Analysis of cryptographic methods for ensuring security in the field of internet of things

Temirbekova Zhanerke Erlanovna[1], Abdiakhmetova Zukhra Muratovna[1], Tynymbayev Sakhybay[2]
[1]Faculty Information Technology, Kazakh National University Named After al-Farabi (KazNU), Almaty, Kazakhstan
[2]Faculty of Computer Technology and Cybersecurity, International IT University (IITU), Almaty, Kazakhstan

Article Info

Article history:

Received Mar 12, 2024
Revised Sep 26, 2024
Accepted Sep 30, 2024

Keywords:

ATmega2560 microcontroller
El-Gamal
Internet of things system
Paillier
RSA

ABSTRACT

The number of internet of things (IoT) devices continues to grow, and so do the associated concerns regarding their security and privacy. Evaluating the efficacy of cryptographic solutions within IoT systems emerges as a crucial endeavor to uphold the integrity and reliability of these systems. Amidst the rapid evolution of IoT technology, safeguarding the confidentiality, integrity, and availability of data emerges as a top priority. This article underscores the significance of deploying robust cryptographic algorithms to fortify IoT devices against a myriad of potential threats. Effective evaluation of cryptographic solutions within IoT systems entails a comprehensive analysis and comparison of diverse algorithms, coupled with an assessment of their performance, resilience against attacks, and resource utilization. Central to evaluating the effectiveness of cryptographic solutions within IoT systems is a consideration of various factors including computational complexity, power consumption of devices, ease of implementation, and compatibility with existing infrastructures. This article reviews a number of cryptographic solutions including Rivest–Shamir–Adleman (RSA), El-Gamal, Paillier. These algorithms are implemented on the ATmega2560 microcontroller, which allows for a comprehensive assessment of key parameters such as efficiency in terms of encryption and decryption time, power consumption, and memory usage of IoT devices.

Corresponding Author:

Abdiakhmetova Zukhra Muratovna
Faculty Information Technology, Kazakh National University Named After al-Farabi (KazNU)
Almaty, Kazakhstan
Email: zukhra.abdiakhmetova@gmail.com

1. INTRODUCTION

Cryptographic security plays a pivotal role in ensuring the confidentiality of data within an internet of things (IoT) system, where numerous devices interact and exchange sensitive information [1]. These devices handle a vast array of data, including personal details, medical records, and financial information. Utilizing cryptographic methods is paramount to encrypting this data, thereby guaranteeing its confidentiality and thwarting unauthorized access [2]. Given the diverse range of cryptographic algorithms available for implementation in IoT systems, it becomes imperative to discern the most suitable solutions tailored to the specific requisites and constraints of such systems. Through the evaluation of cryptographic solutions, factors such as performance, power consumption, and security meticulously scrutinized, enabling a comparative analysis. This endeavor facilitates the identification of optimal algorithms that align with the IoT system's requirements, thereby ensuring robust protection of data. Today, the IoT presents numerous benefits, including automation, real-time management, and the aggregation of vast volumes of data. However, alongside these advantages, new threats have emerged concerning the confidentiality, integrity, and availability of information within IoT networks. This underscores the crucial role of cryptographic solutions as indispensable tools for safeguarding data and ensuring security within the IoT ecosystem.

Selecting effective cryptographic solutions for IoT systems is paramount for ensuring the secure storage of devices and data. Cryptographic algorithms and protocols play a vital role in delivering data encryption, device and user authentication, as well as safeguarding against attacks and unauthorized access. This underscores the critical importance of meticulously choosing robust cryptographic mechanisms to fortify the integrity and confidentiality of IoT ecosystems. To evaluate the efficacy of cryptographic solutions in IoT systems, various aspects need to consider. Among these factors, performance holds significant importance, as IoT systems operate in real-time scenarios necessitating the processing of vast data volumes with minimal delay. Cryptographic algorithms must be optimized to function effectively on devices possessing limited computational resources and low power consumption capabilities. In examining how diverse cryptographic solutions can address the demands of IoT systems, careful attention was paid to various facets including performance, power efficiency, and security. Notably, the assessment encompassed the utilization of El-Gamal, Paillier and Rivest–Shamir–Adleman (RSA) cryptographic algorithms, all integral in safeguarding information within IoT infrastructures. Due to the constrained computing resources, limited memory, and absence of robust processors in IoT devices, they typically do not utilize sophisticated operating systems but rather rely on lightweight real-time operating systems. Moreover, IoT devices are susceptible to various attacks, with physical attacks being a common concern. Attackers may resort to tactics such as stealing the device, probing it in a controlled environment to analyze its internal workings, or attempting to reverse engineer it after purchasing it. In some instances, attackers may exploit vulnerabilities by downloading firmware from official distribution channels, analyzing it to uncover security keys and encoded passwords. Given these vulnerabilities, employing cryptographic methods emerges as an effective strategy to enhance the security of IoT devices and mitigate potential risks.

