Efficient evaluation of cryptographic solutions in IoT systems

Zhanerke Temirbekova[1,†], Zukhra Abdiakhmetova[1,†], Sakhybay Tynymbayev[2,†] and Zhansaya Bekaulova[2,†]

[1] *Al-Farabi Kazakh National University, al-Farabi71/10, Astana, Almaty, 050035, Kazakhstan,*
[2] *International Information Technology University, 34/1 Manas St., Almaty, 050000, Kazakhstan*

Abstract

Selecting cryptographic algorithms for Internet of Things (IoT) devices is a critical process as it directly impacts the security, performance, and viability of these devices. When selecting cryptographic algorithms for IoT devices, many factors need to be taken into account, the central ones being computing power, memory usage, and power consumption. The optimal choice of algorithms can significantly improve the security of devices while maintaining their performance and energy efficiency. Balancing these factors is the key to successfully integrating cryptography into the IoT ecosystem. The research paper evaluates the effectiveness of cryptographic solutions, compares and analyzes their performance, power consumption, security, and other parameters. As a result, optimal algorithms are selected that meet the requirements of the IoT system and provide effective protection.

Keywords

IoT system, cryptographic algorithms, DES cryptosystem, 3DES, AES cryptographic algorithm, Atmega 32 microcontroller

1. Introduction

Cryptographic security is essential for maintaining data confidentiality in IoT systems, where multiple devices interact and exchange sensitive information [1]. These devices manage a wide range of data, including personal details, medical records, and financial information. Encrypting this data through cryptographic methods is crucial for ensuring its confidentiality and preventing unauthorized access [2]. With a variety of cryptographic algorithms available for IoT applications, it is important to identify the most appropriate solutions that meet the specific needs and constraints of these systems. Evaluating cryptographic solutions involves a detailed analysis of factors such as performance, power consumption, and security, which allows for a comparative assessment and the identification of optimal algorithms for data protection. The IoT offers numerous advantages, including automation, real-time management, and the handling of large volumes of data. However, these benefits are accompanied by new threats to the confidentiality, integrity, and availability of information within IoT networks. This highlights the critical importance of cryptographic solutions in securing data and ensuring the overall security of the IoT ecosystem.

Cryptographic data encryption is the primary method for protecting information in IoT systems. Abber Bertino et al. [3] note that encryption algorithms like AES and RSA are commonly used to secure transmitted data. However, due to the limited computing resources of IoT devices, it is essential to consider the efficiency and performance of these algorithms, as discussed by Elisa Assiri et al. [4] Yang Wang et al. [5] review symmetric and asymmetric encryption algorithms, hash

DTESI 2024: 9th International Conference on Digital Technologies in Education, Science and Industry, October 16–17, 2024, Almaty, Kazakhstan

* Corresponding author.
† These authors contributed equally.

temyrbekovazhanerke2@gmail.com (Zh. Temirbekova); zh.bekaulova@iitu.edu.kz (Zh. Bekaulova); s.tynym@mail.ru (S. Tynymbayev); zukhra.abdiakhmetova@gmail.com (Zh. Abdiakhmetova)
0000-0003-3909-0210 (Zh. Temirbekova); 0000-0002-6619-9417 (Zh. Abdiakhmetova); 0000-0002-9326-9476 (S. Tynymbayev)

functions, and authentication and key management techniques, evaluating their benefits, drawbacks, and suitability for IoT systems. Michel Yu Zhu et al. [6] examine encryption and hashing algorithms optimized for low-performance and low-power devices, focusing on their effectiveness and security within the IoT environment. Hamed Hellaoui et al. [7] investigate data transfer protocols such as MQTT, CoAP, and HTTP, and propose criteria for selecting cryptographic methods to secure data transfer within these protocols. Chatzivasilis Georgios et al. [8] explore the vulnerabilities and threats associated with low-level communication protocols and recommend cryptographic techniques to counteract such attacks. Sandra Dhanda et al. [9] address key management challenges in IoT deployments and offer suggestions for improving key management practices to bolster IoT system security. Effectively evaluating cryptographic solutions involves assessing various metrics, including processing speed, memory usage, and power consumption. Shubhangi Handore et al. [10] emphasize that the evaluation of cryptographic algorithms should account for the resource constraints of IoT devices, necessitating adaptations of traditional methods for practical use. Additionally, the resilience of cryptographic solutions to different types of attacks is a crucial aspect of evaluation. Research by Singh Retino et al. [11] highlights the need for regular vulnerability testing and updates to cryptographic algorithms in response to new threats, including those from quantum computing. The rapid pace of technological advancement and the emergence of new threats require ongoing updates to cryptographic solutions. Rajani Singh Retino et al. [12] stress the importance of a dynamic approach to cryptographic security, involving continuous testing and updating of algorithms.

