client has been authenticated.

The main features of such an authentication scheme:

1) Fast process of symmetric authentication;

2) Cryptographic chips of symmetric authentication on both sides provide secure storage of secret keys. The disadvantage of this scheme is the storage of the secret key on the host side [14].

3. Symmetric authentication without storing a secret key on the host side. Symmetric authentication can be performed without storing secret keys on the host side, ie without a cryptographic chip on the host side. This approach is called "fixed request". In this case, the host no longer uses random numbers. Instead, certain pairs of pre-calculated numbers are used (query values and corresponding responses that are written to the non-volatile memory of the microcontroller on the host side).

However, this method has a relatively low cryptographic resistance, as the request has no random component and the cryptanalyst can use a logic analyzer on the information exchange bus to intercept messages and reveal secrets.

Advantages of the approach: fast process of symmetric authentication; no cryptographic chip on the host side is required, the host performs the authentication function under the control of the microcontroller; it is not necessary to securely store secret keys on the host side. The disadvantage of this scheme is low cryptocurrency.

4. Symmetric authentication according to the scheme with an intermediate key and without storing the secret key on the host side. This approach is implemented by an additional hashing procedure with an intermediate key (Fig. 2).

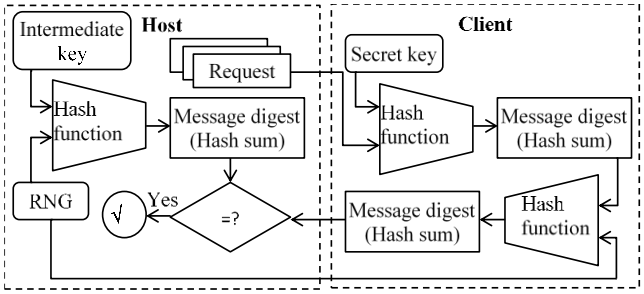

Fig. 2. Symmetric authentication according to the scheme with an intermediate key and without storing the secret key on the host side

As in the previous case, the values of the requests and pre-calculated and related responses are pre-written to the non-volatile memory of the host microcontroller. But now these answers become Intermediate keys. The process begins with a request (Request), which the host sends to the client. This request is hashed (Hash function) in the cryptographic chip on the client side with a secret key (Secret key). The received digest (Message digest), which will be an intermediate key, then hashes again, but with a random number (RNG), for example, a combination of current values of date and time, which is generated by the host and sent to the client. The re-received digest (Message digest) is a client response that is sent to the host and compared there with the digest received by the host by the same calculations with the same data. If the digests coincide, the client is considered verified. Note that in this scheme, the client's response will be new each time, because the random number (RNG) is different each time. This significantly increases the level of information security of the system - now the use of a logic analyzer will not give the cryptanalyst the desired result. Thus, the host request has a random component and as a result the client forms a response that is not repeated. Therefore, the level of cryptocurrency is significantly increased, and the storage of the secret key is on the client side and does not require secure storage of secret keys on the host side.

5. Protection of downloading the original firmware code (application) using secret encryption and authentication keys. This approach uses the encryption of the original firmware code (application) and the generation of authentication code on the host side and includes operations of decryption, authentication and download of source code on the client side (Fig. 3).