Cryptographic data encryption is the primary means of protecting information in IoT systems. Bertino and Sandhu [3], encryption algorithms such as AES and RSA are widely used to ensure the confidentiality of transmitted data. However, given the limited computing resources of IoT devices, the efficiency and performance of these algorithms must be taken into account Assiri and Almagwashi [4]. Yang *et al.* [5] discuss symmetric and asymmetric encryption algorithms, hash functions, and authentication and key management methods. For each method, its advantages, disadvantages and applicability in the context of IoT systems are assessed. Ma *et al.* [6] analyze various encryption and hashing algorithms optimized for low-performance and low-power devices. Particular attention is paid to the effectiveness and security of these methods in the context of the IoT environment. Hellaoui *et al.* [7] analyze data transfer protocols such as MQTT, CoAP and HTTP, and propose criteria for selecting cryptographic methods to ensure the security of data transfer in these protocols. Hatzivasilis *et al.* [8] analyze the vulnerabilities and threats associated with low-level communication protocols and propose cryptographic techniques to protect against such attacks. Dhanda *et al.* [9] discusses the challenges of key management in IoT deployments and provides recommendations for improving key management practices to enhance the security of IoT systems. Effective evaluation of cryptographic solutions involves analyzing various metrics such as processing speed, memory usage, and power consumption. As noted by Sakhi *et al.* [10] the evaluation of cryptographic algorithms should take into account the resource requirements of IoT devices, which requires the adaptation of traditional cryptographic methods to ensure their practical application. The resilience of cryptographic solutions to various types of attacks is also a key aspect of the evaluation. According to the research conducted by Singh *et al.* [11] it is necessary to conduct regular vulnerability tests and update cryptographic algorithms in response to new threats. Cryptographic methods should be able to withstand both current and potential future attacks, including quantum computing attacks. With the rapid development of technology and the emergence of new threats, constant updating of cryptographic solutions is required. Research such as the work Singh *et al.* [12] highlights the need for a dynamic approach to cryptographic security that includes regular testing and updating of the algorithms used. Effectively evaluating cryptographic solutions in IoT systems is a complex and multifaceted task that requires consideration of many factors, including performance, attack resistance, and resource constraints. But the authors did not consider various cryptographic techniques, such as performance, power consumption, and memory usage, or provide recommendations for their use in specific scenarios. Effective evaluation of cryptographic solutions in IoT systems is the basis for ensuring reliability, security and privacy of data. This evaluation not only helps protect against current threats, but also allows systems to adapt to future challenges. With the growing number of connected devices and increasing volumes of transmitted information, high-quality cryptographic protection is becoming an integral part of reliable and secure IoT systems. This research paper discusses the use of effective cryptographic algorithms to protect IoT devices from various threats, and also conducts a comprehensive assessment of key parameters, such as efficiency in terms of encryption and decryption time, power consumption and memory use.

2. METHOD

Data protection is necessary to ensure that the data is not misused or accessed without permission, while maintaining the privacy and personal information of the user. In this research paper, asymmetric encryption algorithms are used to protect data in IoT. It is important to carefully select and implement encryption algorithms based on the specific needs of the IoT system, and securely manage encryption keys to maintain encryption integrity.

2.1. El-Gamal cryptographic system

El-Gamal cryptographic system is an asymmetric cryptographic protocol based on the combinatorial discrete logarithm problem in limited fields [13], [14]. The El-Gamal cryptographic system consists of three main steps:

1) Generating the key through Alice generating public and private keys. Alice generates the public key and sends it to Bob;
2) Bob uses Alice's public key to send Alice an encrypted message (cipher text);
3) Alice uses Bob's private key to decrypt the encrypted text.

Key generation (Alice is the recipient of the message):

- p select (p – large prime number);
- p finding the simple root: g;
- a random integer x is chosen as its private key ($1 < x < p - 1$);
- y calculation: $y = g^x mod\ p$.
- public key: (p, g, y), closed key: (x).

Encryption (Bob – message sender):

- Open text: m ($m < p$);
- Select a random integer k: ($1 < k < p - 1$ and $\gcd(k, p - 1) = 1$);
- C_1 and C_2 calculation of the two values, where $C_1 = g^k mod\ p, C_2 = m \times y^b mod\ p$.

Decryption (Alice is the recipient of the message):

Encrypted text: (C_1, C_2);

Open text: $m = b \times a^{p-1-x}\ mod\ p$.

Example: message value by ASCII code: $codes[4] = \{67, 111, 109, 112\}$

Key generation:

$$p = 137$$
$$g = 3$$
$$x = 85$$
$$y = g^x \ \%\ p = 3^{85}\ \%\ 137 = 10$$

Public key: (p, g, y), private key: x.

Encryption:

$$k = 55$$

$$a[0] = 3^{55}\ \%\ 137 = 104, b[0] = 10^{55} \times 67\ \%\ 137 = 130$$

$$a[1] = 3^{55}\ \%\ 137 = 104, b[1] = 10^{55} \times 111\ \%\ 137 = 107$$

$$a[2] = 3^{55}\ \%\ 137 = 104, b[2] = 10^{55} \times 109\%\ 137 = 52$$

$$a[3] = 3^{55}\ \%\ 137 = 104, b[3] = 10^{55} \times 112\ \%\ 137 = 66$$

Decryption:

$$m[0] = 130 \times 104^{51}\ \%\ 137 = 67$$

$$m[1] = 107 \times 104^{51}\ \%\ 137 = 111$$

$$m[2] = 52 \times 104^{51}\ \%\ 137 = 109$$

$$m[3] = 66 \times 104^{51}\ \%\ 137 = 112$$

2.2. Paillier cryptographic system

The Paillier cryptosystem stands as an asymmetric algorithm renowned for its unique capability to execute operations like addition and multiplication while preserving the integrity of both encryption and encrypted data [15]-[17].

