

SECURITY ANALYSIS FOR SMART HOME BASED  
ON INTERNET OF THINGS USING ENCRYPTION  
ALGORITHMS

MA RUNZI

UNIVERSITI KEBANGSAAN MALAYSIA

SECURITY ANALYSIS FOR SMART HOME BASED ON INTERNET OF  
THINGS USING ENCRYPTION ALGORITHMS

MA RUNZI

PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE OF  
MASTER OF CYBER SECURITY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY  
UNIVERSITI KEBANGSAAN MALAYSIA  
BANGI

2025

SECURITY ANALYSIS FOR SMART HOME BASED ON INTERNET OF  
THINGS USING ENCRYPTION ALGORITHMS

MA RUNZI

PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN  
DARIPADA SYARAT UNTUK MEMPEROLEH IJAZAH  
SARJANA KESELAMATAN SIBER

FAKULTI TEKNOLOGI SAINS DAN MAKLUMAT  
UNIVERSITI KEBANGSAAN MALAYSIA  
BANGI  
2025

## DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

I have not used any AI tools or technologies to prepare this report.

04 February 2025

MA RUNZI  
P136198

## ACKNOWLEDGEMENT

Praise to Allah the almighty, blessings and guidance have illuminated my path and given me the strength to successfully complete this study.

I am very fortunate to have Dr Faizan Qamer as my research supervisor. His excellent academic level and patient guidance gave me great help during the writing process of my thesis. Here I would like to express my heartfelt thanks to him

I would like to thank all the graduate students and PhD students of FTSM of UKM for their help and friendship during my time in block H building and for creating a pleasant working environment for me.

To my most beloved parents, Mr. Ma Feng, Ms. Huang Wei, and my dearest girlfriend, Miss. Dong Qian, thank you for your continued support.

LIBRARY FTSM

## ABSTRAK

Dengan perkembangan pesat teknologi IoT, sistem rumah pintar secara beransur-ansur disepadukan ke dalam kehidupan Sehari-hari Rakyat, membawa kemudahan yang tidak pernah berlaku sebelum ini. Walau bagaimanapun, kemajuan ini juga telah mewujudkan cabaran keselamatan maklumat baharu. Sambungan rapat peranti pintar kepada aplikasi mudah alih dan perkhidmatan awan menjadikan pelanggaran keselamatan berpotensi menjadi ancaman serius kepada privasi pengguna dan keselamatan data. Penyulitan data yang cekap dan selamat pada peranti IoT yang terhad sumber menjadi masalah mendesak untuk diselesaikan. Kajian ini memberi tumpuan kepada algoritma penyulitan yang sesuai untuk persekitaran sedemikian dan menilai prestasinya dalam senario aplikasi yang berbeza secara mendalam. Kajian ini bertujuan untuk menilai kelebihan dan kekurangan algoritma penyulitan keselamatan rumah pintar sedia ada dan senario aplikasinya. Dengan mereka bentuk eksperimen simulasi, parameter pemrosesan, kelewatan, ketepatan dan prestasi kehilangan paket algoritma AES, RSA dan ECC dalam persekitaran rumah pintar dibandingkan, dan prestasi algoritma ini dalam aplikasi praktikal dan penilaian objektifnya dianalisis. Melalui kajian literatur terperinci, kajian ini secara menyeluruh menilai algoritma penyulitan ringan sedia ada, dan membincangkan prinsip teknikalnya, ciri unik dan senario aplikasi biasa. Berdasarkan persekitaran simulasi MATLAB, satu siri eksperimen direka bentuk dan dilaksanakan untuk mengesahkan prestasi sebenar tiga algoritma penyulitan di atas. Kajian telah menunjukkan bahawa AES amat sesuai untuk peranti IoT yang terhad sumber kerana mekanisme pengkomputerannya yang cekap. RSA, kerana keselamatannya yang tinggi, sesuai untuk mewujudkan dan mengesahkan sambungan selamat, terutamanya pada peringkat awal komunikasi untuk memastikan kesahihan dan integriti maklumat. ECC menyediakan keselamatan yang setanding dengan RSA pada panjang kunci yang lebih pendek dengan penggunaan sumber yang dioptimumkan, sambil mengurangkan kerumitan pengiraan untuk memenuhi cabaran pengkomputeran kuantum masa hadapan. Melalui penilaian terperinci dan perbandingan prestasi, kajian ini menyediakan rujukan berharga untuk penyelidikan dan amalan masa hadapan, membantu meningkatkan kepercayaan pengguna terhadap privasi dan keselamatan data, sekali gus meningkatkan kepuasan keseluruhan.

## ABSTRACT

With the rapid development of IoT technology, smart home systems have gradually integrated into People's Daily lives, bringing unprecedented convenience. However, this progress has also created new information security challenges. The close connection of smart devices to mobile applications and cloud services makes security breaches potentially a serious threat to user privacy and data security. Efficient and secure data encryption on resource-constrained IoT devices becomes an urgent problem to be solved. This study focuses on encryption algorithms suitable for such environments and evaluates their performance in different application scenarios in depth. This study aims to evaluate the advantages and disadvantages of existing smart home security encryption algorithms and their application scenarios. By designing simulation experiments, the throughput, delay, accuracy and packet loss performance parameters of AES, RSA and ECC algorithms in smart home environments are compared, and the performance of these algorithms in practical applications and their objective evaluation are analysed. Through a detailed literature review, this study comprehensively evaluates the existing lightweight encryption algorithms, and discusses their technical principles, unique characteristics, and typical application scenarios. Based on MATLAB simulation environment, a series of experiments are designed and implemented to verify the actual performance of the above three encryption algorithms. Studies have shown that AES is particularly suitable for resource-constrained IoT devices due to its efficient computing mechanism. RSA, because of its high security, is suitable for establishing and verifying secure connections, especially in the early stages of communication to ensure the authenticity and integrity of information. ECC provides security comparable to RSA at a shorter key length with optimized resource usage, while reducing computational complexity to meet the challenges of future quantum computing. Through a detailed evaluation and comparison of performance, this study provides a valuable reference for future research and practice, helping to enhance user trust in privacy and data security, thereby increasing overall satisfaction.

