UDC 004.056: 004.9

MODERN ENCRYPTION METHODS IN IOT: HARDWARE SOLUTIONS AND CRYPTOGRAPHIC LIBRARIES FOR DATA PROTECTION

Ph.D. I. Rozlomii ORCID: 0000-0001-5065-9004
Cherkasy State Technological University, Ukraine
E-mail: inna-roz@ukr.net
Ph.D. A. Yarmilko ORCID: 0000-0003-2062-2694
Bohdan Khmelnytsky National University of Cherkasy, Ukraine
E-mail: a-ja@ukr.net
S. Naumenko ORCID: 0000-0002-6337-1605
Bohdan Khmelnytsky National University of Cherkasy, Ukraine
E-mail: naumenko.serhii1122@vu.cdu.edu.ua
P. Mykhailovskyi ORCID: 0009-0008-4324-1724
Bohdan Khmelnytsky National University of Cherkasy, Ukraine
E-mail: mykhailovskyi.pavlo1123@vu.cdu.edu.ua

Abstract. *This article presents a detailed analysis of hardware encryptors and cryptographic libraries for ensuring security in IoT systems. The Internet of Things (IoT) is currently one of the most dynamic fields, where data security is becoming critically important. The study examines how various microcontrollers use hardware and software encryption methods to protect data and provides a comparative analysis of the effectiveness of these methods. Microcontrollers, which form the backbone of many IoT devices, perform tasks ranging from sensor reading to controlling various devices. To ensure the security of data transmitted through IoT systems, it is essential to use reliable encryption methods. Hardware encryptors embedded in microcontrollers provide high performance and energy efficiency. Software cryptographic libraries, such as mbed TLS, AESLib, TinyAES, and others, offer flexibility and can be used on various platforms. The article examines encryption methods, both hardware and software, using popular microcontrollers such as the STM32F407VG and ESP32-WROOM-32 as examples. Performance, energy consumption, and security levels were measured. Performance was assessed by determining the number of encryption operations per second, which allows evaluating the real-time encryption speed. Energy consumption was measured using a precision multimeter to determine the amount of energy consumed during encryption. Security level was assessed through an analysis of the physical security of keys and resistance to various types of attacks, including brute force and side-channel attacks. The results of the study showed that hardware encryption on the STM32F407VG microcontroller provides significantly higher performance, lower energy consumption, and higher security compared to software encryption on the ESP32-WROOM-32. This confirms the high efficiency of hardware encryption for use in IoT systems that require reliable data protection.*

Keywords. *Hardware encryptor; cryptographic libraries; STM32; ESP32; data security; energy consumption; performance*

1. Introduction

In today's technological landscape, the Internet of Things (IoT) is becoming an increasingly significant part of our daily lives [1]. With the growing number of devices connected to the IoT network, the risk of data security also increases, which is one of the most pressing issues today. IoT devices, including microcontrollers, are becoming potential targets for cybercriminals who use various attack methods to gain unauthorized access to confidential information [2]. Data encryption is one of the most effective methods for protecting information in IoT. It allows data to be transformed into a format that cannot be read without a special key, thus ensuring its confidentiality even if intercepted [3]. Furthermore, encryption helps maintain data integrity and device authentication, which is particularly important in large and complex IoT systems [4].

Microcontrollers, which are the backbone of many IoT devices, use both hardware and software encryption [5]. Hardware encryptors, integrated directly into microcontrollers, provide high performance and energy efficiency [6]. At the same time, software encryption libraries offer flexibility and ease of integration into various systems.

Hardware encryptors, such as AES-encryptors and other specialized modules, are designed to accelerate the encryption and decryption processes on microcontrollers, which allows to reduce energy consumption and increase the speed of the system. They are particularly effective in resource-constrained environments, where every processor cycle and millijoule of energy matter [7]. However, hardware solutions have their limitations, including complexity in updating and maintenance, as well as possible problems with compatibility with different platforms [8].

On the other hand, software cryptographic libraries provide greater flexibility and adaptability to different conditions and security requirements. They enable the rapid integration of encryption algorithms into various IoT applications and provide a wider choice of algorithms for developers, which can be useful for adapting to new threats and security requirements [9]. At the same time, software solutions may have lower performance compared to hardware solutions due to dependence on the computing resources of the microcontroller [10]. The integration of hardware and software solutions to ensure cryptographic protection is a promising direction for the development of IoT security. Combined approaches allow you to balance performance, energy efficiency, flexibility and ease of system deployment. Research in this direction helps developers create complex solutions that can meet the specific requirements of various IoT applications [11].

The purpose of this study is to evaluate the effectiveness of hardware and software encryption on microcontrollers in the context of IoT systems. The research aims to determine the performance, energy consumption, and security level of the two encryption approaches: hardware encryption on the STM32F407VG microcontroller and software encryption on the ESP32-WROOM-32 microcontroller using the mbed TLS cryptographic library.

2. Related works

The analysis of existing works allows identifying the trends and challenges faced by IoT device developers, as well as approaches that can be used to optimize security [12], which arise in IoT systems due to shortcomings in the implementation of cryptographic algorithms and protocols. Several publications are dedicated to data protection tasks in IoT networks, with some studies focusing on improving encryption methods [13]. In addition, considerable attention in the literature is devoted to the problems of authentication and authorization in IoT environments. Given the limited resources of IoT devices, traditional authentication approaches such as public key-based protocols are too costly in terms of power consumption and computing power. To solve this problem, light protocols based on cryptography with symmetric keys or algorithms based on hash functions are proposed, which significantly reduces resource requirements and provides a sufficient level of security [14, 15]. New encryption algorithms are adapted for resource-constrained microcontrollers, enabling reliable information protection with minimal consumption [16]. Specifically, in [17], a new lightweight hybrid encryption algorithm with a novel design approach for IoT is discussed, while [18] presents a compact implementation of the CHAM block cipher on lower-class microcontrollers.