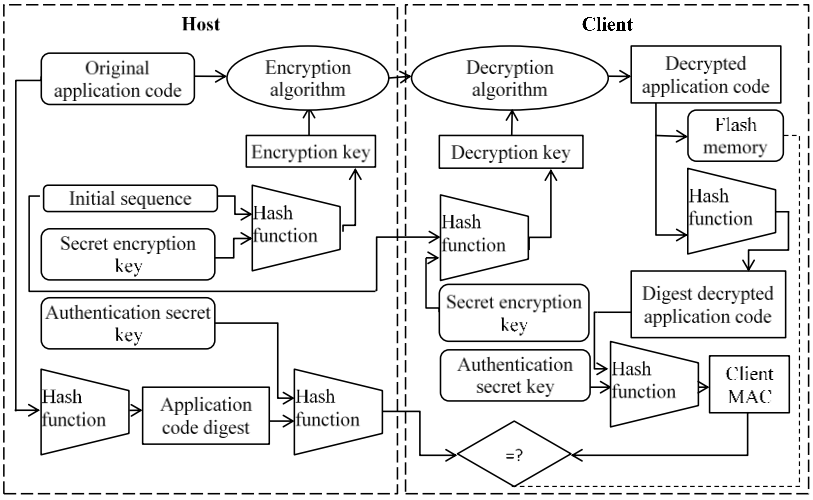

Fig. 3. Protection of firmware code download (application) with secret keys

The encryption process on the host side is performed by hashing (Hash function) of some initial sequence (Initial sequence) together with the secret encryption key (Secret encryption key). The resulting digest of the initial sequence will be a session encryption key (Encryption key) for the symmetric encryption algorithm. The result of this conversion will be an encrypted firmware (application) code, which together with the initial sequence will be transmitted to the client side so that it can be decrypted.

Obtaining the MAC is done by hashing (Hash function) of the original application code (Original application code). As a result, the digest of the application code (Application code digest) is determined, which is then hashed together with the secret authentication key (Authentication secret key). The resulting digest is a MAC and is sent to the client.

On the client side, the received encrypted code is first decrypted, and then its authentication is performed to confirm the authenticity. To decrypt the code, the initial sequence (Hash function) is hashed together with the secret encryption key, which is also stored on the client side. The result of hashing is a decryption key, which, of course, coincides with the encryption key.

This key restores the Decrypted application code. The process of obtaining a MAC on the client side follows the same procedure as on the host side. If the client-side MAC (Client MAC) matches the MAC obtained from the host, the resulting source code (Decrypted application code) is considered valid and can be loaded into Flash memory for execution.

Thus, this approach allows you to create unique downloads of the original firmware code (application) using secret encryption and authentication keys. In this case, the session keys for encrypting the firmware code or decrypting it are formed on the client and host side, respectively. This approach allows creating unique downloads of the original firmware code (application) by preventing cryptanalysts from obtaining its images and algorithms [14].

6. Creating and exchanging session encryption data keys. A data encryption session key is a parameter for cryptographically converting data into a single communication session. The session key is limited in time and is typically used to encrypt and decrypt within a single communication session. This is due to the fact that encrypting with the same key several ciphertexts increases the probability of compromising the key, i.e. increases the possibility of cryptanalysis of data in the system.

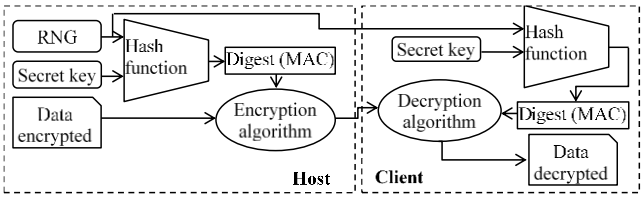

Fig. 4. Exchange a symmetric message encryption session key

Session key exchange begins with random number generation using the built-in RNG random number generator. This number, along with the Secret key stored in the host's secure memory, is hashed using a hash function to produce a digest (Digest MAC). The first or second 16 bytes (128 bits) are taken from the received digest, which become the session key of AES encryption. This key can be used to perform the procedure of encrypting the required source data (Data encrypted) according to the AES-128 algorithm.

The encrypted data and the random number are then forwarded to the client side and used to recover the session key using the same secret key stored in the client's secure memory.

Since the secret keys are the same on both sides, the hash operation with the same random number on each side will result in the same digests. Also on the client side, the first or second 16 bytes of the received digest are taken as a session key for the AES-128 algorithm. This key is used to decrypt the received encrypted data by executing the decryption algorithm AES-128.