Lightweight cryptography and post-quantum cryptography are two important areas of modern cryptographic research. Lightweight cryptography aims to create efficient algorithms for devices with limited resources, while quantum-resistant algorithms are developed to protect information from potential threats associated with the development of quantum computing. Lightweight cryptography focuses on algorithms that can operate in environments with limited computing power and memory, which is especially relevant for the IoT, mobile devices, and embedded systems [13].

Evaluating cryptographic solutions in IoT systems is a complex task that must consider performance, attack resistance, and resource constraints. However, previous studies have not fully addressed various cryptographic techniques or provided specific recommendations for their application in different scenarios. A thorough evaluation of cryptographic solutions is crucial for ensuring data reliability, security, and privacy. This evaluation not only protects against current threats but also prepares systems for future challenges. As the number of connected devices and the volume of transmitted information continue to grow, robust cryptographic protection is essential for maintaining reliable and secure IoT systems. This research paper explores effective cryptographic algorithms for safeguarding IoT devices and provides a comprehensive assessment of key parameters, including encryption and decryption efficiency, power consumption, and memory usage.

2. Method

Data protection is essential to ensure that this data is not misused or accessed without permission, preserving user privacy and personal information. It is important to carefully select and implement encryption algorithms based on the specific needs of the IoT system, as well as securely manage encryption keys to maintain encryption integrity.

2.1 DES cryptosystem

Until 2001, the DES algorithm was the US federal standard for the protection of information not related to state secrets [14]. It was supported by NIST and ABA. The DES algorithm allows software and hardware implementation.

Consider an exemplary scheme of the DES algorithm. Let K be the encryption key (length 64 bits, of which 8 control bits); IP – initial permutation of bits in a block of plain text P with a length of 64 bits; $I\,P^{-1}$ – reverse to IP permutation; L and R are, respectively, the left and right half-blocks (32 bits long) of block P; k_i – internal encryption key of the i-th round, 48 bits long $(k_i = KS(i, K))$; f is the encryption function, the input of which is a 32-bit block, and the output is also a 32-bit block.

A generalized diagram of the encryption process in the DES block algorithm is shown in Figure 1.

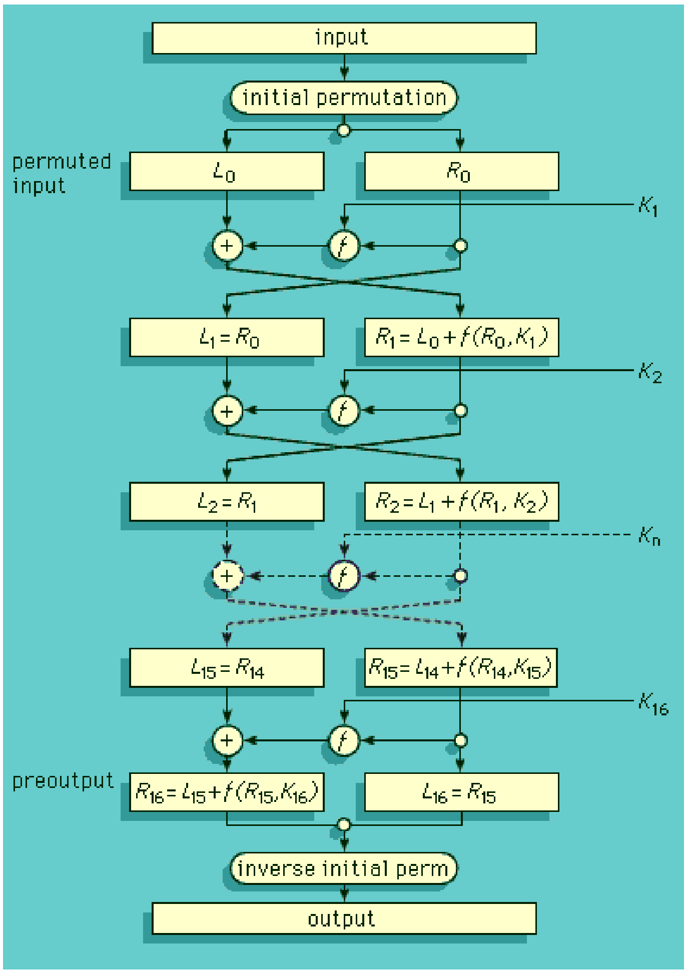


Figure 1: Encryption process of block in DES.