Regarding the libraries for implementing these ciphers on microcontrollers, several researchers [19] have developed their own libraries to facilitate the encryption implementation process. For example, the study [20] presented a multiprecision ANSI C library, which ensures efficient execution of cryptographic algorithms across various platforms. This library is optimized for resource-constrained microcontrollers and allows for high-performance execution of basic encryption operations. Another study describes Seal-embedded, a homomorphic encryption library for IoT [21]. However, it is important to note that not all libraries are equally efficient in terms of resource consumption and resistance to attacks [22].

Many studies also address the optimization of microcontroller resources when applying encryption, particularly ways to reduce computational costs and memory requirements for implementing encryption algorithms [1]. Other authors compare the encryption efficiency on different devices to ensure IoT security [23] and evaluate the performance of IoT encryption algorithms in the context of memory and energy consumption [24].

Research also considers the use of adaptive encryption methods that allow devices to dynamically change the level of protection depending on the current state of resources and the level of threat, which increases the efficiency of IoT systems in conditions of variable load. An important trend is the integration of machine learning technologies to analyze threats and automatically optimize security measures, allowing IoT devices to respond to attacks in real time. Despite some research in this area, there are certain unresolved issues and tasks that require further investigation. One of the main challenges is ensuring maximum data security with the limited resources of microcontrollers. Additionally, research on the effectiveness of encryption optimization methods to ensure low energy consumption and high data processing speed in IoT networks remains relevant.

3. Microcontrollers in IoT

Microcontrollers are the foundation of most IoT devices, providing data processing, communication, and device management. Their typically rich functional capabilities make them suitable for a variety of applications. Microcontrollers play a key role in IoT, as they are used to control and monitor connected devices and collect data from them.

The role of microcontrollers in an IoT system usually depends on their functionality and the tasks they are assigned. The use of microcontrollers in IoT has several typical directions, including:

Sensors and Data Collection. Microcontrollers play an important role in connecting to a variety of sensors that collect information about the environment, such as temperature, humidity, light level, or motion. This data is then processed and can be used for decision making or further processing. For example, the STM32F103C8 microcontroller from STMicroelectronics' STM32 series has built-in capabilities to connect to sensors via I2C, SPI, UART interfaces, making it ideal for data acquisition projects. Thanks to its performance and low power consumption, this microcontroller can process large amounts of data in real time, ensuring efficient operation in resource-constrained environments.

Device Control. Microcontrollers can control various devices, including relays, LEDs, motors, and other actuators. This allows them to respond to changes in the environment by executing commands, such as turning on lights when motion is detected or adjusting the temperature in the HVAC system [25]. Popular microcontrollers such as the STM32 offer high performance and reliability in controlling devices, making them the choice for many IoT applications.

Communication with the Cloud or Network. Microcontrollers with built-in communication modules such as Wi-Fi, Bluetooth or Zigbee allow easy integration of IoT devices into the global network. They can transfer data to cloud service servers, providing access to information at any time and from anywhere in the world. For example, Espressif's ESP8266 offers built-in Wi-Fi support, allowing data transfer to popular platforms such as AWS, Google Cloud or Microsoft Azure without additional communication modules [26]. On-site Data Processing. Some modern microcontrollers have enough power to process data on the device before sending it to the cloud. This allows you to reduce delays, improve the efficiency of the use of network resources and increase the overall level of security, since data does not leave the boundaries of the local network. For example, the Nordic nRF52840 has a powerful ARM Cortex-M4 processor, which allows processing data received from sensors without the need to transfer them to an external server. Overall, microcontrollers are a fundamental component of IoT systems, as they perform critical tasks related to sensors, device control, communication, and data processing. In the context of these functions, ensuring data security becomes extremely important.

The requirements for protecting the confidentiality, integrity, and availability of data are the main reasons why encryption plays a crucial role in IoT. Microcontrollers collect and process a lot of sensitive information, and without proper protection, this data can become a target for cybercriminals. The use of modern cryptographic methods, such as hardware and software encryption, helps ensure reliable data protection at all levels of the IoT system.

4. Encryption on microcontrollers

Encryption on microcontrollers plays a critical role in ensuring data security in embedded systems, particularly in the context of IoT [27]. The primary objective of this process is to prevent unauthorized access to data and devices or alteration of information by transforming it into a cryptographically unintelligible form using specialized algorithms. Implementing encryption on a microcontroller involves a complex sequence of operations aimed at ensuring data confidentiality, integrity, and authenticity [28]. Many modern microcontrollers are equipped with built-in hardware modules for encryption. These hardware encryptors allow cryptographic operations to be performed at the hardware level, ensuring high data processing speed and low power consumption. Hardware encryption significantly reduces the load on the microcontroller's central processor, freeing it to perform other important tasks.

For microcontrollers that do not have built-in hardware encryption modules, software cryptographic libraries are used. These libraries provide necessary functions for encryption, decryption, and key management, implementing cryptographic algorithms at the software level.

The features of hardware encryption and the use of cryptographic libraries will be discussed in the following sections, where we will analyze their advantages, disadvantages, and compare their effectiveness in various usage scenarios in IoT systems.