Thus, the exchange of a symmetric session key encryption of messages is performed on the basis of generating a random number on the host side, using a secret key stored on the host side and the client. The session key is defined as the result of hashing a random number with a secret key and extracting a certain part from the digest obtained by the hashing result. To ensure security, the transmission of the session key to the client side is performed as a result of encrypting some data with the session key. The allocation of the session key on the client side is carried out according to the same scheme as on the host side - based on a random number and a secret key. The peculiarity of this methodology is that the session key is changed for each session, which increases the level of security. This methodology can be implemented using cryptographic chips of symmetric authentication ATSHA204A [14].

7. Secure storage and transmission of data using symmetric encryption. In the operation of distributed data collection and processing systems connected to data banks on remote servers or cloud storage, secure from the point of view of information security data exchange is very important. It involves the use of a proven and reliable encryption algorithm and encryption keys, which must be stored in a secure place. The capabilities of cryptographic devices in this sense look better compared to traditional software-based solutions.

Let the remote client obtain confidential information from the host. This information must be encrypted beforehand. However, for further decryption on the side of the remote client it is necessary to have a decryption key. There is a task of sending it on the same communication channels without the risk of disclosure. CryptoAuthentication chips can help solve this problem.

Symmetric encryption is used for this purpose (Fig. 5). The information intended to be sent to the client is encrypted on the server side by a fast symmetric AES algorithm. The session encryption key for this operation is formed by hashing some initial sequence (random number) and a secret key, which is securely stored on the encryption side. Encrypted data (Encrypted file) and Initial sequence are stored on the server, and the session key is destroyed.

The initial sequence and encrypted data (file) are then sent to the client. The encrypted file is received by the client's system microcontroller, and the initial sequence is received by the CryptoAuthentication chip, which stores the same secret key as the remote host. This initial sequence is hashed in the secure hardware environment of the cryptographic device with the secret key. The result of the hashing will be a session key, with which the decrypted file will be decrypted according to the AES symmetric encryption algorithm.

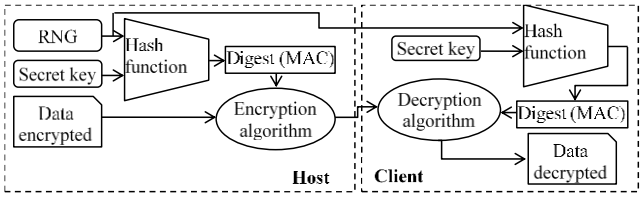

Fig. 5. Secure data storage using symmetric encryption

Note that the provided scheme of secure storage and transmission of data using symmetric encryption is

very simple. Another cryptographic chip from the CryptoAuthentication family is added to the system, which can also be used to store other small amounts of sensitive data.

8. Password protection from copying and reading when logging in. Sometimes it is necessary to compare the password entered into the system with the reference value stored in it, so that the password can not be copied or read. Since the firmware of the standard microcontroller can be hacked, in such cases it is recommended to store both the password in a secure hardware environment and to compare the value of the entered password with the reference, i.e. to perform authentication.

CryptoAuthentication family cryptographic chips can be used for this purpose. In the example in Fig. 6, the cryptographic device sends a request to the microcontroller in the form of a random number at the time when the password begins to be entered into the system from the outside. This random number is generated (RNG) and then hashed (Hash function) in the cryptographic chip with the reference value of the password, which is stored in its protected independent memory.

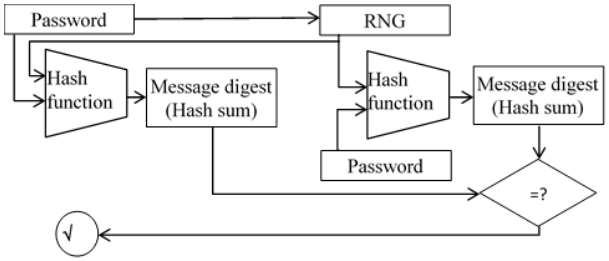

Fig. 6. Password protection from copying and reading when logging in

After entering the password into the system, the microcontroller hashes the obtained value with a random number sent to it in advance from the cryptographic chip. The received second digest is sent from the microcontroller to the cryptographic device, which compares it with the already calculated first digest. When they match, the entered password is considered valid, the cryptographic chip signals the microcontroller, thus allowing the system to work.