## TABLE OF CONTENTS

		<b>Page</b>
<b>DECLARATION</b>		<b>iii</b>
<b>ACKNOWLEDGEMENT</b>		<b>iv</b>
<b>ABSTRAK</b>		<b>v</b>
<b>ABSTRACT</b>		<b>vi</b>
<b>TABLE OF CONTENTS</b>		<b>vii</b>
<b>LIST OF TABLES</b>		<b>ix</b>
<b>LIST OF ILLUSTRATIONS</b>		<b>x</b>
<b>LIST OF ABBREVIATIONS</b>		<b>xi</b>
<b>CHAPTER I</b>	<b>INTRODUCTION</b>	
1.1	Introduction	1
1.2	Problem Statement	2
1.3	Research Questions	4
1.4	Research Hypothesis	4
1.5	Research Objective	5
1.6	Contribution	5
1.7	Research Methodology	6
1.8	Thesis Format	6
	1.8.1 Introduction phase	6
	1.8.2 Literature Review phase	6
	1.8.3 Research Methodology phase	7
	1.8.4 Results and Discussion phase	7
	1.8.5 Conclusion and Recommendations phase	7
<b>CHAPTER II</b>	<b>LITERATURE REVIEW</b>	
2.1	Introduction	8
2.2	Symmetric Encryption Algorithm	9
	2.2.1 AES	9
	2.2.2 DES	14
	2.2.3 3DES	17
2.3	Asymmetric Encryption Algorithm	20
	2.3.1 RSA	20

	2.3.2	ECC	23
2.4		Other Encryption Algorithm	28
	2.4.1	ASCON	28
	2.4.2	SHA-256	31
<b>CHAPTER III</b>	<b>METHODOLOGY</b>		
3.1		Introduction	36
	3.1.1	Simulation Environment	37
	3.1.2	Dynamics of the Simulation Process	39
3.2		RSA Algorithm And Implementation	41
3.3		AES Algorithm And Implementation	42
3.4		ECC Algorithm And Implementation	44
3.5		Performance Parameters	46
	3.5.1	Throughput	46
	3.5.2	Delay	47
	3.5.3	Accuracy	48
	3.5.4	Packet Loss Ratio	48
<b>CHAPTER IV</b>	<b>RESULTS AND DISCUSSION</b>		
4.1		Results	49
	4.1.1	Throughput performance results	49
	4.1.2	Delay performance results	53
	4.1.3	Accuracy performance results	56
	4.1.4	Packet Loss Ratio performance results	59
4.2		Analysis	63
	4.2.1	Throughput Analysis	63
	4.2.2	Delay Analysis	64
	4.2.3	Accuracy Analysis	65
	4.2.4	Packet Loss Analysis	66
	4.2.5	Overall Comparative Insights	68
<b>CHAPTER V</b>	<b>CONCLUSION</b>		
5.1		Introduction	69
5.2		Experimental Summary	69
5.3		Challenges And Countermeasures	70
<b>REFERENCES</b>			<b>72</b>

**LIST OF TABLES**

<b>Table No.</b>		<b>Page</b>
Table 2.1	Algorithm comparison table	8
Table 2.2	Summary of AES literature review	14
Table 2.3	Summary of DES literature review	17
Table 2.4	Summary of 3DES literature review	19
Table 2.5	Summary of RSA literature review	23
Table 2.6	Summary of ECC literature review	27
Table 2.7	Summary of ASCON literature review	31
Table 2.8	Summary of SHA-256 literature review	35
Table 4.1	Comparison table of algorithm results	68

**LIST OF ILLUSTRATIONS**

<b>Figure No.</b>		<b>Page</b>
Figure 3.1	Workflow of the Simulation Environment	40
Figure 3.2	RSA Algorithm Flow Diagram	42
Figure 3.3	AES Algorithm Flow Diagram	43
Figure 3.4	ECC Algorithm Flow Diagram	45
Figure 4.1	Throughput vs Number of IoT Devices	49
Figure 4.2	Throughput vs Encryption Sizes	50
Figure 4.3	Throughput vs Number of Hops	51
Figure 4.4	Delay vs Number of IoT Devices	53
Figure 4.5	Delay vs Encryption Sizes	54
Figure 4.6	Delay vs Number of Hops	55
Figure 4.7	Accuracy vs Number of IoT Devices	56
Figure 4.8	Accuracy vs Encryption Sizes	57
Figure 4.9	Accuracy vs Number of Hops	58
Figure 4.10	Packet Loss Rate vs Number of IoT Devices	59
Figure 4.11	Packet Loss Rate vs Encryption Sizes	60
Figure 4.12	Packet Loss Rate vs Number of Hops	62