4.1. Hardware encryption

Hardware encryption is an essential component of the functionality of modern microcontrollers, providing high performance and low power consumption during cryptographic operations. Hardware encryptors are integrated directly into microcontrollers and allow encryption and decryption of data at the hardware level. Figure 1 illustrates the block diagram showing the main components of the hardware encryptor and their interaction. From Figure 1, it can be seen that the input data is fed into the Encryption Processor, which performs the core cryptographic operations (Table 1) to transform Plaintext into Ciphertext.

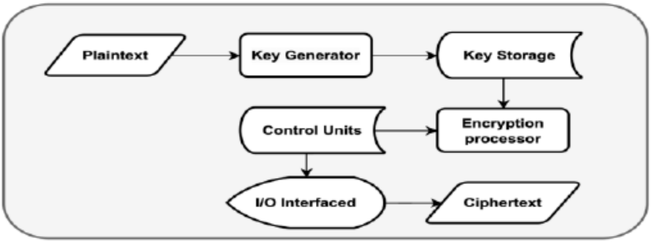

Figure 1. *Block diagram of hardware encryption process*

The Key Generator block denotes the generation of cryptographic keys using a Pseudo-Random Number Generator (PRNG) to create reliable keys. These keys are stored in the key memory and passed to the encryption processor and control blocks. Keys are stored in **The Key Storage** and are only accessible to the encryption processor through the control blocks. **Control Units** manage the encryption process, provide operation synchronization, and control access to keys and data. Input/Output **(I/O Interfaces)** Interfaces facilitate data exchange between the hardware encryptor and other parts of the microcontroller or external devices. They transmit encrypted data to the next processing or storage stages. Encrypted data (Ciphertext) emerges at the output of the encryption processor, which can be safely transmitted or stored. Upon completion of the encryption process, the data will be protected from unauthorized access.

Let's consider the properties of popular microcontrollers and SoCs that have built-in hardware capabilities for encryption.

1. **STM32.** STM32 microcontrollers typically feature built-in AES (Advanced Encryption Standard) hardware encryption, providing efficient and secure data encryption [29]. Using this hardware encryption allows encryption operations to be performed without significant CPU resource consumption [30].
2. **ESP8266 and ESP32.** These microcontrollers from Espressif also support hardware encryption [31]. For example, ESP8266 has a built-in AES hardware encryptor, while ESP32 additionally supports hardware encryption using SHA (Secure Hash Algorithm) hash functions.
3. Raspberry Pi and BeagleBone. These single-board computers can also utilize hardware encryption. For instance, Raspberry Pi can employ a built-in AES hardware encryptor available through the processor's advanced capabilities [32].
4. PIC. Microcontrollers from Microchip, such as PIC, may also feature hardware encryption [33]. For example, some PIC models have built-in AES hardware encryptor.
5. Nordic nRF52. This microcontroller has built-in AES hardware encryption support, ensuring efficient data encryption with minimal energy consumption, which is particularly crucial for battery-operated IoT devices.
6. TI CC3200. Texas Instruments' microcontroller supports AES hardware encryption, making it ideal for IoT connected devices due to its built-in Wi-Fi and low power consumption.

7. NXP LPC. Some LPC microcontroller models feature a built-in AES hardware encryptor, enabling cryptographic operations to be performed more efficiently with less strain on the processor, ensuring high computational power and flexibility.

Table 1

Encryption processor operations

Operation Description	Formula
Non-linear substitution of each byte of the plaintext using a substitution table (S-box)	$SubBytes(a) = S(a)SubBytes(a) = S(a)$
Cyclic shift of bytes in each row of the state matrix	$ShiftRows(s) = s'ShiftRows(s) = s'$
Linear transformation of each column of the state matrix	$MixColumns(s) = s''MixColumns(s) = s''$
Bitwise XOR between the state matrix and the round key	$AddRoundKey(s,k) = s \oplus k\ AddRoundKey(s,k) = s \oplus k$

Hardware encryption not only increases the performance and efficiency of IoT devices, but also provides flexibility in choosing an approach to data protection. Depending on the needs of a specific system, developers can choose between different options of microcontrollers and SoCs with integrated hardware encryption tools, optimizing the relationship between performance, power consumption and level of security. The use of hardware encryption becomes especially important in resource-constrained environments where microcontrollers run on batteries or have other power constraints. Built-in cryptographic accelerators allow you to perform cryptographic operations quickly and with minimal impact on the performance of the main processor, which extends the life of devices and ensures constant security. In addition, the integration of hardware encoders into microcontrollers makes it possible to implement multi-level protection strategies. For example, hardware data encryption can be combined with additional security measures, such as cryptography based on symmetric and asymmetric keys or the use of hashing algorithms for data integrity. Such a combined approach makes it possible to significantly increase the level of security of the IoT system against various types of attacks, in particular brute force attacks or physical attacks. The use of microcontrollers with built-in cryptographic accelerators allows you to create modern IoT solutions that meet the requirements for protecting information in conditions of real threats and limited resources.

4.2. Cryptographic libraries for microcontrollers

Hardware encryption provides significant advantages in performance and security, as encryption is performed by specialized components of the microcontroller. However, not all microcontrollers have built-in cryptographic accelerators. In such cases, software cryptographic libraries come to the rescue, implementing the necessary cryptographic algorithms and functions at the software level. Cryptographic libraries play an important role in ensuring data security in microcontroller-based systems, especially when hardware encryptors are unavailable or insufficient to perform all necessary cryptographic operations. These libraries implement various cryptographic algorithms and functions at the software level, providing flexibility and a wide range of capabilities for developers.