A feature of the scheme is the secure storage of system passwords, inexpensive, reliable and easy to implement solution.

Conclusions. Conclusions. Thus, the level of complex security of the Internet of Things on the route "client-host" solves the following tasks: symmetric or asymmetric authentication of interacting system devices; creation and exchange of session encryption keys; storage of encryption keys; data integrity; data confidentiality; software and data download protection.

Asymmetric cryptography based on RSA, ECC public key cryptographic algorithms is a powerful tool, but it requires much more computing power to encrypt and decrypt data compared to symmetric DES, AES block encryption algorithms.

Analysis of basic symmetric authentication schemes is characterized by the fact that:

1) The authentication scheme with storage of the secret key on the host side provides a fast process of symmetric authentication, but requires secure storage of the secret key on the host side;

2) Authentication scheme without storing a secret key on the host side, i.e. without a cryptographic chip on the host side, provides a fast process of symmetric authentication, but has a relatively low cryptographic stability, as the interaction in the system is performed without a random component in cryptographic transformations;

3) To increase the cryptographic stability of the scheme of paragraph 2, it is advisable to introduce into the system of interaction of a random component in cryptographic transformations and the use of additional hashing procedures with an intermediate key.

Downloading the original firmware to the system is done using secret encryption and authentication keys, which are stored permanently in the protected non-volatile memory of cryptographic chips on the client and host side. In this case, the session keys for encrypting the firmware code or decrypting it are formed on the client and host side, respectively. This approach allows you to create unique downloads of the original firmware code (application) by preventing cryptanalysts from receiving its images and algorithms.

The peculiarity of the scheme of exchanging symmetric session keys encryption of messages is:

1) Execution based on the generation of a random number on the host side and the use of a secret key stored on the host and client side;

2) The session key is defined as the result of hashing a random number with a secret key and the selection of a certain part of the digest obtained by the hashing result;

3) To ensure the security of the transmission of the session key to the client is performed by encrypting some data with the session key;

4) The allocation of the session key on the client side is carried out on the same as on the host side based on a random number and a secret key.

In the operation of distributed data collection and processing systems connected to data banks on remote hosts, secure from the point of view of information security data exchange involves the use of a proven and reliable encryption algorithm and encryption keys, which must be stored in a secure place.

The capabilities of cryptographic devices in this sense look better compared to traditional software-based solutions.

Fast symmetric AES encryption is used for secure data storage. In this case, the session encryption key for this operation is formed by hashing some initial sequence (random number) and a secret key, which is securely and securely stored on the encryption side. Transmission of confidential data over the network is possible in an encrypted file. The encrypted file on the receiving side is received by the client's system microcontroller, and the initial sequence is received by a cryptographic chip, in the memory of which the same secret key is stored as on the remote host. This initial sequence is hashed in the secure hardware environment

of the cryptographic device with the secret key. The result of the hashing will be a session key, which will be used to decrypt the file using the symmetric AES encryption algorithm.

This scheme of secure storage and transmission of data is very simple. Another cryptographic chip from the CryptoAuthentication family is added to the system, which can also be used to store other small amounts of sensitive data.

To securely store system passwords, you must store them in a secure non-volatile memory of the cryptographic chip, as the firmware of a standard microcontroller can be cracked. In this case, the comparison of passwords entered into the microcontroller with the reference stored in the memory of the cryptographic chip, it is advisable to perform in a secure environment, as well as comparing the results of hashing these passwords with a certain random number.