**LIST OF ABBREVIATIONS**

AELB	Atomic Energy Licensing Board
AES	Advanced Encryption Standard
ECC	Elliptic curve cryptography
EDE	Encryption-Decryption-encryption
FPGA	Field Programmable Gate Array
IAEA	International Atomic Energy Agency
IoT	Internet of Things
MQTT	Message Queuing Telemetry Transport
NIST	National Institute of Standards and Technology
PQC	Post Quantum Cryptography
PSoC	Programmable system-on-chip
RSA	Rivest-Shamir-Adleman
UKM	Universiti Kebangsaan Malaysia
XOR	Exclusive OR

## CHAPTER I

### INTRODUCTION

#### 1.1 INTRODUCTION

With the rapid development of IoT technology, smart home systems have gradually spread to daily life, providing users with unprecedented convenience. This progress has also created new information security challenges. Smart home devices are closely connected to mobile applications and cloud services, and any security breach may pose a serious threat to users' privacy and data security (Mousavi et al. 2021). Research and implementation of effective encryption algorithms to ensure the confidentiality, integrity and availability of smart home systems has become an essential issue.

This paper focuses on the security analysis of IoT smart home systems, especially emphasizing the core role of encryption algorithms in ensuring information transmission and storage processes, and the increasing interconnection and data exchange frequency between them makes it particularly important to ensure the security of these data (Singh et al. 2024). As the core defense line of information security, encryption technology plays an indispensable role in maintaining the confidentiality, integrity and availability of smart home systems. It can not only effectively prevent data from being stolen, tampered with or destroyed during transmission and storage, but also ensure that even if the information accidentally falls into illegal hands, if there is no corresponding decryption key, they are difficult to interpret valuable content, thus ensuring the security and stability of the smart home ecosystem (Tchagna Kouanou et al. 2022). In order to gain an in-depth understanding of this field, this paper conducts a comprehensive and detailed literature review, aiming to discuss the technical principles, unique characteristics, typical application scenarios of various encryption algorithms, and the challenges encountered in practical applications. Through extensive

investigation of existing research results, we can systematically analyze the characteristics of different encryption algorithms and identify their advantages and disadvantages. On this basis, this paper focuses on three widely used encryption algorithms: AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and ECC (Elliptic curve cryptography). Each algorithm has different advantages and limitations due to its unique mathematical foundation and technical implementation.

In order to evaluate the performance of these encryption algorithms in the smart home environment, a set of strict simulation experiments is designed, and the throughput, delay, accuracy and packet loss rate are selected as the key indicators. These indicators not only reflect the effectiveness and practicality of the encryption algorithm, but also take into account the impact of resource constraints in the smart home environment (Acar et al. 2020). Such assessments can guide smart home device manufacturers and service providers in choosing the most appropriate encryption technology to protect users' privacy and data security while optimizing the overall user experience. Through a comparative analysis of the performance results of different encryption algorithms, it is found that AES performs well in most cases, especially in terms of high throughput requirements and large-scale device deployment. In contrast, although RSA provides higher security, its performance overhead is high, and it is not suitable for real-time or high-traffic networks (Medileh et al. 2020). ECC provides a balance, finding a compromise between security and performance, but is still less efficient than AES.

## **1.2 PROBLEM STATEMENT**

Smart home provides convenient home automation and intelligent management functions for today's society. This progress also brings new information security challenges. The tight connectivity between smart home devices and mobile applications and cloud services exposes the entire system to multiple security risks. These issues not only affect the privacy and data security of users, but can also have a serious impact on the security and stability of the home environment. Ensuring the confidentiality, integrity and availability of smart home systems has become an important issue to be solved urgently.

In a smart home environment, each device can present unique security risks. These devices often communicate with each other over wireless networks and interact with cloud services, increasing the potential for security vulnerabilities. The presence of a large number of devices in smart home systems and the complex communication patterns among them make it difficult for traditional security measures to effectively deal with emerging threats (Zeadally et al. 2021). To protect users from such risks, a comprehensive approach is needed to identify and classify all types of devices and their associated security hazards. As the core defense line to maintain the security of information transmission and storage, different types of encryption algorithms have significant differences in performance and applicability. Choosing the right encryption algorithm is critical to optimizing the overall performance of a smart home system, as an inappropriate encryption scheme can lead to issues such as increased delay, decreased throughput, or excessive resource consumption, which in turn affects the user experience. Although AES, RSA and ECC have their unique advantages, they face different challenges in practical application (Abu-Tair et al. 2020). Although AES provides efficient encryption and decryption capabilities, its complexity is high, especially on resource-constrained IoT devices, which may affect response speed or power consumption. Although RSA has strong security, it is expensive to compute, especially when dealing with large-sized keys, which may cause performance degradation, and is not suitable for real-time or high-traffic networks. ECC performs well in resource-constrained environments, but is not as efficient as AES in real-time and high-traffic scenarios (Panahi et al. 2021). In addition, SHA-256, as a high-strength hash function, plays an important role in ensuring data integrity, but also presents performance challenges on resource-constrained devices due to its computation-intensive nature.

The purpose of this thesis is to discuss the information security in smart home system, especially the choice and application of encryption algorithm. Through a comprehensive analysis of the technical principles, characteristics, application scenarios of different encryption algorithms and the challenges faced in practical applications, it is hoped that it can provide valuable reference for the security practice in the field of smart home, and promote the development of more comprehensive security solutions in this field . It is also expected to improve the overall security of

smart home systems in the future to provide users with a more reliable and convenient life experience. Through detailed risk analysis, a comprehensive security model is built, covering all levels from hardware to software, and simulation experiments are used to evaluate the differences between AES, RSA and ECC in terms of throughput, delay, etc., to find out the most suitable application scenarios.