The first step in implementing encryption is initializing the necessary cryptographic libraries on the microcontroller. This stage involves selecting the cryptographic algorithm, generating keys, and configuring operating modes (encryption or decryption). Subsequent procedures involve data preparation, including dividing it into blocks if the algorithm requires working with data blocks, and adding additional information, such as an initialization vector, to enhance encryption security. Data encryption is performed according to the selected cryptographic algorithm and key, including adding data for authentication (if necessary), key expansion, and performing mathematical encryption operations on data blocks. Upon completion of the encryption operations, encrypted data is obtained, which can be used for transmission or storage in a secure location.

The list of popular cryptographic libraries relevant for microcontroller-based projects is quite extensive:

1. **Crypto** – a cryptographic function library for Arduino and other microcontrollers [34]. It contains implementations of various encryption algorithms, such as AES (Advanced Encryption Standard) and DES (Data Encryption Standard), as well as hash functions like MD5 and SHA. However, it may require more resources than some other libraries.
2. **AESLib** – a library specializing in the implementation of the AES algorithm, widely used for data encryption. It allows the use of different keys and AES modes of operation to protect information. It works on most Arduino boards, including Arduino Uno, although the operation speed may be limited due to resource constraints [35].
3. **TinyAES** – a library for implementing AES on Arduino microcontrollers. It is known for its efficiency and small size, making it an ideal choice for resource-constrained embedded systems. This library offers AES implementation on AVR microcontrollers (on which Arduino Uno is based) and is optimized for the limited resources of these devices. It can be particularly useful for projects where code size and speed are crucial [36].
4. **uECC** – a library used to implement elliptic curve cryptography on microcontrollers [37]. It provides the ability to use elliptic curves for key generation and data signing.
5. **Mbedtls** – a library offering implementations of a wide range of cryptographic algorithms, but may be somewhat heavy for use on Arduino Uno platforms due to its large amount of code and resource requirements [38].

In addition to the mentioned libraries, it is important to pay attention to new developments in cryptography for embedded systems that offer improved performance and security. For example, libraries focused on post-quantum cryptography are becoming increasingly relevant due to the threat of future quantum computing that could potentially break traditional cryptographic algorithms. Such libraries can use lightweight algorithms designed to run on devices with limited computing resources and power consumption. In particular, the wolfSSL library is one of such modern cryptographic libraries developed for resource-dependent devices [39]. It supports a large set of algorithms such as RSA, ECC, AES, and post-quantum cryptography algorithms. Thanks to its modular architecture and low resource requirements, it is ideal for IoT devices that require a high level of security with minimal energy consumption. wolfSSL is actively used in projects with high security requirements, such as financial and medical applications.

Another library worth noting is BearSSL, which is designed with a focus on small code size and high performance. It supports major cryptographic algorithms and protocols, such as TLS, and can be easily integrated into existing systems [40]. Its feature is low power consumption and light weight, which allows it to be used in projects with strict limitations on memory and computing power. Choosing a cryptographic library for a microcontroller depends on specific security requirements, performance, hardware compatibility, and resource constraints.

The combination of hardware and software solutions, such as the use of hardware accelerators together with lightweight cryptographic libraries, provides effective data protection and allows you to create secure and reliable IoT systems ready for today's and future cyber security challenges.

4.3. Library compatibility with microcontrollers

Compatibility of encryption libraries with different microcontroller modifications is a crucial aspect in the development of security systems for embedded devices. The choice of an appropriate library depends on several factors, including the computational power of the microcontroller, the amount of available memory, energy consumption requirements, and support for specific cryptographic algorithms. Let's consider the main aspects of compatibility:

1. **Computational Power.** Microcontrollers with higher computational power (e.g., ARM Cortex-M series) can support more complex cryptographic algorithms and libraries, such as mbed TLS or WolfSSL. Less powerful microcontrollers (e.g., AVR or PIC) may require libraries optimized for them, such as TinyCrypt.
2. **Memory.** The amount of available RAM and Flash memory also affects the choice of library. Some encryption libraries require significant memory to store keys and perform algorithms, which can be an issue for microcontrollers with limited resources. For such cases, there are optimized libraries like micro-ecc for ECC cryptography.
3. **Energy Consumption.** For battery-powered devices, it is important to choose libraries optimized for low energy consumption. This might include the use of specific algorithms or hardware cryptography accelerators if they are available in the microcontroller.

4. **Support for Cryptographic Algorithms.** Different applications may require different cryptographic algorithms (AES, RSA, ECC, etc.). It is important to ensure that the library supports the necessary algorithms and that they are optimized for the specific microcontroller.

Comparison of the properties of cryptographic libraries for microcontrollers is presented in Table 2. In addition to technical aspects, it is important to consider the factors of support and updating of libraries. For example, some libraries, such as Mbed TLS or wolfSSL, are actively supported by the community and have regular updates that provide protection against new threats and improve performance. This can be critical for projects that require a high level of security and flexibility in adapting to new requirements. Another important aspect is the possibility of optimizing libraries for the specific needs of the project. For example, TinyAES or micro-ecc can be modified to reduce code size or increase speed, which is important for applications that operate under resource-constrained conditions or have specific power consumption requirements. Such optimization may include the use of special processor instructions or hardware accelerators, which significantly improve overall performance and security. The choice of the appropriate library also depends on the possibility of integration with other system components. For example, libraries that support SSL/TLS protocols, such as Mbed TLS or wolfSSL, may be necessary for secure communication in Internet applications. At the same time, libraries with less functionality, such as AESLib or TinyAES, may be better suited for simple applications where data encryption is the main concern.