References

1. Burg, A., Chattopadhyay, A. and Lam, K. (2018), "Wireless Communication and Security Issues for Cyber-Physical Systems and the Internet-of-Things", *Proc. IEEE*, no. 106, pp. 38–60.
2. *CryptoAuthentication™ Family*. URL: https://www.microchip.com/en-us/products/security-ics/cryptoauthentication-family.
3. David J., Wu, Ankur, Taly, Asim, Shankar and Dan, Boneh Privacy, (2017), "Discovery, and Authentication for the Internet of Things", *Computer Science. Cryptography and Security*, URL: https://arxiv.org/abs/1604.06959.
4. Eustace, Asanghanwa and Ronald, Ih. (2020), *Security for Intelligent, Connected IoT Edge Nodes. White Paper, Security ICs, CryptoAuthentication Marketing*, URL: https://www.microchip.com/content/dam/mchp/documents/OTH/ProductDocuments/SupportingCollateral/Atmel-8994-Security-for-Intelligent-Connected-IoT-Edge-Nodes_Whitepaper.pdf.
5. Froiz-Míguez, I., Fernández-Caramés, T.M., Fraga-Lamas, P. and Castedo L. (2018), "Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes", *Sensors*, no. 18(8), 42 p., DOI: https://doi.org/10.3390/s18082660.
6. Lo'ai, Tawalbeh, Fadi, Muheidat, Mais, Tawalbeh and Muhannadm Quwaider (2020), "IoT Privacy and Security: Challenges and Solutions", *Appl. Sci.*, no. 10(12), 17 p., DOI: https://doi.org/10.3390/app10124102.
7. Mostafa, Yavari, Masoumeh, Safkhani, Saru, Kumari, Sachin, Kumar and Chien-Ming, Chen (2020), "An Improved Blockchain-Based Authentication Protocol for IoT Network Management", *Security and Communication Networks*, article ID 8836214, 16 p., DOI: https://doi.org/10.1155/2020/8836214.
8. Rainer, Falk and Steffen, Fries (2016), "Advanced Device Authentication: Bringing Multi-Factor Authentication and Continuous Authentication to the Internet of Things", *CYBER 2016: The First International Conference on Cyber-Technologies and Cyber-Systems*, Germany, pp. 69–74.
9. Zongqing, Tian, Biwei, Yan, Qiang, Guo, Jianyun, H. and Qingyu Du (2020), "Feasibility of Identity Authentication for IoT Based on Blockchain", Procedia Computer Science, vol. 174, pp. 328–332, DOI: https://doi.org/10.1016/j.procs.2020.06.094.
10. Asangkhanva. Yu, Yi. R. and Syrov, A. (2019), "Improving the security level of the edge nodes of the IoT using microchip ATECC608A chips", *Electronics NTB*, no. 7 (00188), p. 60-64, DOI: https://doi.org/10.22184/1992-4178.2019.188.7.60.64.
11. Gnusov, Y. B., Klimushin, P. S., Kolisnyk, T. P. and Mozhayev, M. O. (2020), "Analysis of microcontroller modeling systems with additional modules of cryptographic information protection", *Bulletin of the National Technical University "KhPI"*, no. 1 (3), pp. 79–84, DOI: https://doi.org/10.20998/2079-0023.2020.01.14.
12. Gorbenko, Y. I., Grinenko, T. O. and Narezhniy, O. P. (2015), "Analysis of statistical properties of the hardware generator of random sequences", *Collection of scientific works of Kharkiv University of the Air Force*, no. 4 (45), pp. 74–77.
13. Gorbovsky, A. I. and Voitovich, O. P. (2020), *Internet safety research*, URL: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17277/2805.pdf?sequence=3 pdf.
14. Krivchenko, I. (2015), "Hardware-protected chips of the CryptoAuthentication family: potential applications of ATSHA204A", *Components and technologies*. no. 10, pp. 60–65.
15. Crinon, G. (2018), *Internet of Things security: existing problems and their solutions*, URL: https://controleng.ru/wp-content/uploads/In_08.pdf (accessed 21.05.2021).
16. Petrenko, A. B., Shmatok, O. S. and Ageenko, E. A. (2014), "Analysis of time attacks on the hardware encoder of the personal means of cryptographic protection of information SHIPKA", *Science-intensive technologies*, no. 2 (22), pp. 187–191.
17. Puleko, I. V. and Chumakevich, V. O. (2021), *IoT sensors with time representation of measuring information*, URL: https://conf.ztu.edu.ua/wp-content/uploads/2019/06/44.pdf.
18. Sovin, Y. R., Nakonechny, Y. M., Opirsky, I. R. and Stakhiv, M. Yu. (2018), "Analysis of hardware support for cryptography in IoT devices", *Ukrainian Scientific Journal of Information Security*, vol. 24, issue 1, pp. 36–48.
19. Shlykov, D. I. (2018), "About the fast implementation of the AES cipher in the Sdicrypt library", *Information systems*. no. 3 (53), pp. 34–40.