### **1.3 RESEARCH QUESTIONS**

1. What are the advantages and disadvantages of different encryption algorithms in the smart home environment and their applicable scenarios?
2. How to evaluate the actual performance of different encryption algorithms in a smart home environment using key performance indicators?
3. What are the actual performance differences of different encryption algorithms in smart home environments?

### **1.4 RESEARCH HYPOTHESIS**

1. In the smart home environment, different encryption algorithms have their advantages and disadvantages in performance, and their applicability depends on the needs of specific scenarios.
2. Key performance indicators can be measured to fully evaluate the performance of encryption algorithms in smart home environments. The throughput, delay, accuracy and packet loss rate combined can effectively compare the practicability of different encryption algorithms in smart home systems.
3. In smart home environments AES shows superior performance in most cases due to its efficient performance and low resource consumption. RSA can provide higher security, but it is not suitable for real-time or high-traffic network environments due to its high performance overhead. ECC is better suited for resource-constrained environments, but is still less efficient than AES.

## 1.5 RESEARCH OBJECTIVE

1. To evaluate existing smart home security encryption algorithms, as well as their advantages and disadvantages and applicable scenarios.
2. To design simulation experiments to compare the performance parameters (throughput, delay, accuracy and packet loss rate) of AES, RSA and ECC in smart home environment
3. To analyze the performance of different encryption algorithms in the smart home environment, and conduct an objective evaluation of their performance.

## 1.6 CONTRIBUTION

Aiming at the information security problem in smart home system, this research aims to provide valuable insights and solutions through in-depth analysis of algorithms and rigorous experimental design. By conducting a comprehensive evaluation of the application of existing encryption algorithms in the smart home environment, we not only identified potential security vulnerabilities, but also provided directions for future improvements. In particular, this study designed and implemented a series of simulation experiments to compare the performance parameters of these mainstream encryption algorithms in real-world applications, such as throughput, delay, accuracy, and packet loss rates, which can help determine the best encryption scheme for smart home environments, especially when considering real-time and high-traffic network requirements.

Through the objective evaluation of the performance of different encryption algorithms, the advantages and disadvantages of each algorithm are revealed, and the guiding suggestions are provided for the selection of appropriate encryption technology, which is of great significance to ensure the confidentiality, integrity and availability of smart home systems. This research is committed to promoting the information security of the smart home ecosystem, enhancing user privacy and data security by building a more secure and reliable smart living space, thereby enhancing users' trust in their privacy and data security, and thus improving users' overall satisfaction.

## 1.7 RESEARCH METHODOLOGY

The methodology for the three research objective are briefly described below :

1. Evaluate the effectiveness of existing smart home security encryption algorithms, including a literature review to assess their advantages and disadvantages and applicable scenarios.
2. Design experiments with MATLAB to simulate and capture key performance indicators such as throughput, delay, accuracy and packet loss rate in the IoT environment. This section describes how to compare the performance of AES, RSA, and ECC under different conditions (Number of IoT Devices, Encryption Size, and Number of Hops) based on the values of the selected indicator.
3. The simulation results of smart home IoT environment are analysed and discussed. Focus on analysing and interpreting key result values and quantities in line charts.

## 1.8 THESIS FORMAT

### 1.8.1 Introduction phase

The introduction section aims to provide background information on the research, outlining the importance of smart home systems and their information security. This section discusses the central role of encryption algorithms in maintaining the confidentiality, integrity, and availability of information in smart home systems, emphasizing their irreplaceable role in preventing data theft, tampering, or destruction. The introduction also describes the purpose and motivation of the study, as well as the specific problems it aims to address, and briefly mentions the content arrangement of subsequent chapters.

### 1.8.2 Literature Review phase

The literature review section explores in depth various encryption algorithms applied in the field of IoT smart home, including symmetric encryption, asymmetric encryption

and lightweight encryption. Based on the comprehensive analysis of the existing literature, this section systematically describes the technical principles, unique characteristics, typical application scenarios and advantages and disadvantages of these algorithms. This section not only summarizes the current research status, but also points out the unsolved problems and the direction of future research, which provides a theoretical basis for the following research design.

### **1.8.3 Research Methodology phase**

The research Methods section describes in detail the experimental design and simulation processes used to evaluate the performance of different encryption algorithms. This section explains how to select experimental parameters and why they are important for evaluating encryption algorithms in a smart home environment. It will also introduce the way of setting up the experimental platform, including the selection of software tools and parameter configuration, to ensure the effectiveness and repeatability of the experimental results.

### **1.8.4 Results and Discussion phase**

The results and discussion section presents the main findings obtained by the above experimental methods. This section presents the specific performance of different encryption algorithms in the smart home environment, compares their advantages and disadvantages, and discusses the actual application scenarios. Visually present the experimental results in the form of charts and statistics to better understand the differences between algorithms.

### **1.8.5 Conclusion and Recommendations phase**

Conclusions and Suggestions summarize the main results of this study, reiterate the significance of the research findings, and put forward specific suggestions for the application of encryption algorithms in smart home environments.

## CHAPTER II

### LITERATURE REVIEW

#### 2.1 INTRODUCTION

As the core defence line of information security, encryption algorithm plays an irreplaceable role in maintaining the confidentiality, integrity and availability of smart home systems. By encrypting the data, even if the information accidentally falls into the hands of criminals, if they cannot obtain the corresponding decryption key, it is difficult to interpret the valuable content, thus ensuring the security and stability of the smart home ecology. This chapter focuses on various encryption algorithms applied in the field of IoT smart home, aiming to systematically analyse the technical principles, unique characteristics, typical application scenarios, and challenges faced in practical applications of different encryption algorithms through a comprehensive and in-depth literature review.