Table 2

Comparison of encryption libraries for microcontrollers

Library	Supported Algorithms	Advantages	Compatibility
Crypto	AES, RSA, SHA, others	Wide range of functions, flexibility	All well-known microcontrollers
AESLib	AES	Low resource consumption, simplicity	AVR, ESP8266
TinyAES	AES	Low memory usage, high speed	AVR, ARM Cortex-M, ESP8266, ESP32
uECC	ECC	Low memory usage, high security	STM32, AVR, ESP32
Mbedtls	AES, RSA, ECC, SSL/TLS	Easy integration, high performance	STM32, ESP32, ARM Cortex-M
mbed TLS	AES, RSA, ECC, others	High performance, flexibility	STM32, ESP32, NXP LPC

5. Results

The study used two types of microcontrollers: one with hardware encryption (STM32F407VG) and one with software encryption (ESP32-WROOM-32). To evaluate the efficiency of hardware and software encryption, the performance, energy consumption and security metrics were used

5.1. Performance

The number of encryption operations per second was measured to assess how quickly encryption can be performed in real-time.

Encryption performance was measured by determining the number of encryption operations executed per second.

This allows for evaluating the real-time encryption speed and comparing the efficiency of different encryption approaches. For software encryption on the ESP32-WROOM-32, the mbed TLS cryptographic library was used. Each microcontroller was set up to perform encryption on a fixed-size data block (16 bytes) using the AES algorithm.

Specific programs were developed to execute encryption in a loop for this purpose. The testing procedure involved running the encryption program on the microcontroller and measuring the time required to perform a certain number of encryption operations (1000 operations).

The number of encryption operations per second was then calculated by dividing the total number of operations by the total execution time. Data was collected for each type of encryption (hardware and software).

The test results were recorded and analyzed to determine the average performance value, after which graphs were created to visualize the performance comparison between hardware and software encryption.

The measurements showed that hardware encryption on the STM32F407VG significantly outperformed software encryption on the ESP32-WROOM-32 in terms of the number of encryption operations per second (Figure 2).

This confirms the high efficiency of hardware encryption for use in systems requiring fast and reliable data encryption.

5.2. Energy consumption

Energy consumption was measured during the execution of encryption operations. This is crucial for devices with limited power resources, such as battery-powered devices. Software encryption on the ESP32-WROOM-32 was performed using the mbed TLS cryptographic library.

Each microcontroller was configured to encrypt a fixed-size data block (e.g., 16 bytes) using the AES algorithm. Corresponding programs were developed to perform encryption in a loop.

Energy consumption was measured using a precision multimeter or similar equipment that can accurately measure the current and voltage consumed by the microcontroller.

After starting the encryption program on the microcontroller, the current and voltage were measured during the encryption operations. Power was calculated, and the time required to perform a certain number of encryption operations (1000 operations) was measured.

Data was collected for each type of encryption (hardware and software), and the measurement results were recorded and analyzed to determine the average energy consumption during the encryption operations.

Graphs were then created to visualize the comparison of energy consumption between hardware and software encryption.

Measurements showed that hardware encryption on the STM32F407VG had significantly lower energy consumption compared to software encryption on the ESP32-WROOM-32 (Figure 2).

This confirms the high efficiency of hardware encryption for use in energy-constrained systems that require effective encryption with minimal energy consumption.

5.3. Security

The level of protection was assessed, including the physical protection of keys and resistance to attacks. Evaluating the security level of encryption involved analyzing the physical protection of keys and resistance to various types of attacks. This stage is critically important for ensuring reliable data protection, especially in the face of increasing threats in the field of information security. Hardware encryption provides a higher level of security because keys are stored in specialized memory of the microcontroller, making unauthorized access more difficult. Software encryption was supported by the mbed TLS cryptographic library, which stores keys in the microcontroller's general memory, making them more vulnerable to attacks.

Resistance to attacks was assessed by conducting simulations of various types of attacks, including brute-force attacks, side-channel attacks, and differential attacks. Tests were conducted for each microcontroller to model possible attack scenarios and determine how effectively each type of encryption could withstand these threats.

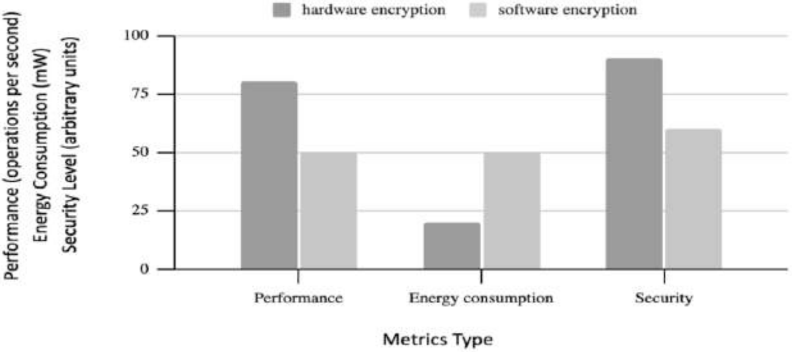

Figure 2. *Performance, energy consumption and security levels in hardware and software encryption*

Data on the physical protection of keys and the results of tests for resistance to attacks were collected and analyzed to determine the overall security level of each type of encryption. Graphs were then constructed to visualize the comparison of security levels between hardware and software encryption.

The results showed that hardware encryption on the STM32F407VG provides a significantly higher level of security compared to software encryption on the ESP32-WROOM-32 (Figure 2). This confirms the importance of using hardware encryption to protect critical data in the face of increasing threats in the field of information security.