Received (Надійшла) 11.06.2021
Accepted for publication (Прийнята до друку) 25.08.2021

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Клімушин Петро Сергійович – кандидат технічних наук, доцент, доцент кафедри протидії кіберзлочинності Харківського Національного університету внутрішніх справ, Харків, Україна;
Petro Klimushin – Candidate of technical science, associate professor, associate professor of Countering Cybercrime Department Kharkiv National University of Internal Affairs, Kharkiv, Ukraine;
e-mail: klimushyn@ukr.net; ORCID: https://orcid.org/0000-0002-1020-9399.

Соляник Тетяна Миколаївна – кандидат технічних наук, доцент, доцент кафедри протидії кіберзлочинності Харківського Національного університету внутрішніх справ, Харків, Україна;
Tetiana Solianyk – Candidate of technical science, associate professor, associate professor of Countering Cybercrime Department Kharkiv National University of Internal Affairs, Kharkiv, Ukraine;
e-mail: t.solianyk@khai.edu; ORCID: https://orcid.org/0000-0003-3695-0019.

Колісник Тетяна Петрівна– кандидат педагогічних наук, доцент, доцент кафедри протидії кіберзлочинності Харківського Національного університету внутрішніх справ, Харків, Україна;
Tetiana Kolisnyk – Candidate of pedagogical science, associate professor, associate professor of Countering Cybercrime Department Kharkiv National University of Internal Affairs, Kharkiv, Ukraine;
e-mail: ktp201505@gmail.com; ORCID: https://orcid.org/0000-0002-7442-8136.

Можаєв Олександр Олександрович –доктор технічних наук, професор, професор кафедри інформаційних технологій Харківського Національного університету внутрішніх справ, Харків, Україна;
Oleksandr Mozhaev – Doctor of technical science, professor, professor of information technology Department Kharkiv National University of Internal Affairs, Kharkiv, Ukraine;
e-mail: mozhaev1957@gmail.com; ORCID: https://orcid.org/0000-0002-1412-2696.

Потенційне застосування апаратно захищених мікросхем симетричної автентифікації для забезпечення безпеки інтернет речей