Table 2.1 Algorithm comparison table

Algorithm	Parameters Calculated	Advantages	Disadvantages
AES	Throughput, Delay, Packet Loss, Accuracy	High speed, low power, adaptable to IoT constraints	High resource demand for certain implementations
RSA	Authentication, Integrity	Strong security, suitable for establishing secure connections	High computational overhead, not suitable for real-time systems
ECC	Key Exchange, Authentication	Lightweight, efficient on resource-constrained devices	Performance bottlenecks in key generation under heavy loads
ASCON	Encryption and Integrity Verification	Lightweight, robust against certain quantum attacks	Susceptibility to certain side-channel attacks (e.g., voltage burrs)

... to be continued

... continuation

SHA-256	Data integrity verification	High security, strong collision resistance, widely used in blockchain and other fields	The computing complexity is high, which may burden resource-limited devices
---------	-----------------------------	--	---

---

## 2.2 SYMMETRIC ENCRYPTION ALGORITHM

### 2.2.1 AES

The fundamental idea behind AES, a symmetric encryption method that is frequently used in IoT smart homes, is to encrypt data blocks of a set length by performing several rounds of intricate transformation operations. Specifically, it uses 128-bit (also can support 192 or 256 bit) data blocks as units, and uses substitution, substitution, obfuscation and other operations in each round of encryption, with the help of specific S-box, row shift and column obfuscation transformation methods, so that the original data is gradually encrypted into ciphertext (Rahman et al. 2022). AES-128 prevents hackers from intercepting and cracking commands during data transmission, preventing unauthorized unlock. According to the research of Javed, AES-128 encryption algorithm is deeply applied to the transmission encryption of user unlock password and fingerprint information. After a long time, large-scale actual operation test and monitoring, the system adopts AES encryption technology, successfully resisted multiple malicious attack attempts from the outside world, and there is no security incident caused by encryption vulnerabilities, effectively protecting the security and privacy of users' families (Javed et al. 2017) .

The AES algorithm has a high encryption speed and can encrypt a large amount of real-time data in the smart home system. In addition, it supports multiple key length options and can be flexibly configured according to the security level requirements of different smart home application scenarios. AES-128 provides sufficient security for common smart home device control command encryption scenarios. For sensitive information scenarios, such as home financial data storage and smart home system management permission control, we can choose AES-192 or AES-256 to further

enhance security (Setiawan 2021). When using the AES algorithm, if the key of a smart home device is stored in an insecure memory area, or is hacked during the key distribution process, the security of the entire system will be affected, even if the AES algorithm itself is highly secure (Fadhil et al. 2021). The long-term security of the AES algorithm is also facing certain issues because to the ongoing advancements in computer technology, particularly the slow emergence of quantum computing technology, even though there are no efficient quantum attack methods that can readily crack AES, but the cryptography community has been widely carried out related research and discussion, aimed at exploring strategies and methods to deal with potential threats in advance.

This paper focuses on smart home security in the IoT environment, with special emphasis on the application of enhanced AES algorithms. By introducing chaos theory and Logistic mapping technology, a new key generation method is proposed, which significantly improves the encryption and decryption efficiency of AES, and at the same time ensures high security. The three dimensional key generation mechanism (3DKGM), combined with chaotic encryption and Logistic mapping, not only speeds up encryption/decryption, but also reduces computing resource consumption, and improves the throughput by about 19.24% compared with traditional AES. The SeLPC method mentioned in the study further reduces power consumption and enhances security by reducing the encryption cycle of AES and using dynamic D-Box instead of S-Box. MLAES simplifies the process and improves efficiency by removing the MixColumns step and adding multi-stage XOR, shift loops, and SHA3-256. These improvements demonstrate that deploying lightweight AES variants in IoT environments can effectively address the challenge of balancing resource constraints with security needs (Rahman et al. 2022).

This paper discusses the application of AES-256 encryption algorithm in MQTT protocol, especially its role in ESP32-based smart home system. The paper points out that smart homes in the IoT environment face security risks such as illegal intrusion, information monitoring and leakage, DoS/DDoS attacks and forgery. The research uses the AES-256 encryption technique, which is renowned for its excellent security and low power consumption features, to safeguard the sent data in order to counter these attacks.

Using a 256-bit key length, the symmetric encryption technique AES-256 offers a high degree of security. When used with the MQTT protocol, it greatly enhances security, efficiency, and defence against brute-force assaults. Because MQTT simply uses the topic as a data filter and password authentication to connect to the MQTT broker, it has a poor level of security. The smart home system efficiently fends off possible security threats when paired with AES-256 encryption, which also improves the confidentiality and integrity of the data transfer process (Setiawan 2021).

This paper discusses the application of AES algorithm in IoT environment, especially a lightweight AES (LAES) algorithm is proposed to adapt to resource-constrained IoT devices. LAES is based on traditional AES improvements designed to enhance security and efficiency by simplifying certain operations and introducing chaotic systems. It is pointed out that LAES replaces ShiftRows with initial permutation (IP) and MixColumns with dynamic ShiftRows, and all the necessary components such as S-Box, key, permutation table and shift value depend on different chaotic system designs. This not only makes LAES faster than the original AES, but also validates its safety through NIST statistical testing. LAES's hardware implementation on Raspberry Pi 3 Model B shows that the algorithm consumes only 1.2% CPU usage and 26% memory, which has the advantage of high performance and low resource consumption. The design complexity of LAES increases the design difficulty because small changes in chaotic systems can lead to huge changes in output. Modified AES algorithms proposed in other studies, such as Chowdhury et al. 's reduced encryption cycle and Tsai et al.' s SeLPC method, improve efficiency and reduce power consumption, but these improvements may also introduce some implementation complexity (Fadhil et al. 2021).