2.2 3DES cryptosystem

In situations where the reliability of the DES algorithm is considered insufficient, its modification is used - 3DES [15]. It should be noted that there are several variants of 3DES. As shown in Figure 2. the most commonly used variant of encryption on three keys: the plain text is encrypted on the first key, the received ciphertext - on the second and, finally, the data received after the second step - on the third key. All three keys are selected independently of each other. This cryptoalgorithm is sufficiently resistant to all attacks. Obviously, if you need to improve the security of a large fleet of equipment using DES, then it is cheaper to switch to 3DES schemes than to switch to another type of cryptosystem.

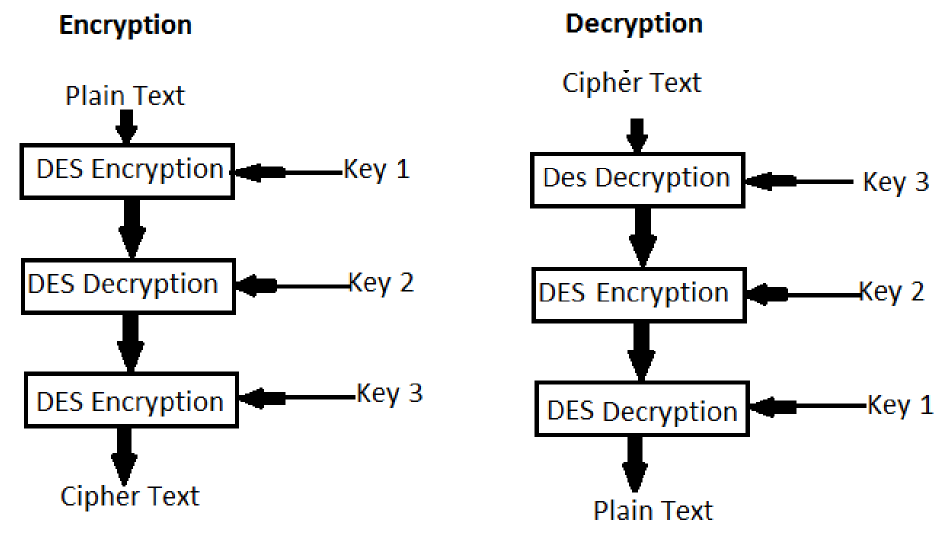


Figure 2: 3DES algorithm work with 3 keys.

3DES – Example

It is the same as DES, but there 3 keys are used.

Key1 in English: «CE1907CE» (8 ASCII characters, 1 byte each)

Key1 in Hex (64 bits): $43\,45\,31\,39\,30\,37\,43\,45$

Key2 in English: «Computer» (8 ASCII characters, 1 byte each)

Key2 in Hex (64 bits): $43\,6F6D\,70\,75\,74\,65\,72$

Key3 in English: «Kazakh12» (8 ASCII characters, 1 byte each)

Key3 in Hex (64 bits): $4B\,61\,7A\,61\,6B\,68\,31\,32$

Encryption:

DES Encryption with Key1: 9C 78 0A D4 19 C0 8C 11

DES Decryption with Key2: 77 89 4C FA 31 E5 90 73

DES Encryption with Key3: 38 D9 D6 F5 DA 22 BA 3B

Decryption:

DES Decryption with Key3: 77 89 4C FA 31 E5 90 73

DES Encryption with Key2: 9C 78 0A D4 19 C0 8C 11

DES Decryption with Key1: 41 74 79 72 61 75 30 36

2.3 AES algorithm

It was originally called «Rijndael Cipher» (pronounced «Rine Dale») after the names of the developers. It was an entrant in a competition held by NIST in 1997, to find a new secure encryption method [16]. It was the winner of this competition and thus named «AES» by 2001. It is now the most widely used symmetric key encryption algorithm in the world.

AES is a block cipher which encrypts 128 bits (16 bytes) of data at a time. It treats the 16 bytes as a grid of 4x4. Messages which are longer than 128 bits are broken into blocks of 128 bits. Each block is encrypted separately using exactly the same steps. If the message is not divisible by the block length, then padding is appended. For example, if the message is 425 bytes, 7 bytes of padding is needed to make the message 432 bytes long.

The encryption and decryption processes in AES shown in Figure 3.