Key generation:

- $gcd\big(pq,(p-1)(q-1)\big) = 1$ large primes p and q satisfying the condition are chosen;
- $n = p \times q$ and $\lambda = lcm(p-1, q-1)$ calculated;
- $g = n + 1$ calculated;
- $\mu = \big(L(g^y \, mod \, n^2)\big)^{-1} \, mod \, n$ calculated; where $L(u) = div(\frac{u-1}{n})$;
- public key: (n, g), private key: (λ, μ).

Encryption:

- m message, where $m \in \ \mathbb{Z}_n$;
- r $(r \in \mathbb{Z}^*_{n^2})$ a random number is selected;
- $c = g^m \times r^n \, mod \, n^2$ encrypted text is calculated.

Decryption:

- c turns out an encrypted text, where $(c \in \mathbb{Z}^*_{n^2})$;
- $m = L\big(c^\lambda \, mod \, n^2\big) \times \mu \, mod \, n$ open text is calculated.

For example

By ASCII code: $codes[4]$ = {67, 111, 109, 112}

Key generation:

$$p = 31, q = 37$$
$$n = p \times q = 31 \times 37 = 1147$$
$$\lambda = 180$$
$$g = n + 1 = 1147 + 1 = 1148$$
$$\mu = \frac{g^\lambda \,\%\, n^2 - 1}{n}^{-1} \% \, n = \frac{1148^{180} \,\%\, 1147^2 - 1}{1147}^{-1} \% \, 1147 = \ 873$$

Public key:(n, g), private key: (λ, μ).

Encryption:

$$r = 600$$

$$c[0] = 1148^{67} \times 600^{1147} \ \% \ 1147^2 = 692336$$

$$c[1] = 1148^{111} \times 600^{1147} \ \% \ 1147^2 = 1614524$$

$$c[2] = 1148^{109} \times 600^{1147} \ \% \ 1147^2 = 265652$$

$$c[3] = 1148^{112} \times 600^{1147} \ \% \ 1147^2 = 490464$$

Decryption:

$$m[0] = \frac{692336^{180} \,\%\, 1147^2 - 1}{1147} \times 873 \ \% \ 1147 = 67$$

$$m[1] = \frac{1614524^{180} \,\%\, 1147^2 - 1}{1147} \times 873 \ \% \ 1147 = 111$$

$$m[2] = \frac{265652^{180} \,\%\, 1147^2 - 1}{1147} \times 873 \ \% \ 1147 = 109$$

$$m[3] = \frac{490464^{180} \,\%\, 1147^2 - 1}{1147} \times 873 \ \% \ 1147 = 112$$

2.3. RSA cryptographic system

In 1977, Rivest formulated an algorithm based on the difficulty of factoring the product of two large prime numbers. In the future, this algorithm and the entire encryption system was named RSA after the first letters of the names of the creators [18]-[21].

Before using the RSA algorithm, you first need to generate public and private keys according to the following scenario:

- Choose two large prime numbers p and q;
- Determine $n = p \times q$;
- Choose a large random number d, which should be coprime with the result of multiplication $(p-1) \times (q-1)$;
- Determine a number e, for which the relation $(e \times d)\ mod\ \left((p-1) \times (q-1)\right) = 1$ is satisfied;
- Public key - (e, n), private key – (d, n).

Encryption: To encrypt data using a public key (e, n), the following steps are performed:

- Split the source text into blocks, each of which can be denoted by an integer from 0 to $(n-1)$;
- Encrypt the text, considered as a sequence of numbers $M[i]$, on each of which the operation $C[i] = (M[i]^e)\ mod\ n$ is performed.

Decryption: Decryption is done in the same way as encryption, but here the secret key (d, n) is used: The operation $M[i] = (C[i]^d)\ mod\ n$ is performed.

RSA – Example

Input text: ASCII codes of string characters: $codes[5]$ = {78, 117, 114, 100, 97}

Key generation:

$$p = 13, q = 29$$
$$n = p \times q = 13 \times 29 = 377$$
$$\phi(N) = (p-1)(q-1) = 12 \times 28 = 336$$
$$e = 5, d = 269$$

Public key:(e, n), private key: (d, n).

Encryption:

$$c[0] = 78^5\ \%\ 377 = 169$$

$$c[1] = 117^5\ \%\ 377 = 117$$

$$c[2] = 114^5\ \%\ 377 = 316$$

$$c[3] = 100^5\ \%\ 377 = 354$$

$$c[4] = 97^5\ \%\ 377 = 327$$

Decryption:

$$m[0] = 169^{269}\ \%\ 377 = 78$$

$$m[1] = 117^{269}\ \%\ 377 = 117$$

$$m[2] = 316^{269}\ \%\ 377 = 114$$

$$m[3] = 354^{269}\ \%\ 377 = 100$$

$$m[4] = 327^{269}\ \%\ 377 = 97$$

To bolster the security of IoT devices, Atmel AVR microcontrollers were specifically chosen to implement, evaluate, and analyze cryptographic algorithms. Renowned for their low power consumption, Atmel AVR microcontrollers are particularly well-suited for IoT devices reliant on battery power or constrained power sources [22], [23]. Their cost-effectiveness further enhances their appeal for IoT device production, where cost considerations are paramount. Additionally, Atmel AVR microcontrollers are equipped with built-in peripherals such as ADC, UART, SPI, I2C, and PWM, mitigating the need for supplementary components and streamlining the design process for IoT devices [24]. The technical characteristics of the ATmega2560 microcontroller of the Atmel AVR family are shown in Table 1.