In this paper, AES algorithm for smart home is discussed in depth, and an enhanced AES algorithm based on chaos theory and Logistic mapping technology is proposed to solve the vulnerability of traditional AES in key generation and improve the ability of real-time data protection. By combining 3D chaotic key generation mechanism (3DKGM), this method not only enhances the security of the key, but also improves the speed of encryption and decryption. Although traditional AES provides robust security features, its key generation method is vulnerable to appropriate key

cracking techniques, which brings risks to the protection of critical and real-time data. Enhancing AES with chaos and Logistic mapping can increase the difficulty of key generation and reduce the likelihood of cracking, which can maintain the integrity of important IoT data (Zhang et al. 2022).

The paper emphasizes the importance of single-key AES in protecting data transmitted over the network, especially when it comes to ensuring end-to-end data security. When it comes to image encryption, using single-key AES can cause parts with the same pixel value to remain the same value after encryption, making encrypted images vulnerable. In order to solve this problem and improve the security of images, the Multiple key Elliptic Curve Cryptography AES (MECC-AES) scheme is proposed, which combines elliptic curve cryptography (ECC) to generate multiple keys and AES algorithm for image encryption. The experimental findings demonstrate that MECC-AES is noticeably superior to the conventional single-key AES in terms of entropy, PSNR (peak signal-to-noise ratio), MSE (mean square error), and correlation. This technique improves the security and accuracy of the encrypted pictures. In addition, despite the larger image size used (12.1MB), MECC-AES 'encryption and decryption time is comparable to single-key AES at about 12 seconds. MECC-AES is suitable for IoT technologies that can or already use AES algorithms for security protection, and may not be suitable for IoT devices that require lightweight security solutions (Salim et al. 2021).

The AES (Advanced Encryption Standard) algorithm mentioned in the article is combined with ECC (Elliptic Curve Cryptography) -256r1 to form an optimized hybrid encryption framework in a smart home environment (Popoola et al. 2024). While addressing the insufficiency of current schemes to safeguard sensitive health data and showcasing robustness against new threats posed by quantum computing, this framework seeks to guarantee the confidentiality and security of data during transmission. According to studies, this combination offers unparalleled security, processing speed, and energy efficiency, outperforming current systems. When running in EAX mode, AES-128 has a processing speed of just 0.006 seconds (for both client and server) and an energy efficiency of 3.65 watts (client) and 95.4 watts (server), indicating that it provides a level of anti-quantum security comparable to AES-128.

The traditional AES mode can introduce significant processing time, delay, bandwidth usage, and power consumption overhead. In order to balance security and efficiency, the researchers analysed and selected AES-EAX as the preferred mode because it provides the necessary security features while maintaining computational efficiency. Although AES-GCM is also considered efficient, AES-EAX shows better performance in this particular area. An exhaustive experimental evaluation has demonstrated that ECC-AES is particularly suitable for IoT-powered smart Home Health environments (SHHE) because it mitigated the impact of device resource constraints while ensuring safety, performance, and adaptability (Popoola et al. 2024).

The Advanced Encryption Standard is widely used in IoT systems due to its high performance, low power consumption and low memory requirements, especially the short encryption time and significant avalanche effect on various platforms that make it the benchmark for data encryption applications. For applications with limited computing power and memory, such as smart homes, the complex implementation of AES requires large resources, limiting its efficiency. Although AES is more secure, when faced with large amounts of data, its encryption and decryption speed is not as fast as lighter standards such as SAFER+. While AES provides strong security, in resource-constrained and real-time IoT environments, researchers are exploring simplified AES or turning to secure and fast alternatives such as SAFER+ to optimize performance (Abdulla et al. 2021).

The article introduces an AES algorithm for smart home security, which combines ESP32 and biometric encryption technology to enhance the security of the IoT by generating a minute-changing key (one bio-key) and the owner's biometric information for authentication. After the data is encrypted by ESP32, it is sent to the local server via HTTP protocol and stored in a database built in Python language. The advantage of this method lies in its strong encryption and authentication mechanism, which can effectively resist various attacks and ensure the safe transmission of sensitive information. The complexity of AES algorithms and the high demand on computing resources can be a challenge in resource-limited IoT environments. The hardware components mentioned in this paper, such as Raspberry Pi, not only have high cost and small memory, but also put forward economic and technical limitations on practical

applications. While ensuring efficient safety, it is also necessary to consider the balance between cost effectiveness and technical implementation(Radhi & Hussain 2023).

Table 2.2 Summary of AES literature review

Author	Application scenario	Advantages	Disadvantages
Lujain S. Abdulla, Musaria K. Mahmood	Data encryption in IoT smart homes	Fast encryption speed, easy software implementation	Requires significant computational power and memory
Batool M. Radhi, Mohammed A. Hussain	Wi-Fi data transmission security in smart buildings	Commonly used standard, offers strong security protection	AES has high complexity and high requirements on computing resources
Fauzan Budi Setiawan, Magfirawaty	Secure communication in smart homes via MQTT protocol	Resistant to brute force attacks, efficient and low-power	MQTT protocol has inherent lower security
Olusogo Popoola, Marcos A Rodrigues, Jims Marchang	Hybrid encryption framework in smart home healthcare environment	High security, fast processing speed, low energy consumption	The traditional AES mode introduces a lot of processing time, delay, bandwidth usage, and power consumption overhead.
M. S. Fadhil et al.	Security in IoT networks	Modification of AES to fit IoT requirements	Modified algorithms may be more susceptible to being cracked
Ziaur Rahman, Xun Yi, Mustain Billah, Mousumi Sumi	Improving data security in IoT smart homes	Enhanced security through chaos theory, efficient key generation	Complexity in implementation may increase resource requirements
Meryam Saad Fadhil, Alaa Kadhim Farhan	Lightweight cryptography in IoT systems	Consumes less time in encryption/decryption, high throughput	Potential limitations in handling large-scale data or complex environments