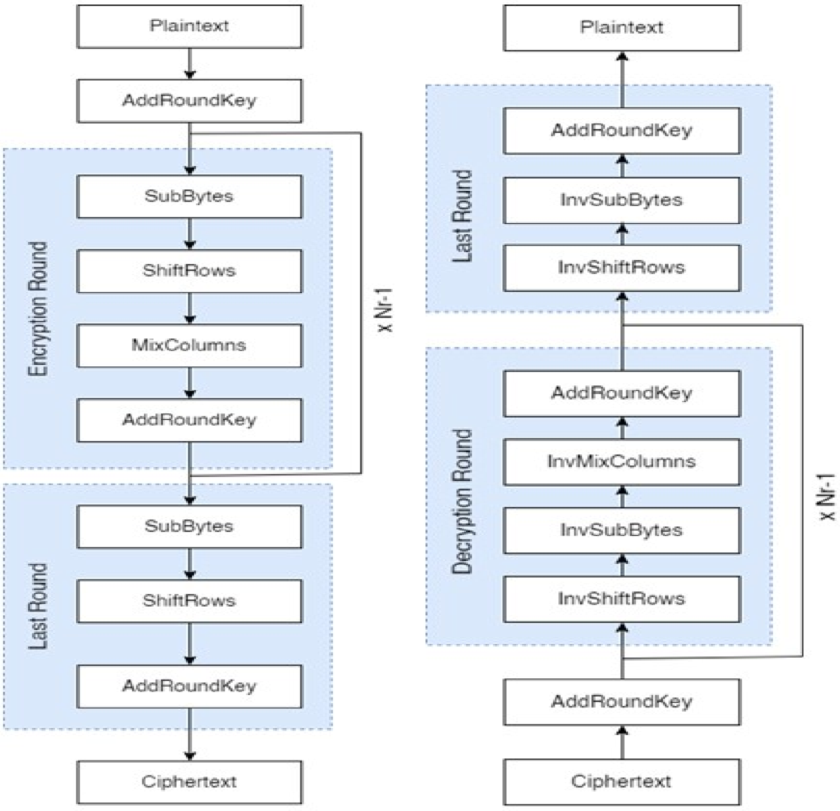


Figure 3: Encryption and decryption in AES.

3. Result analysis and discussion

In this article, it decided to work on a microcontroller from the Atmel AVR family, more precisely ATmega2560. Atmel AVR microcontrollers can be a good choice for IoT device manufacturing.

Atmel AVR microcontrollers consume very little power, which is critical for IoT devices that are often battery powered or have limited power supplies. They are relatively inexpensive, making them an attractive option for IoT device manufacturing, where cost is a major factor. They are also highly customizable, allowing developers to tailor them to specific applications and IoT requirements. They have a large community of developers and enthusiasts who provide support and resources, making it easy for developers to get started and troubleshoot any issues. Atmel AVR microcontrollers have a reputation for being rugged and reliable, which is important for IoT devices that may need to operate in challenging environments or in a variety of environments. They often come with onboard peripherals such as ADC, UART, SPI, I2C, and PWM, which can help reduce the need for additional components and simplify the design of IoT devices [17].

The technical characteristics of the ATmega2560 microcontroller of the Atmel AVR family are shown in Table 1.

Encryption algorithms in microcontroller ATmega2560

C programming language and the Atmel Studio integrated development environment created by Microchip Technology for developing and debugging applications based on AVR and ARM microcontrollers are used to program the ATmega2560 microcontroller. It is a powerful tool that provides a complete solution for developing, building and debugging applications for AVR microcontrollers. Then main task is to implement all encryption algorithms using C programming language in Atmel Studio 7. Pseudocodes of these algorithms are shown in Figures 4, 5, 6.

Table 1
Technical characteristics of ATmega2560

Characteristics	Value
Architecture	8-bit AVR
Operating Voltage	2.7V – 5.5V
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	Up to 16 MHz
I/O Pins	86
Analog Input Pins	16
UART	4
SPI	1
I2C	1
Characteristics	Value
Architecture	8-bit AVR

```
KeyExpansion()
state message
AddRoundKey(0)
for round 1 to 9 do
   SubBytes()
   ShiftRows()
   MixColumns()
   AddRoundKey(round)
end for
SubBytes()
ShiftRows()
AddRoundKey(10)
```

a)

```
state ciphertext
AddRoundKey(10)
invShiftRows()
invSubBytes()
AddRoundKey(9)
for round 8 to 0 do
  invMixColumns()
  invShiftRows()
  invRoundKey(round)
  AddRounKey(round)
end for
```

b)