In this research, USBasp v2.0 as shown in Figure 1 serves as the tool for writing program code (firmware) onto AVR microcontrollers via a computer through a USB interface [25]. This programmer grants direct access to AVR microcontrollers for programming and debugging, eliminating the necessity for supplementary equipment. It facilitates the direct downloading of programs into the integrated flash memory of the AVR microcontroller.

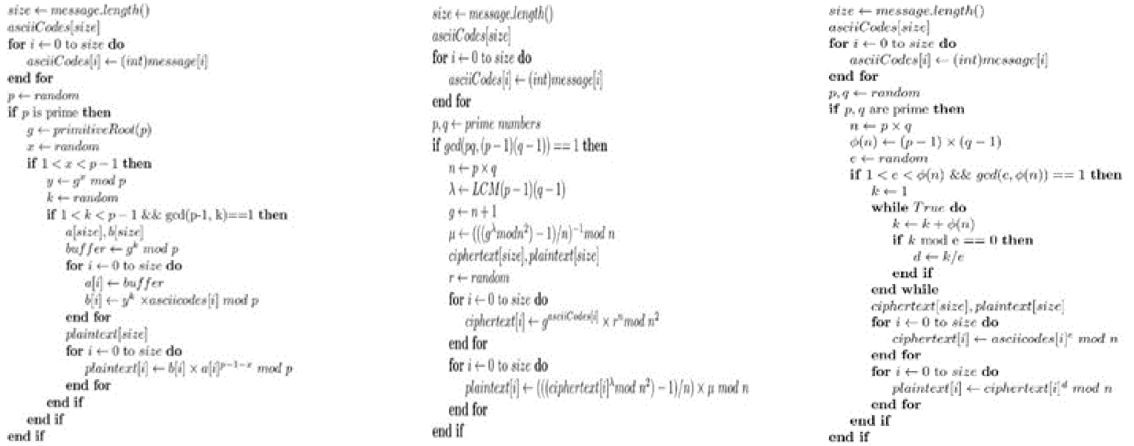
Table 1. Technical characteristics of the ATmega2560 microcontroller

Description	Value
Architecture	8-bit AVR
Voltage	2.7V – 5.5V
Flash memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock speed	Above 16 MHz
I / O pins	86
Analog input pins	16
UART	4
SPI	1
I2C	1
ADC resolution	10-bit
Temperature	-40°C to 85°C

Figure 1. USBasp v2.0 programmer

Encryption algorithms were calculated in Atmel Studio 7 using the C programming language. The pseudocodes of these algorithms are shown in Figures 2. Where Figure 2(a) shows the pseudocode of the El-Gamal cryptosystem, Figure 2(b) pseudocode of the Paillier cryptosystem, and Figure 2(c) pseudocode of the RSA cryptosystem.

The schematic diagram illustrating the connection between the ATmega2560 microcontroller and the USBasp v2.0 programmer was created using the Fritzing program. This visual representation is depicted in Figure 3, providing a clear illustration of the wiring and configuration required for the setup. Following the connection of the USBasp programmer and the ATmega2560 microcontroller, the USBasp was subsequently linked to the computer using a USB cable. Subsequent testing of the cryptosystems was conducted using the Khazama AVR program.



```
$size \leftarrow message.length()$
$asciiCodes[size]$
for $i \leftarrow 0$ to $size$ do
    $asciiCodes[i] \leftarrow (int)message[i]$
end for
$p \leftarrow random$
if $p$ is prime then
    $g \leftarrow primitiveRoot(p)$
    $x \leftarrow random$
    if $1 < x < p - 1$ then
        $y \leftarrow g^x \ mod \ p$
        $k \leftarrow random$
        if $1 < k < p - 1$ && gcd(p-1, k)==1 then
            $a[size], b[size]$
            $buffer \leftarrow g^k \ mod \ p$
            for $i \leftarrow 0$ to $size$ do
                $a[i] \leftarrow buffer$
                $b[i] \leftarrow y^k \times asciicodes[i] \ mod \ p$
            end for
            $plaintext[size]$
            for $i \leftarrow 0$ to $size$ do
                $plaintext[i] \leftarrow b[i] \times a[i]^{p-1-x} \ mod \ p$
            end for
        end if
    end if
end if
```

(a)

```
$size \leftarrow message.length()$
$asciiCodes[size]$
for $i \leftarrow 0$ to $size$ do
    $asciiCodes[i] \leftarrow (int)message[i]$
end for
$p, q \leftarrow prime \ numbers$
if $gcd(pq, (p-1)(q-1)) == 1$ then
    $n \leftarrow p \times q$
    $\lambda \leftarrow LCM(p-1)(q-1)$
    $g \leftarrow n + 1$
    $\mu \leftarrow (((g^\lambda mod n^2) - 1)/n)^{-1} mod \ n$
    $ciphertext[size], plaintext[size]$
    $r \leftarrow random$
    for $i \leftarrow 0$ to $size$ do
        $ciphertext[i] \leftarrow g^{asciiCodes[i]} \times r^n mod \ n^2$
    end for
    for $i \leftarrow 0$ to $size$ do
        $plaintext[i] \leftarrow (((ciphertext[i]^\lambda mod \ n^2) - 1)/n) \times \mu \ mod \ n$
    end for
end if
```