6. Discussion

Further research in the field of hardware and software encryption on microcontrollers could focus on several key aspects. Firstly, analyzing the efficiency of hardware encryption on new and emerging microcontrollers in the market would be worthwhile.

This would allow assessing the potential of new hardware platforms in terms of security and performance.

Secondly, it's important to develop and optimize software cryptographic libraries to achieve better performance and energy efficiency across different hardware platforms.

Optimization may include algorithm enhancements, reducing memory consumption, and lowering energy consumption, which is particularly crucial for resource-constrained devices.

Integrating hardware and software encryption with other security protocols, such as TLS, is another promising direction. This would provide comprehensive protection for IoT systems, enhancing their resilience to various types of attacks and ensuring secure data transmission.

Also, one of the directions of future research is the integration of post-quantum protocols in encryption on microcontrollers for IoT devices [32]. Given the future threat of quantum computers capable of breaking traditional cryptographic algorithms, the development and implementation of quantum attack-resistant encryption methods is a critical task. The primary focus should be on adapting post-quantum algorithms, such as CRYSTALS-DILITHIUM, Kyber, NTRUEncrypt, and others, to the limited resources of microcontrollers. This includes optimizing memory usage, power consumption, and ensuring compatibility with existing communication protocols and hardware interfaces. Expanding the scope of testing is also an important aspect of further research. Conducting more extensive tests using different scenarios and data types would provide a more comprehensive picture of the efficiency and security of encryption under various conditions. This would help identify potential weaknesses and refine existing encryption methods.

Overall, further research has the potential to significantly enhance the security and efficiency of IoT systems. It would assist developers in selecting the best solutions for data protection, considering the specific requirements of their projects.

The results of this research could serve as a basis for creating more reliable and energy-efficient IoT devices that meet modern security challenges.

7. Conclusion

Based on the research findings, it is recommended to use microcontrollers with built-in hardware encryption engines, such as STM32, for projects requiring high performance, low power consumption, and high security levels. Hardware encryption is the most efficient solution for critical IoT applications where data reliability and processing speed are crucial.

Software cryptographic libraries like mbed TLS, AESLib, TinyAES, and others remain essential tools for projects where hardware encryption engines are unavailable or where flexibility in choosing encryption algorithms is needed. The choice of a specific library should be based on the performance, power consumption, and security requirements of the particular project.

Thus, encryption on microcontrollers serves not only as a means of data protection but also as a critical component for supporting functional security and stability in IoT systems. Implementing effective encryption methods helps mitigate risks associated with data breaches, unauthorized access, and other cyber threats, ensuring the reliability and longevity of connected devices in the modern Internet of Things world.

8. References

[1] I. Rozlomii, A. Yarmilko, S. Naumenko, Data security of IoT devices with limited resources: challenges and potential solutions, in: Proceedings of the 4th Edge Computing Workshop (doors-2024), 5 April 2024, Zhytomyr, Ukraine, ceur-ws.org, vol. 3666, 2024, pp. 85-96.

[2] W. H. Hassan, Current research on Internet of Things (IoT) security: A survey. Computer networks 148 (2019): 283-294.

[3] B. Pearson, C. Zou, Y. Zhang, Z. Ling, X. Fu, Sic 2: Securing microcontroller based IoY devices with low-cost crypto coprocessors, in: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), 02-04 December 2020, Hong Kong, pp. 372-381. doi: 10.1109/ICPADS51040.2020.00057

[4] S. Rajagopalan, S. Janakiraman, A. Rengarajan, Medical image encryption: Microcontroller and fpga perspective, in: Medical Data Security for Bioengineers, IGI Global, 2019, pp. 278-304. doi: 10.4018/978-1-7998-7705-9.ch013

[5] P. Arpaia, F. Bonavolonta, A. Cioffi, Problems of the advanced encryption standard in protecting Internet of Things sensor networks. Measurement 161 (2020): 107853.

[6] D. Dinu, A. S. Khrishnan, P. Schaumont, SIA: Secure intermittent architecture for off-the-shelf resource-constrained microcontrollers, in: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 05-10 May 2019, McLean, VA, USA, pp. 208-217. doi: 10.1109/HST.2019.8740834

[7] N. Ahmad, S. R. Hasan, A new ASIC implementation of an advanced encryption standard (AES) crypto-hardware accelerator. Microelectronics Journal 117 (2021): 105255.

[8] R. Ueno, S. Morioka, N. Miura, K. Matsuda, M. Nagata, S. Bhasin, Y. Mathieu, T. Graba, J.-L. Danger, N. Homma, High throughput/gate AES hardware architectures based on datapath compression. IEEE Transactions on Computers 69(4) (2019): 534-548.

[9] W. El Hadj Youssef, A. Abdelli, F. Dridi, M. Machhout, Hardware implementation of secure lightweight cryptographic designs for IoT applications. Security and Communication Networks, 2020(1), (2020): 8860598.

[10] B. Aslan, Yavuzer F. Aslan, M. T. Sakallı, Energy consumption analysis of lightweight cryptographic algorithms that can be used in the security of Internet of Things applications. Security and Communication Networks 2020(1) (2020): 8837671.

[11] P. Kietzmann, L. Boeckmann, L. Lanzieri, T. C. Schmidt, A Performance Study of Crypto-Hardware in the Low-end IoT, in: Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks (EWSN '21), 17 February 2021, Delft, The Netherlands, pp. 79-90.

[12] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, A. Zanella, IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. IEEE Internet of Things Journal 6(5) (2019): 8182-8201.