Figure 5. Encryption pseudocode of DES algorithm.

```
size ← message.length()
key1_64 ← decToBinary(key1)
key1_48 ← key_gen(key1_64)
key2_64 ← decToBinary(key2)
key2_48 ← key_gen(key2_64)
key3_64 ← decToBinary(key3)
key3_48 ← key_gen(key3_64)
for block← 0 to size/8 do
      text_bin ← decToBinary(message)
      ciphertext ← encrypt (text_bin, key1_48)
      plaintext ← decrypt (ciphertext, key2_48)
      ciphertext ← encrypt (plaintext, key3_48)
      cipherHex ← BinaryToDecimal (ciphertext)
end for
```

```
size ← message.length()
key1_64 ← decToBinary(key1)
key1_48 ← key_gen(key1_64)
key2_64 ← decToBinary(key2)
key2_48 ← key_gen(key2_64)
key3_64 ← decToBinary(key3)
key3_48 ← key_gen(key3_64)
for block← 0 to size/8 do
      text_bin ← decToBinary(message)
      ciphertext ← encrypt (text_bin, key3_48)
      plaintext ← decrypt (ciphertext, key2_48)
      ciphertext ← encrypt (plaintext, key3_48)
      cipherHex ← BinaryToDecimal (ciphertext)
end for
```

Figure 6. Encryption, decryption pseudocode of 3DES algorithm.

USBasp programmer: USBasp v2.0 (Figure 7) is a programmer that is used to program AVR microcontrollers like ATmega, ATtiny, etc. It is a reliable way to program and debug AVR microcontrollers, which are commonly used in a wide range of applications such as robotics, automation, and control systems.

Figure 7: USBasp programmer.

USBasp v2.0 is an affordable programmer that is widely available. It is compatible with a wide range of AVR microcontrollers, including ATmega, ATtiny, and ATxmega. USBasp v2.0 can program AVR microcontrollers at a high speed, which makes it ideal for use in time-critical applications. The firmware for USBasp v2.0 is open-source, which means that it can be modified and customized to suit specific requirements.

ATmega2560 and USBasp programmer connection shown in Figure 8.

The NIST is responsible for developing and promoting standards and guidelines to ensure the security and interoperability of information systems. In recent years, there has been a growing demand for cryptographic algorithms that are lightweight, meaning they can be efficiently implemented on resource-constrained devices such as IoT devices and embedded systems. In response to this demand, NIST has launched a process to standardize lightweight cryptographic algorithms. he context of IoT security. These criteria, as shown in Table 3.2, are designed to help IoT developers choose encryption algorithms that are secure and appropriate for their specific use cases.