(b)

```
$size \leftarrow message.length()$
$asciiCodes[size]$
for $i \leftarrow 0$ to $size$ do
    $asciiCodes[i] \leftarrow (int)message[i]$
end for
$p, q \leftarrow random$
if $p, q$ are prime then
    $n \leftarrow p \times q$
    $\phi(n) \leftarrow (p-1) \times (q-1)$
    $e \leftarrow random$
    if $1 < e < \phi(n)$ && $gcd(e, \phi(n)) == 1$ then
        $k \leftarrow 1$
        while $True$ do
            $k \leftarrow k + \phi(n)$
            if $k$ mod $e == 0$ then
                $d \leftarrow k/e$
            end if
        end while
        $ciphertext[size], plaintext[size]$
        for $i \leftarrow 0$ to $size$ do
            $ciphertext[i] \leftarrow asciicodes[i]^e \ mod \ n$
        end for
        for $i \leftarrow 0$ to $size$ do
            $plaintext[i] \leftarrow ciphertext[i]^d \ mod \ n$
        end for
    end if
end if
```

(c)

Figure 2. Encryption algorithm: (a) pseudocode of the El-Gamal cryptosystem, (b) pseudocode of the Paillier cryptosystem, and (c) pseudocode of the RSA cryptosystem

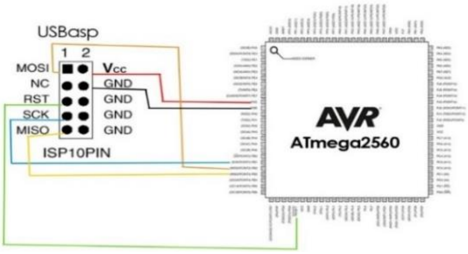


Figure 3. ATmega2560 and USBasp connection

3. RESULT ANALYSIS AND DISCUSSION

The National Institute of Standards and Technology (NIST), holds the responsibility of devising and advocating standards and guidelines to ensure the security and interoperability of information systems [26], [27]. In recent times, there has been a surge in demand for lightweight cryptographic algorithms, particularly those that can be efficiently implemented on devices with constrained resources, such as IoT devices and embedded systems. In response to this growing need, NIST has instituted a standardization process for lightweight cryptographic algorithms and has established a set of criteria for evaluating encryption algorithms within the context of IoT security. These criteria, outlined in Table 2, are meticulously crafted to aid IoT developers in selecting encryption algorithms that are not only secure but also well-suited for specific use cases.

Table 2. NIST evaluation criteria for encryption algorithms

Physical	Performance
Memory usage (RAM / ROM) implementation	Speed power (W)

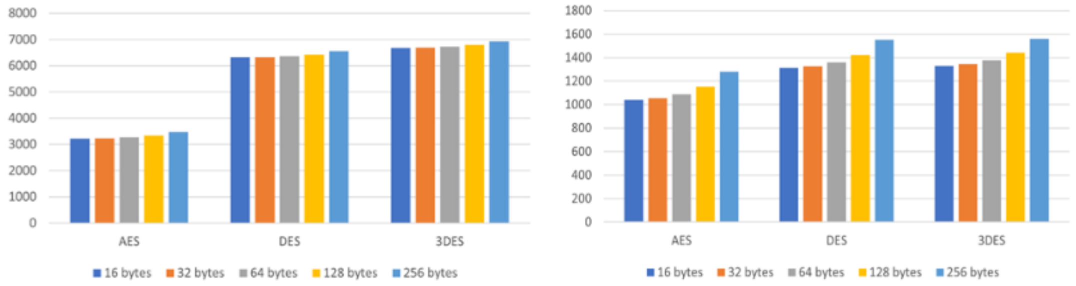
3.1. Memory usage and velocity

Memory usage in Atmel Studio 7 can be easily monitored for any program. After writing the program, simply compile the solution by selecting “Build” and then “Build Solution”. The assembly process is then displayed in the output window upon successful completion. In assessing the encryption algorithms, various input data memory sizes were tested and resulted in a comparison as shown in Figure 4. The results of data and program memory usage for the RSA, El-Gamal, and Paillier encryption algorithms are tabulated in Table 3 and shown in Figures 4(a) and 4(b).

The use of program memory (bytes) and data memory (bytes) in a microcontroller plays a key role in the design and operation of embedded systems. Both types of memory have their own unique characteristics and importance that affect the performance, functionality, and reliability of the device. Effective management of program memory and data is the basis for the successful design and operation of embedded systems. Understanding and properly using these resources helps ensure stable operation of the device, high performance, and the ability to expand functionality. As shown in Table 3, when comparing the usage of program memory (bytes) and data memory (bytes) in the microcontroller of three cryptographic algorithms, the RSA algorithm showed an effective result.