П. С. Клімушин, Т. М. Соляник, Т. П. Колісник, О. О. Можаєв

Анотація. Метою роботи є визначення основних схем та їх характеристик для убезпечення вузлів інтернет речей з використанням криптографічних мікросхем симетричної автентифікації. **Результатами** роботи, що були отримані за допомогою **методу** структурно-функціонального проектування, є потенційно можливі варіанти застосування кріптомікросхем симетричної автентифікації для забезпечення захисту вузлів інтернет речей. Аналіз функціонування представлених схем дозволив сформувати наступні **висновки**. Схема автентифікації зі зберіганням таємного ключа на стороні хоста забезпечує швидкий процес симетричної автентифікації, але вимагає захищеного зберігання таємного ключа на стороні хоста. Найбільш проста схема автентифікації без зберігання таємного ключа на стороні хоста, яка не передбачає застосування криптографічної мікросхеми на стороні хоста, також забезпечує швидкий процес симетричної автентифікації, але має відносно невисоку криптостійкість, так як взаємодія в системі виконується без випадкової складової в криптографічних перетвореннях, що зумовлює незмінний характер запитів в системі, а отже можливість криптоаналізу повідомлень. Для підвищення криптостійкості цієї схеми доцільне введення в систему випадкової складової в криптографічних перетвореннях та використання додаткових процедур хешування з проміжним ключем, що приводить к ускладненню схеми за рахунок подвійного хешування, але значно підвищує рівень інформаційної безпеки вузлів IoT. Завантаження програмного забезпечення в системі реалізується за допомогою таємних ключів шифрування та автентифікації, які зберігаються постійно в захищеної енергонезалежної пам'яті криптографічних мікросхем вузлів IoT. При цьому сеансові ключі шифрування коду мікропрограми або її розшифрування формуються відповідно на стороні клієнта і хоста. Цей підхід дозволяє створювати унікальні завантаження оригіналу коду мікропрограм (додатку) шляхом недопущення отримання криптоаналітиками її образів і алгоритмів. Особливістю схеми обміну симетричними сеансовими ключами шифрування повідомлень є: використання таємного ключа, що зберігається на стороні хоста і клієнта; визначення сеансового ключа виконується як результат хешування випадкового числа з таємним ключем, тобто обмін сеансовим ключем виконується в зашифрованому безпечному вигляді.

Ключові слова: інтернет речей; кібербезпека; симетрична криптографія; криптоавтентифікація; алгоритми шифрування; криптографічні мікросхеми; мікроконтролери; мікроакселератори; туманні та хмарні обчислення.

Потенциальное применение аппаратно-защищенных микросхем симметричной аутентификации для обеспечения безопасности интернет вещей

П. С. Климушин, Т. Н. Соляник, Т. П. Колесник, А. А. Можаев

Аннотация. Целью работы является, определение основных схем и их характеристик для обеспечения безопасности узлов интернет вещей с использованием криптографических микросхем симметричной аутентификации. **Результаты** работы, полученные **методом** структурно-функционального проектирования, представляют собой потенциально возможные варианты применения криптомикросхем симметричной аутентификации для обеспечения защиты узлов интернет вещей. Анализ функционирования представленных схем позволил сформировать следующие **выводы**. Схема аутентификации с хранением секретного ключа на стороне хоста обеспечивает быстрый процесс симметричной аутентификации, но требует защищенного хранения секретного ключа на стороне хоста. Наиболее простая схема аутентификации без хранения секретного ключа на стороне хоста, которая не предполагает применение криптографической микросхемы на стороне хоста, также обеспечивает быстрый процесс симметричной аутентификации, но имеет относительно невысокую криптостойкость, так как взаимодействие в системе выполняется без случайной составляющей в криптографических преобразованиях, что предполагает неизменный характер запросов в системе, а, следовательно, возможность криптоанализа сообщений. Для повышения криптостойкости такой схемы целесообразным введение в систему взаимодействия случайной составляющей в криптографических преобразованиях и использование дополнительных процедур хеширования с промежуточным ключом, что приводит к усложнению схемы за счет двойного хеширования, но значительно повышает уровень информационной безопасности узлов IoT. Загрузка программного обеспечения в системе реализуется с помощью секретных ключей шифрования и аутентификации, которые хранятся постоянно в защищенной энергонезависимой памяти криптографических микросхем узлов IoT. При этом сеансовые ключи шифрования кода микропрограммы или ее расшифровки формируются соответственно на стороне клиента и хоста. Этот подход позволяет создавать уникальные загрузки оригинала кода микропрограмм (приложения) путем недопущения получения криптоаналитиками ее образов и алгоритмов. Особенностью схемы обмена симметричными сеансовыми ключами шифрования сообщений являются: использование секретного ключа, хранящегося на стороне хоста и клиента; определение сеансового ключа выполняется как результат хеширования случайного числа с секретным ключом, то есть обмен сеансовым ключом выполняется в зашифрованном безопасном виде.

Ключевые слова: интернет вещей; кибербезопасность; симметричная криптография; криптоаутентификация; алгоритмы шифрования; криптографические микросхемы; микроконтроллеры; микроакселераторы; туманные и облачные вычисления.