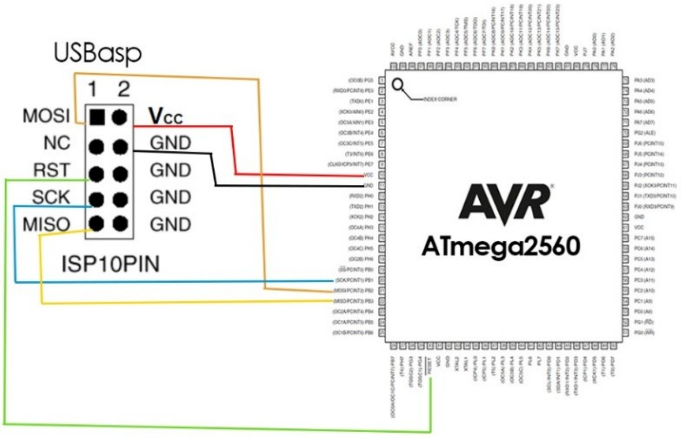


Figure 8: ATmega2560 and USBasp connection

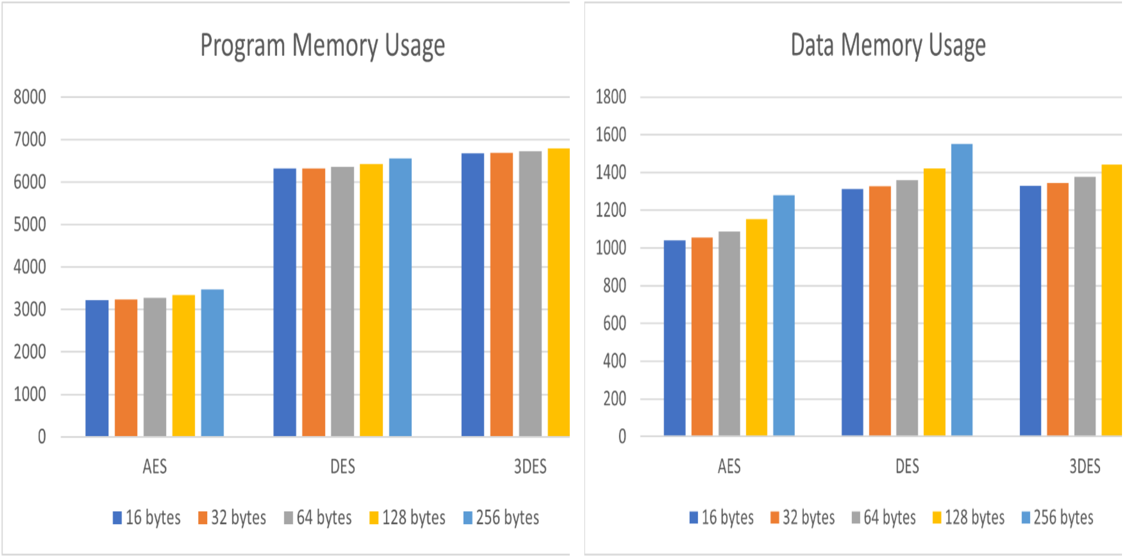
Table 2
Evaluation criteria of NIST for encryption algorithms

Evaluation criteria	
Physical	Performance
Memory usage (RAM/ROM)	
mplementation	Speed

All encryption algorithms were tested on different sizes of input data memory. The memory usage results of symmetric encryption algorithms are shown in Table 2 and Figure 9.

Table 3
Memory usage of symmetric encryption algorithms

Symmetric encryption algorithms	Program memory usage (bytes)	Data memory usage (bytes)
	Input data: 16 bytes	
AES	3224	1040
DES	6322	1311
3DES	6678	1329
	Input data: 32 bytes	
AES	3240	1056
DES	6328	1327
3DES	6694	1345
	Input data: 64 bytes	
AES	3272	1088
DES	6430	1423
3DES	6796	1441
	Input data: 16 bytes	
AES	3224	1040
DES	6322	1311
3DES	6930	1560



a) b)

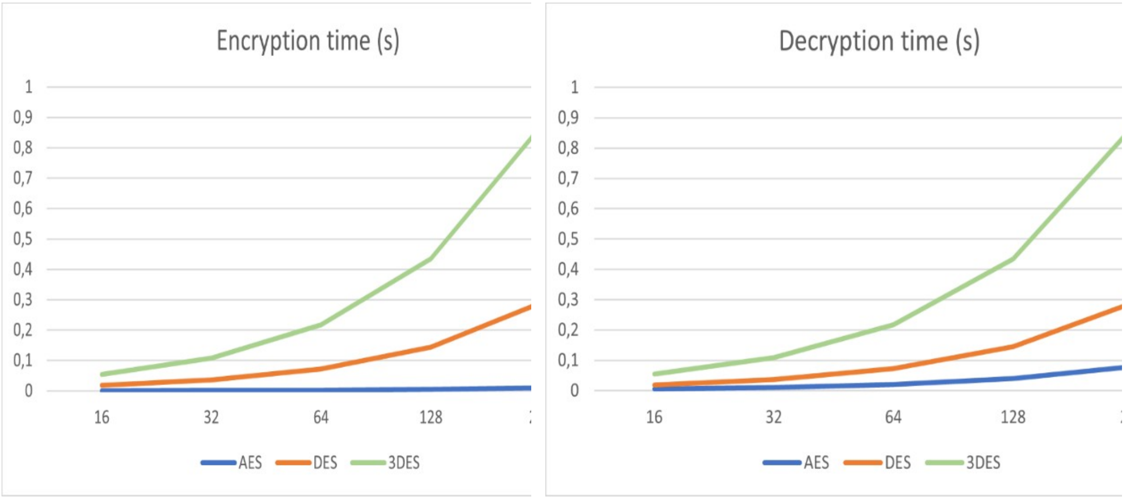
Figure 9: a)-program memory usage, b)-data memory usage of symmetric encryption algorithms.

Implementation: All encryption algorithms implemented in this article 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.