- Implementation: all encryption algorithms implemented in this paper are software-based encryption algorithms. The code that is written in Atmel Studio is software implementation, as written and compiled to be executed on the microcontroller. Loading the code onto the microcontroller using USBasp programmer involves hardware implementation, as USBasp hardware device physically connected to the microcontroller to transfer the compiled code from computer to the microcontroller's memory. Overall, the encryption algorithm that implemented is a software-based encryption algorithm, but the process of loading it onto the microcontroller involves both hardware and software components. Transferring code into the microcontroller using a USBasp programmer entails a hardware implementation, as the USBasp hardware device is physically linked to the microcontroller to transmit the compiled code from the computer to the microcontroller's memory. Although the encryption algorithm itself is typically software-based, the process of loading it into the microcontroller necessitates the involvement of both hardware and software components.
- Velocity: when estimating the encryption and decryption velocity of microcontroller encryption algorithms, myriad factors come into play, including microcontroller architecture, clock frequency, memory size, and algorithm implementation. One approach to gauging the velocity of an encryption

algorithm is by quantifying the number of clock cycles necessary to encrypt a message of a particular size. This can be accomplished utilizing a timer integrated within the microcontroller, coupled with measuring the elapsed time. The results are described in Table 4 and demonstrated in Figure 5, and are shown in Figure 5(a) encryption time Figure 5(b) decryption time.



(a) (b)

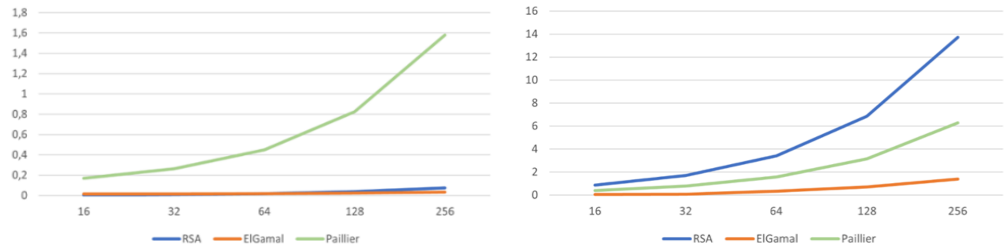
Figure 4. Comparing memory usage results for RSA, El-Gamal, and Paillier encryption (a) program memory usage and (b) data memory usage

Table 3. Memory usage results for RSA, El-Gamal, and Paillier encryption

Encryption algorithms	Use of program memory (bytes)	Data memory usage (bytes)
	Input data: 16 bytes	
RSA	3986	328
ElGamal	8404	448
Paillier	6176	360
	Input data: 32 bytes	
RSA	4100	600
ElGamal	8416	848
Paillier	6204	632
	Input data: 64 bytes	
RSA	4672	1144
ElGamal	8448	1648
Paillier	6238	1176
	Input data: 128 bytes	
RSA	5306	2232
ElGamal	8512	3248
Paillier	6302	2264
	Input data: 256 bytes	
RSA	5436	4408
ElGamal	8642	6448
Paillier	6432	4440

Table 4. Velocity of encryption algorithms (encryption, decryption)

Encryption algorithms	Efficiency in velocity	
	Encryption time (s)	Decryption time (s)
	Input data: 16 bytes	
RSA	0.004730	0.856784
El-Gamal	0.014317	0.04376
Paillier	0.169025	0.392487
	Input data: 32 bytes	
RSA	0.00946	1.713568
El-Gamal	0.015616	0.087520
Paillier	0.262294	0.784974
	Input data: 64 bytes	
RSA	0.01892	3.427136
El-Gamal	0.018214	0.350080
Paillier	0.450294	1.569948
	Input data: 128 bytes	
RSA	0.03784	6.854272
El-Gamal	0.023410	0.700160
Paillier	0.826294	3.139896
	Input data: 256 bytes	
RSA	0.07568	13.708544
El-Gamal	0.033802	1.400320
Paillier	1.578294	6.279792



(a) (b)

Figure 5. Comparison of the speed results of asymmetric cipher algorithms
(a) encryption time and (b) decryption time

Velocity and security are pivotal considerations when assessing the efficiency of encryption algorithms on a microcontroller. For instance, while El-Gamal boasts both robust security and rapid execution, the Paillier algorithm exhibits formidable cryptographic resilience yet suffers from sluggish speed. Therefore, when selecting an encryption algorithm to safeguard IoT devices, meticulous attention must be paid to both security imperatives and microcontroller performance.

3.2. Power consumption

The power draw of the ATmega2560 microcontroller hinges on various factors, including clock frequency, operational mode, and the utilization of peripherals and features. Power consumption escalates notably when the microcontroller operates in an active mode, executing instructions and engaging peripheral devices. This consumption can fluctuate from a few milliamps to tens of milliamps. It is imperative to assemble the circuit precisely as depicted in Figure 6 and gauge the current in the microcontroller employing the UNI-T UT120C multimeter. Subsequent to current measurement, computation of power consumption becomes essential utilizing the formula $P = V \times I$, where V denotes voltage (utilizing Arduino 5V) and I represents the measured current. The outcomes of power consumption for encryption algorithms are delineated in Table 5.