[13] M. Yabu, K. Sakiyama, T. Sugawara, Low-memory implementation of authenticated encryption algorithm saeaes on arm cortex-m0 microcontroller, in: 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), 13-16 October 2020, Kobe, Japan, 181-185. doi: 10.1109/GCCE50665.2020.9291948

[14] S. Mitsenko, S. Naumenko, I. Rozlomii, A. Yarmilko, Information Protection and Recovery Hamming Codes Based' Hash Technique, in: Proceedings of the 11-th International Conference «Information Control Systems & Technologies» (ICST-2023), 21-23 September 2023, Odesa, Ukraine, CEUR Workshop Proceedings, vol. 3513, pp. 64-77, 2023.

[15] A. Yarmilko, I. Rozlomii, H. Kosenyuk, Hash Method for Information Stream's Safety in Dynamic Cooperative Production System. In: Shkarlet, S., et al. Mathematical Modeling and Simulation of Systems (MODS 2021). Selected Papers of 16th International Scientific-practical Conference, 28 June – 01 July 2021, Chernihiv, Ukraine. Lecture Notes in Networks and Systems, vol 344 (2022): pp. 173–183. Springer, Cham. doi: 10.1007/978-3-030-89902-8_14

[16] I. Rozlomii, A. Yarmilko, S. Naumenko, P. Mykhailovskyi, IoT Smart Implants: Information Security and the Implementation of Lightweight Cryptography, in: 6th International Conference on Informatics & Data-Driven Medicine (IDDM'2023), 17-19 November 2023, Bratislava, Slovakia, ceur-ws.org, vol. 3609, 2023, pp. 145–156.

[17] A. Vahi, S. Jafarali Jassbi, SEPAR: A new lightweight hybrid encryption algorithm with a novel design approach for IoT. Wireless Personal Communications 114 (2020): 2283-2314.

[18] H. Kwon, H. Kim, S. J. Choi, K. Jang, J. Park, H. Kim, H. Seo, Compact implementation of CHAM block cipher on low-end microcontrollers, in: Information Security Applications: 21st International Conference (WISA 2020), 26-28 August 2020, Jeju, Korea, pp. 127-141. doi: 10.1007/978-3-030-65299-9_10

[19] Z. Temirbekova, A. Pyrkova, Z. Abdiakhmetova, A. Berdaly, Library of fully homomorphic encryption on a microcontroller, in: 2022 International Conference on Smart Information Systems and Technologies (SIST), 28-30 April 2022, Nur-Sultan, Kazakhstan, pp. 1-5. doi: 10.1109/SIST54437.2022.9945722

[20] J. Říha, J. Klemsa, M. Novotný, Multiprecision ANSI C Library for Implementation of Cryptographic Algorithms on Microcontrollers, in: 2019 8th Mediterranean Conference on Embedded Computing (MECO), 10-14 June 2019, Budva, Montenegro, pp. 1-4. doi: 10.1109/MECO.2019.8760285

[21] D. Natarajan, W. Dai, Seal-embedded: A homomorphic encryption library for the internet of things. IACR Transactions on Cryptographic Hardware and Embedded Systems 2021(3) (2021): 756-779. doi: 10.46586/tches.v2021.i3.756-779

[22] P. Panahi, C. Bayılmış, U. Çavuşoğlu, S. Kaçar, Performance evaluation of lightweight encryption algorithms for IoT-based applications. Arabian Journal for Science and Engineering 46(4) (2021): 4015-4037.

[23] A. Pyrkova, Z. Temirbekova, Compare encryption performance across devices to ensure the security of the IOT. Indonesian Journal of Electrical Engineering and Computer Science 20(2) (2020): 894-902.

[24] S. Maitra, D. Richards, A. Abdelgawad, K. Yelamarthi, Performance evaluation of IoT encryption algorithms: memory, timing, and energy, in: 2019 IEEE sensors applications symposium (SAS), 11-13 March 2019, Sophia Antipolis, France, pp. 1-6. doi: 10.1109/SAS.2019.8706017

[25] G. Lymperopoulos, P. Ioannou, Building temperature regulation in a multi-zone HVAC system using distributed adaptive control. Energy and Buildings 215 (2020): 109825.

[26] I. Rozlomii, A. Yarmilko, S. Naumenko, Analysis of Information Security Issues in Balancing Multiple Independent Containers on a Single Server, in: Proceedings of the 3rd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2023), 22–24 November 2023, Ternopil, Ukraine, Opole, Poland, ceur-ws.org, vol. 3628, 2023, pp. 450-461.

[27] C. Gao, L. Luo, Y. Zhang, B. Pearson, X. Fu, Microcontroller based IoT system firmware security: Case studies, in: 2019 IEEE International Conference on Industrial Internet (ICII), 11-12 November 2019, Orlando, FL, USA, pp. 200-209. doi: 10.1109/ICII.2019.00045

[28] Y. Kim, H. Kwon, S. An, H. Seo, S. C. Seo, Efficient implementation of ARX-based block ciphers on 8-Bit AVR microcontrollers. Mathematics 8(10) (2020): 1837.

[29] K. Rzepka, P. Szary, K. Cabaj, W. Mazurczyk, Performance evaluation of Raspberry Pi 4 and STM32 Nucleo boards for security-related operations in IoT environments. Computer Networks 242 (2024): 110252.

[30] J. H. Guo, L. T. Yuan, K. H. Hsia, Development of the IoT module SCADA using MQTT Protocol and AES. Journal of Advances in Artificial Life Robotics 2(3) (2021): 109-114.

[31] M. Al-Mashhadani, M. Shujaa, IoT security using AES encryption technology based ESP32 platform. Int. Arab J. Inf. Technol. 19(2) (2022): 214-223.