Speed: The speed evaluation of encryption algorithms in a microcontroller depends on several factors such as the architecture of the microcontroller, the clock frequency, the memory size, and the algorithm implementation. One way to evaluate the speed of an encryption algorithm is to measure the number of clock cycles it takes to encrypt a message of a certain size. This can be done using a timer in the microcontroller and measuring the elapsed time. Measured encryption and decryption time for symmetric encryption algorithms shown in Table 4, Figure 10.

Table 4
Speed performance of symmetric encryption algorithms

Symmetric encryption algorithms	Speed performance	
	Encryption time (s)	Decryption time (s)
	Input data: 16 bytes	
AES	0,000633	0,004978
DES	0,018082	0,01809
3DES	0,054288	0,054296
	Input data: 32 bytes	
AES	0,001266	0,009956
DES	0,036164	0,03618
3DES	0,108576	0,108592
	Input data: 64 bytes	
AES	0,002532	0,019912
DES	0,072328	0,07236
3DES	0,217152	0,217184
	Input data: 16 bytes	
AES	0,010128	0,079648
DES	0,289312	0,28944
3DES	0,868608	0,868736



a)

b)

Figure 10: a – encryption time, b-decryption time of symmetric encryption algorithms

When evaluating the speed of encryption algorithms in a microcontroller, it's important to consider the tradeoff between speed and security. Some encryption algorithms, such as AES, are considered to be secure and fast, while others, such as 3DES, are more secure but slower. Therefore, the choice of encryption algorithm depends on the security requirements and the performance limitations of the microcontroller.

Power Consumption: The power consumption of ATmega2560 microcontroller depends on several factors like clock frequency, operational mode, and the peripherals and features being utilized. Power consumption of microcontroller can be higher when microcontroller is in active mode and executing instructions and actively using peripherals. It can range from a few milliamps to tens of milliamps. It is necessary to assemble the circuit as shown in Figure 11, and measure the current on the microcontroller using a multimeter UNI-T UT120C. After measuring the current strength, you need to calculate the Power by the formula $P = V \times I$, where V is the voltage (Arduino use 5V), I is the measured current. Results of symmetric encryption algorithms power consumption shown in Table 5.

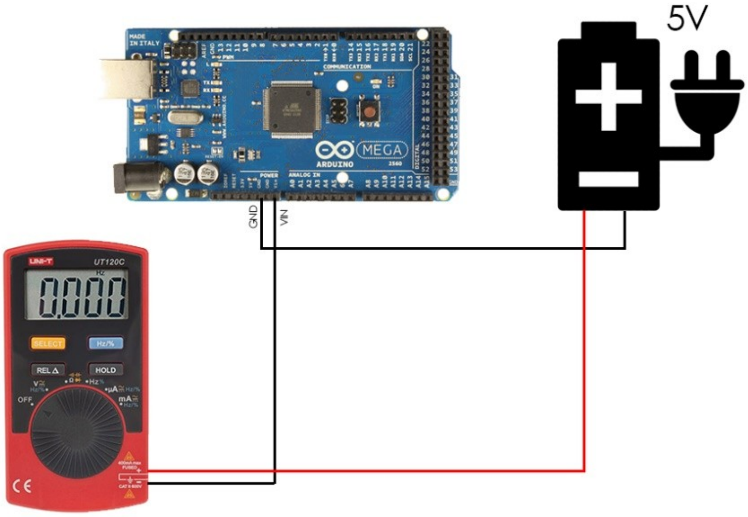


Figure 11: Scheme to measure current of Arduino.

Table 5
Power consumption of symmetric algorithms

Symmetric encryption algorithms	Electric current (amp)	Power (watt)
	Input data: 16 bytes	
AES	20,11	0,10055
DES	20,28	0,1014
3DES	20,35	0,10175
	Input data: 32 bytes	
AES	20,12	0,1006
DES	20,30	0,1015
3DES	20,37	0,10185
	Input data: 64 bytes	
AES	20,14	0,1007
DES	20,31	0,10155
3DES	20,40	0,102
	Input data: 16 bytes	
AES	20,15	0,10075
DES	20,33	0,10165
3DES	20,42	0,1021

Arduino power consumption is not directly related to memory usage. However, memory usage can affect the overall performance and efficiency of the Arduino, which in turn can affect power consumption. If an application or sketch uses a large amount of memory or performs complex data operations, it may require more computing resources and therefore more power for the Arduino to run. However, memory usage itself is not a direct factor in determining Arduino power consumption.