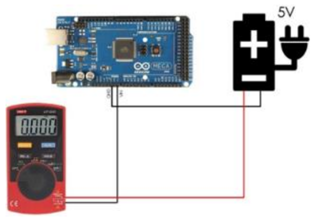


Figure 6. Arduino current measurement scheme

Table 5. Power consumption analysis of encryption algorithms

Encryption algorithms	Electric current (amp)	Power (watt)
	Input data: 16 bytes	
RSA	23.30	0.1165
El-Gamal	21.61	0.10805
Paillier	23.14	0.1157
	Input data: 32 bytes	
RSA	23.32	0.1166
El-Gamal	21.63	0.10815
Paillier	23.14	0.1157
	Input data: 64 bytes	
RSA	23.32	0.1166
El-Gamal	21.65	0.10825
Paillier	23.16	0.1158
	Input data: 128 bytes	
RSA	23.33	0.11665
El-Gamal	21.66	0.1083
Paillier	23.16	0.1158
	Input data: 256 bytes	
RSA	23.35	0.11675
El-Gamal	22.68	0.1134
Paillier	23.18	0.1159

The power consumption of an Arduino is not inherently tied to memory usage. Nonetheless, memory utilization does impact the Arduino's general performance and efficacy, consequently influencing power consumption. When an application demands significant memory resources or engages in intricate data operations, it may necessitate heightened computing resources and subsequently increase power requirements for the Arduino to function optimally.

4. CONCLUSION

Assessing the efficacy of cryptographic solutions within IoT systems is paramount in ensuring their security and reliability. With the rapid evolution of IoT technology, safeguarding data confidentiality, integrity, and availability is increasingly crucial. This study underscores the significance of employing robust cryptographic algorithms to shield IoT devices from diverse threats. The evaluation of cryptographic solutions in IoT systems necessitates comprehensive analysis and comparison of various algorithms, along with assessing their performance, resilience against attacks, and resource utilization. Throughout this study, several pivotal criteria emerged for evaluating the effectiveness of cryptographic solutions in IoT systems. These include algorithmic computational complexity, device power consumption, ease of implementation, and compatibility with existing systems. Consequently, identifying the most suitable solutions tailored to the specific requirements and constraints of IoT systems becomes imperative. Such assessments facilitate the selection of optimal encryption algorithms poised to offer robust protection for IoT devices.

REFERENCES

[1] M. A. Alhija, O. Al-Baik, A. Hussein, and H. Abdeljaber, “Optimizing blockchain for healthcare IoT: a practical guide to navigating scalability, privacy, and efficiency trade-offs,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 35, no. 3, pp. 1773–1785, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1773-1785.

[2] A. U. Karimy and P. C. Reddy, “A lightweight distributed ELM-based security framework for the internet of vehicles,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 35, no. 3, pp. 1702–1709, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1702-1709.

[3] E. Bertino and R. Sandhu, “Database security-concepts, approaches, and challenges,” *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 2–18, Jan. 2005, doi: 10.1109/TDSC.2005.9.

[4] A. Assiri and H. Almagwashi, “IoT security and privacy issues,” in *1st International Conference on Computer Applications and Information Security, ICCAIS 2018*, IEEE, Apr. 2018, pp. 1–5. doi: 10.1109/CAIS.2018.8442002.

[5] W. Yang, S. Wang, J. HuHu, and N. M. Karie, “Multimedia security and privacy protection in the internet of things: research developments and challenges,” *International Journal of Multimedia Intelligence and Security*, vol. 4, no. 1, p. 20, 2022, doi: 10.1504/ijmis.2022.121282.

[6] Z. Ma, L. Zhu, F. R. Yu, and J. James, “Protection of surveillance recordings via blockchain-assisted multimedia security,” *International Journal of Sensor Networks*, vol. 37, no. 2, p. 69, 2021, doi: 10.1504/IJSNET.2021.118486.

[7] H. Hellaoui, M. Koudil, and A. Bouabdallah, “Energy-efficient mechanisms in security of the internet of things: a survey,” *Computer Networks*, vol. 127, pp. 173–189, Nov. 2017, doi: 10.1016/j.comnet.2017.08.006.

[8] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, “A review of lightweight block ciphers,” *Journal of Cryptographic Engineering*, vol. 8, no. 2, pp. 141–184, Jun. 2018, doi: 10.1007/s13389-017-0160-y.

[9] S. S. Dhanda, B. Singh, and P. Jindal, “Lightweight cryptography: a solution to secure IoT,” *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, Jun. 2020, doi: 10.1007/s11277-020-07134-3.

[10] A. Sakhi, S. E. Mansour, and A. Sekkaki, “Enhancing security mechanisms for robot-fog computing networks,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 33, no. 3, pp. 1660–1666, Mar. 2024, doi: 10.11591/ijeecs.v33.i3.pp1660-1666.

[11] R. Singh, A. D. Dwivedi, R. R. Mukkamala, and W. S. Alnumay, “Privacy-preserving ledger for blockchain and internet of things-enabled cyber-physical systems,” *Computers and Electrical Engineering*, vol. 103, p. 108290, Oct. 2022, doi: 10.1016/j.compeleceng.2022.108290.

[12] R. Singh, A. D. Dwivedi, G. Srivastava, P. Chatterjee, and J. C. W. Lin, “A privacy-preserving internet of things smart healthcare financial system,” *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 18452–18460, Nov. 2023, doi: 10.1109/JIOT.2022.3233783.

[13] L. Malina, J. Hajny, P. Dzurenda, and S. Ricci, “Lightweight ring signatures for decentralized privacy-preserving transactions,” in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, SCITEPRESS - Science and Technology Publications, 2018, pp. 692–697. doi: 10.5220/0006890506920697.