### 2.2.2 DES

The symmetric encryption algorithm known as DES has a significant historical standing. It encrypts and decrypts using the same key, and its basic encryption mode is based on the Feistel network structure (Abdulla et al. 2021). In the encryption process, the plaintext data is divided into fixed-length blocks, usually 64 bits, of which 8 parity

check bits, the actual effective data is 56 bits, and then after multiple rounds of complex substitution and replacement operations, the final ciphertext is generated.

In the face of contemporary sophisticated computational resources, DES is susceptible to fatigue attacks due to its short 56-bit key length. Alzahrani through simulated attack experiments, it is found that using today's computing resources, the data encrypted by DES can be cracked in a short time. Research has shown that using specialized password cracking hardware or large-scale distributed computing systems, it is possible to traverse all possible key combinations to obtain plaintext information in a matter of hours or even less (Alzahrani 2023). This feature makes DES no longer suitable for smart home environments with high security requirements and strong data sensitivity. Due to the wide application of DES in the early encryption field, it may still exist in some old systems or devices with relatively low security requirements and special requirements for compatibility, but from the overall security trend, It has increasingly been supplanted by encryption techniques with greater security (Pirbhulal et al. 2016).

Lujain S. Abdulla et al. (Abdulla et al. 2021) point out that DES, as an early block cipher, has the advantage that it is a mature algorithm that has been extensively studied and verified, and has high reliability. Given current computer capability, the 56-bit DES key length makes it susceptible to brute force attacks. The encryption speed of DES is somewhat sluggish, particularly when handling huge volumes of data, and it is less effective than certain other more sophisticated algorithms like SAFER+. In order to overcome the security weaknesses of DES, triple DES (TDES) is sometimes used, that is, DES is used three times for encryption, but this further increases the computational burden and leads to lower efficiency.

Digital data security is ensured by the symmetric key encryption method known as the DES algorithm. DES functions by offering a means of encrypting data gathered from various sensors in a smart home or larger IoT setting in order to stop unwanted access and any data breaches. The advantages of the algorithm include its maturity and wide acceptance, as it has been used since the 1970s and has served as the basis for many subsequent encryption standards. DES's short 56-bit key length, which is no

longer secure in the face of modern computing power, is vulnerable to brute-force attacks. Although DES can be used as a basic encryption means, its security is not enough to protect against modern network threats. Due to its relatively low complexity, DES Encryption and decryption processes may be faster than newer algorithms such as AES, but this is not enough to make up for its shortcomings in security (Kavitha et al. 2022).

This paper (Tihanyi 2022) discusses the application of DES algorithm in smart home environment, especially for a DES based sensor device authentication algorithm. DES is a block cipher that converts 64-bit plain text to 64-bit ciphertext using a key that is actually 56 bits long. Although DES has four weak keys and six pairs of semi-weak keys, these specific keys make the encryption process produce predictable results. In theory, about 63.2% of DES keys should have at least one fixed point, and practical calculations show that DES does behave like a randomly selected permutation, supporting this theoretical hypothesis. DES is vulnerable to brute-force attacks due to its small key space ( $2^{56}$ ), which makes it no longer suitable for modern encryption needs.

Literature (Pirbhulal et al. 2016) points out that DES is not the optimal choice for applications in wireless sensor networks, because these network nodes usually have limited resources, and DES increases the burden of these nodes due to its relatively high energy consumption and complex key scheduling mechanism. According to the data provided in this paper, the energy consumption of DES is 2.08 microjoules/byte, which is higher than that of other symmetric encryption algorithms. Although DES is designed for 32-bit processors to achieve high efficiency, its high key setup cost makes it possible to allocate the key setup cost to the low encryption cost in application scenarios where the key does not change frequently, but for wireless sensor nodes in the IoT smart home environment, this feature may not be ideal. Because these devices often need to exchange data frequently and undergo key updates to ensure secure communication, this leads to additional energy consumption.

Table 2.3 Summary of DES literature review

Author	Application scenario	Advantages	Disadvantages
Addanki Kavitha <sup>1</sup> , B Srinivasa Rao <sup>2</sup> , Dr Nikhat Akhtar et al.	Healthcare system data security, IoT applications	Symmetric-key algorithm, prevents potential data breaches	Small key size makes it susceptible to brute-force assaults, hence it is no longer regarded as secure.
Norbert Tihanyi et al.	Cracking IoT-based Sensor Device Authentication Algorithm	Demonstrates fixed points for non-weak keys, can be used for cryptanalysis	DES with a 56-bit key is not secure; fixed points can be exploited for hacking
Sandeep Pirbhulal et al.	Securing data transmission in WSNs integrated with IoT	Efficient for low-power nodes, supports large numbers of devices	Not suitable for implementing asymmetric algorithms due to limited computational power of sensor nodes
Lujain S. Abdulla et al.	Evaluation of symmetric key ciphers for IoT smart homes	Provides basis for selecting appropriate encryption for resource-constrained environments	Does not specify unique advantages or disadvantages of DES

### 2.2.3 3DES

3DES is an improved version proposed to enhance the security of DES. It improves the encryption strength by performing three DES encryption operations on the data. Specifically, 3DES can adopt different encryption modes, the common one is EDE (Encryption-Decryption-encryption) mode (Mamvong 2023). In this mode, key K1 is used to encrypt the data first, the encryption result is then decrypted using key K2, and the decryption result is then encrypted using key K3. In general, K1 and K3 can be the same or different, if  $K1 = K3$ , it is called 2-key 3DES, otherwise it is 3-key 3DES.