4. Conclusion

In conclusion, it should be noted that the evaluation of the effectiveness of cryptographic solutions in IoT plays a vital role in ensuring the security and reliability of systems. With the rapid development of IoT, ensuring confidentiality, integrity and availability of data is becoming increasingly important. This dissertation found that the use of effective cryptographic algorithms is crucial to protect IoT devices from various threats. Evaluation of the effectiveness of cryptographic solutions in IoT includes the analysis and comparison of various algorithms, as well as an assessment of their performance, resistance to attacks and resource intensity. The study identified several key criteria that should be taken into account when evaluating the effectiveness of cryptographic solutions in IoT. These include the computational complexity of algorithms, device power consumption, ease of implementation and compatibility with existing systems. As a result of the evaluation of power consumption, encryption speed, memory usage by symmetric encryption algorithms in IoT, the AES showed the best result.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Mwaffaq Abu Alhija, Osama Al Baik, Abdel rahman Hussein, Hikmat Abdeljaber Optimizing blockchain for healthcare IoT: a practical guide to navigating scalability, privacy, and efficiency tradeoffs, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 35, No.3, September 2024, pp. 1773~1785, DOI: http://doi.org/10.11591/ijeecs.v35.i3.pp1773-1785.

[2] Aziz Ullah Karimy, Putta Chandrasekhar Reddy A lightweight distributed ELM - based security framework for the internet of vehicles, Indonesian Journal of Electrical Engineering and Computer Science Vol. 35, No. 3, September 2024, pp. 1702~1709, DOI: http://doi.org/10.11591/ijeecs.v35.i3.pp1702-1709.

[3] E. Bertino and R. Sandhu. Database security - concepts, approaches, and challenges. Dependable and Secure Computing, IEEE Transactions on, 2(1):2–19, Jan.- March 2005.

[4] Assiri and H. Almagwashi, "IoT Security and Privacy Issues," 2023 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2023, pp. 1-5, doi: 10.1109/CAIS.2018.8442002.

[5] Yang W., Wang S., Hu J., Karie N.M. Multimedia security and privacy protection in the internet of things: research developments and challenges, Int. J. Multim. Intell. Secur., 4 (1) (2022), pp. 20-46, 10.1504/IJMIS.2022.121282.

[6] Ma Z., Zhu L., Yu F.R., James J. Protection of surveillance recordings via blockchain-assisted multimedia security, Int. J. Sens. Netw., 37 (2) (2021), pp. 69-80, 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," Comput. Netw., vol. 127, pp. 173-189, 2017.

[8] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A Review of Lightweight Block Ciphers," Journal of Cryptographic Engineering, vol. 8, pp. 141-184, 2018.

[9] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight Cryptography: A Solution to Secure IoT," Wireless Personal Communications, vol. 112, pp. 1947-1980, 2020.

[10] Shubhangi Handore, Pallavi Kolapkar, Pratibha Chavan, Pramod Chavan, “Enhancing security mechanisms for robot-fog computing networks”, Indonesian Journal of Electrical Engineering and Computer Science, vol. 33, no. 3, pp. 1660~1666. March 2024, doi:10.11591/ijeecs.v33.i3.

[11] Singh R., Dwivedi A.D., Mukkamala R.R., Alnumay W.S. Privacy-preserving ledger for blockchain and internet of things-enabled cyber-physical systems Comput. Electr. Eng., 103 (2022), Article 108290, 10.1016/j.compeleceng.2022.108290.

[12] Singh R., Dwivedi A.D., Srivastava G., Chatterjee P., Lin J.C.-W. A privacy preserving internet of things smart healthcare financial system IEEE Internet Things J. (2022), p. 1, 10.1109/JIOT.2022.3233783.

[13] Barker WC (2004). U.S. Department of Commerce, National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm NIST Special Publication 800-67 version 1.1. National Institute of Standards and Technology.

[14] Hafner, J. and K. McCurley (1989). “A rigorous subexponential algorithm for computation of class groups.” J. Amer. Math. Soc., 2 (4), 837– 850.

[15] Nechaev, V.I. (1994). “On the complexity of a deterministic algorithm for a discrete logarithm.” Math. Zametki, 55, 91–101.

[16] Chowdhury, Rakibul Hasan. "AI-driven business analytics for operational efficiency." World Journal of Advanced Engineering Technology and Sciences 12, no. 2(2024): 535-543.

[17] Chowdhury, Rakibul Hasan. "Quantum-resistant cryptography: A new frontier infintechsecurity." World Journal of Advanced Engineering Technology and Sciences 12, no. 2(2024): 614-621.