[14] S. Jang, D. Lim, J. Kang, and I. Joe, “An efficient device authentication protocol without certification authority for internet of things,” *Wireless Personal Communications*, vol. 91, no. 4, pp. 1681–1695, Dec. 2016, doi: 10.1007/s11277-016-3355-0.

[15] P. Kumar, G. P. Gupta, and R. Tripathi, “An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks,” *Computer Communications*, vol. 166, pp. 110-124, Jan. 2021, doi: 10.1016/j.comcom.2020.12.003.

[16] R. Indrayani, H. A. Nugroho, R. Hidayat, and I. Pratama, “Increasing the security of MP3 steganography using AES encryption and MD5 hash function,” in *Proceedings - 2016 2nd International Conference on Science and Technology-Computer, ICST 2016*, IEEE, Oct. 2017, pp. 129–132. doi: 10.1109/ICSTC.2016.7877361.

[17] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, “Ascon v1.2: lightweight authenticated encryption and hashing,” *Journal of Cryptology*, vol. 34, no. 3, p. 33, Jul. 2021, doi: 10.1007/s00145-021-09398-9.

[18] R. Redhu and E. Narwal, “Polar code-based cryptosystem: comparative study and analysis of efficiency,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 32, no. 2, pp. 804-810, Nov. 2023, doi: 10.11591/ijeecs.v32.i2.pp804-810.

[19] R. Khurana and E. Narwal, "Analysis of code-based digital signature schemes," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 5, pp. 5534–5541, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5534-5541.

[20] T. R. N. Rao, "Joint encryption and error correction schemes," *ACM SIGARCH Computer Architecture News*, vol. 12, no. 3, pp. 240–241, Jun. 1984, doi: 10.1145/773453.808188.

[21] U. Musa, M. O. Adebiyi, F. B. Osang, A. A. Adebiyi, and A. A. Adebiyi, "An improved secured cloud data using dynamic rivest-shamiradleman key," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 33, no. 1, pp. 433–441, Jan. 2024, doi: 10.11591/ijeecs.v33.i1.pp433-441.

[22] A. A. Ahmed, A. H. Elmi, A. Abdullahi, and A. Y. Ahmed, "Cybersecurity awareness among university students in Mogadishu: a comparative study," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 32, no. 3, pp. 1580-1588, Dec. 2023, doi: 10.11591/ijeecs.v32.i3.pp1580-1588.

[23] N. Ahmed, U. Kulsum, M. I. Bin Azad, A. S. Z. Momtaz, M. E. Haque, and M. S. Rahman, "Cybersecurity awareness survey: An analysis from Bangladesh perspective," in *5th IEEE Region 10 Humanitarian Technology Conference 2017, R10-HTC 2017*, IEEE, Dec. 2018, pp. 788–791. doi: 10.1109/R10-HTC.2017.8289074.

[24] S. Aljawarneh, M. B. Yassein, and W. A. Talafha, "A resource-efficient encryption algorithm for multimedia big data," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22703–22724, Nov. 2017, doi: 10.1007/s11042-016-4333-y.

[25] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: an overview. Understanding the issues and challenges of a more connected world," The Internet Society (ISOC). [Online]. Available: https://www.academia.edu/34153124/The_Internet_of_Things_An_Overview_Understanding_the_Issues_and_Challenges_of_a_More_Connected_World

[26] A. S. Jaradat, M. M. Barhoush, and R. S. B. Easa, "Network intrusion detection system: machine learning approach," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 25, no. 2, p. 1151, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp1151-1158.

[27] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, "Iot intrusion detection taxonomy, reference architecture, and analyses," *Sensors*, vol. 21, no. 19, p. 6432, Sep. 2021, doi: 10.3390/s21196432.

BIOGRAPHIES OF AUTHORS

Temirbekova Zhanerke Erlanovna was born in Jambyl Region, Kazakhstan in 1989. She received the B.S. degree from the Kazakh National University named after al-Farabi in 2011 and the M.S. degree from the Kazakh National University named after al-Farabi in 2013, both in computer science. She is senior lector at Faculty of Information Technology of Al-Farabi Kazakh National University. She holds a Ph.D. degree in Computer Engineering with specialization in computer science. Her research interests include cryptography, microcontroller security, and microcontroller protection algorithms. She can be contacted at email: temyrbekovazhanerke2@gmail.com.

Abdiakhmetova Zukhra Muratovna was born in East Kazakhstan Region in 1987. She received the B.S. degree from the Kazakh National University named after al-Farabi in 2009 and the M.S. degree from the Kazakh National University named after al-Farabi in 2011, both in computer science. She is senior lector at Faculty of Information Technology of Al-Farabi Kazakh National University. She holds a Ph.D. degree in Computer Engineering with specialization in Computer science. Her research interests include cryptography, microcontroller security, neural networks, and artificial intelligent. She can be contacted at email: zukhra.abdiakhmetova@gmail.com.

Tynymbayev Sakhybay Academic degree, academic title: Professor, Candidate of Technical Sciences; In 1964 he finished Kazakh Polytechnic Institute, after this he studied in postgraduate training and receive Ph.D. degree. Major research interests: network and internet security, high-performance computing, cybersecurity, hardware encryption, and public key cryptography. Total work experience: more than 62 years. Author and co-author of more than 200 scientific works, including 2 monographs and 10 textbooks. Holder of 37 patents for inventions. He can be contacted at email: s.tynym@mail.ru.