[32] P. Galkin, L. Golovkina, I. Klyuchnyk, Analysis of single-board computers for IoT and IIoT solutions in embedded control systems, in: 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), 9-12 October 2018, Kharkiv, Ukraine, pp. 297-302. doi: 10.1109/INFOCOMMST.2018.8632069.

[33] K. Tomar, PIC18F452 Microcontroller Home Security System: A Review Paper. International Journal of Innovative Research in Computer Science & Technology 10(2) (2022): 34-37.

[34] K. Yoshimoto, Y. Uetake, Y. Kodera, T. Kusaka, Y. Nogami, Evaluating a Side-Channel Resistance against Order 4 Rational Points in Arduino Cryptography Library, in: 2019 Seventh International Symposium on Computing and Networking (CANDAR), 25-28 November 2019, Nagasaki, Japan, pp. 245-250. doi: 10.1109/CANDAR.2019.00040

[35] L. Li, T. Riom, T. F. Bissyandé, H. Wang, J. Klein, Revisiting the impact of common libraries for android-related investigations. Journal of Systems and Software 154 (2019): 157-175.

[36] H. Li, M. Ninan, B. Wang, J. M. Emmert, TinyPower: Side-Channel Attacks with Tiny Neural Networks, in: 2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 6-9 May 2024, Tysons Corner, VA, USA, pp. 320-331. doi: 10.1109/HOST55342.2024.10545382

[37] R. A. Nofal, N. Tran, C. Garcia, Y. Liu, B. Dezfouli, A comprehensive empirical analysis of tls handshake and record layer on iot platforms, in: Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM ’19), 25–29 November 2019, Miami Beach, FL, USA, pp. 61-70.

[38] İ. K. Çekiş, A. Toros, N. Apaydın, İ. Ozcelık, Performance comparison of ECC libraries for IoT devices. Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering 25.2 (2024): 278-288.

[39] L. Popa, B. Groza, P. S. Murvay, Performance evaluation of elliptic curve libraries on automotive-grade microcontrollers, in: Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES’19), 26-29 August 2019, University of Kent, Canterbury, UK, pp. 1-7. doi: 10.1145/3339252.3341480

[40] K. Bürstinghaus-Steinbach, C. Krauß, R. Niederhagen, M. Schneider, Post-quantum tls on embedded systems: Integrating and evaluating kyber and sphincs+ with mbed tls, in: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS ‘20), 5-9 October 2020, Taipei, Taiwan, pp. 841-852. doi: 10.1145/3320269.3384725

СУЧАСНІ МЕТОДИ ШИФРОВАННЯ В IOT: АПАРАТНІ РІШЕННЯ ТА КРИПТОГРАФІЧНІ БІБЛІОТЕКИ ДЛЯ ЗАХИСТУ ДАНИХ

Ph.D. І. Розломій ORCID: 0000-0001-5065-9004
Черкаський державний технологічний університет, Україна
E-mail: inna-roz@ukr.net
Ph.D. А. Ярмілко ORCID: 0000-0003-2062-2694
Черкаський національний університет імені Богдана Хмельницького, Україна
E-mail: a-ja@ukr.net
С. Науменко ORCID: 0000-0002-6337-1605
Черкаський національний університет імені Богдана Хмельницького, Україна
E-mail: naumenko.serhii1122@vu.cdu.edu.ua
П. Михайловський ORCID: 0009-0008-4324-1724
Черкаський національний університет імені Богдана Хмельницького, Україна
E-mail: mykhailovskyi.pavlo1123@vu.cdu.edu.ua

Анотація. *У цій статті представлено детальний аналіз апаратних шифраторів і криптографічних бібліотек для забезпечення безпеки в системах IoT. Інтернет речей (IoT) наразі є однією з найбільш динамічних сфер, де безпека даних стає критично важливою. У дослідженні розглядається, як різні мікроконтролери використовують апаратні та програмні методи шифрування для захисту даних, і надається порівняльний аналіз ефективності цих методів. Для забезпечення безпеки даних, що передаються через системи IoT, важливо використовувати надійні методи шифрування. Апаратні шифратори, вбудовані в мікроконтролери, забезпечують високу продуктивність і енергоефективність. Програмні криптографічні бібліотеки, такі як mbed TLS, AESLib, TinyAES та інші, пропонують гнучкість і можуть використовуватися на різних платформах. У статті розглядаються методи шифрування, як апаратні, так і програмні, на прикладі популярних мікроконтролерів, таких як STM32F407VG і ESP32-WROOM-32. Вимірювали продуктивність, споживання енергії та рівень безпеки. Продуктивність оцінювалася шляхом визначення кількості операцій шифрування в секунду, що дозволяє оцінити швидкість шифрування в реальному часі. Споживання енергії було виміряно за допомогою прецизійного мультиметра для визначення кількості енергії, спожитої під час шифрування. Рівень безпеки оцінювався за допомогою аналізу фізичної безпеки ключів і стійкості до різних типів атак, включаючи атаки грубої сили та сторонні атаки. Результати дослідження показали, що апаратне шифрування на мікроконтролері STM32F407VG забезпечує значно вищу продуктивність, менше енергоспоживання та вищу безпеку порівняно з програмним шифруванням на ESP32-WROOM-32. Це підтверджує високу ефективність апаратного шифрування для використання в системах IoT, які потребують надійного захисту даних.*

Ключові слова. *Апаратний шифратор; криптографічні бібліотеки; STM32; ESP32; безпека даних; споживання енергії; продуктивність*