3DES makes up for the lack of security of DES to a certain extent, and its encryption strength is significantly improved compared with DES. In some smart home systems, especially those that need to take into account certain compatibility and have high requirements for security, such as some early deployment but still in use and involving important data transmission of smart home device network upgrade, 3DES is still used (Hasan et al. 2021). The relatively complex encryption process of 3DES results in high computational complexity and high resource consumption. In resource-constrained IoT devices, this can have an impact on device performance such as reduced

response speed and increased power consumption. Compared to modern lightweight encryption algorithms designed specifically for IoT, 3DES is gradually showing its limitations in resource-constrained device application scenarios in the smart home (Santa & Ariza 2019). The use of 3DES in smart homes is rapidly dwindling as a result of ongoing advancements in IoT technology and the introduction of new encryption algorithms; it is now more commonly recognized as a transitional encryption solution.

The 3DES algorithm, as an improvement over the standard DES algorithm, enhances security by repeating the single DES process three times and using two or three different 56-bit keys, thus extending the effective key length to a maximum of 168 bits. The purpose of this multilevel encryption approach is to increase the packet algorithm's security by repeating encrypting the same message block using different keys. The triple encryption/decryption of 3DES also brings a significant increase in resource requirements and delay, which limits its applicability in real-time applications or small devices, which is not in line with the reality of device constraints in IoT. Although 3DES provides higher security than the original DES, it inherits many shortcomings of the standard DES algorithm, especially in the IoT environment with limited processing power, due to high computing costs, slow speed and large demand for resources, 3DES is not a particularly viable choice (Mamvong 2023).

It is pointed out that 3DES algorithm is one of the candidate encryption algorithms for secure information transmission between embedded system and PC, especially for smart home and smart sensor networks. When running on the Python programming language and executed in a PC environment, 3DES shows a much longer initialization time than the AES family of algorithms, almost three times the average time of the other three AES algorithms, making 3DES less cost-effective between performance and security than the AES family of algorithms. When comparing the software implementation on the PC with the hardware-accelerated implementation of the embedded system based on the PSoC 6 microcontroller, the execution time of 3DES is significantly longer than the different versions of AES. The implementation of 3DES in embedded systems still maintains a certain degree of security, and its execution time and decryption time are significantly improved compared to the PC environment, but it is still not as efficient as versions such as AES128 (Santa & Ariza 2019).

When discussing encryption algorithms suitable for smart home environments, the article mentions that the 3DES algorithm enhances security by using the DES algorithm to encrypt data three times, where two or three different keys can be used. This approach makes 3DES more difficult to crack than its predecessor, as it takes more time and computational resources for an attacker to try all possible key combinations. One of the main drawbacks of 3DES is its relatively slow execution speed, mainly due to the high computational overhead of having to run three rounds of encryption. For resource-constrained IoT devices, this delay can be unacceptable, especially in application scenarios that require a fast response (Zhang & Wang 2024).

Table 2.4 Summary of 3DES literature review

Author	Application scenario	Advantages	Disadvantages
Ricardo Martínez Santa, Holman Montiel Ariza	Embedded systems for secure information transmission between PC and embedded system.	Provides a higher level of security compared to DES.	Slower execution time than AES128 on PSoC 6 embedded system; less efficient in terms of computational resources.
Mohammad Kamrul Hasan et al.	Protecting IoT applications from guessing attacks using lightweight cryptographic algorithms.	Enhances security for data encryption in complex IoT environments.	Not specifically mentioned as advantageous or disadvantageous for 3DES in this context; generally slower than some alternatives.
Sandeep Pirbhulal et al.	The data transmission of online transactions, encryption and decryption of sensitive data.	Addressing client-side encryption and safe provisioning, two security issues with limited IoT devices.	Increases key space and provides enhanced security against brute force attacks over single DES.
Limin Zhang, Li Wang	Hybrid encryption approach combining symmetric Blowfish with asymmetric elliptic curves for efficient and secure data transmission in IoT devices.	Not explicitly discussed in the context of this hybrid approach.	The paper does not provide specific advantages or disadvantages for 3DES within its scope.

## 2.3 ASYMMETRIC ENCRYPTION ALGORITHM

### 2.3.1 RSA

The encryption method of big prime number decomposition serves as the foundation for the RSA algorithm., which gives it strong anti-attack ability from the theory level. As long as the large prime number is large enough, the attacker can hardly obtain the private key through decomposition under the existing computing power, thus ensuring the security of the encrypted information (Bagha et al. 2020). The public key can be publicly distributed, which greatly facilitates the establishment of secure communication channels between devices, without the need to share the key in advance like symmetric encryption, and reduces the difficulty and risk of key management (Zahan et al. 2020). From the point of view of mathematical principles, RSA algorithm first needs to generate two large prime numbers  $p$  and  $q$ , Calculate their product  $n=p \times q$ . Then, according to  $n$ , we determine an integer 'e' that is prime to  $\phi(n)=(p-1)(q-1)$ , and the public key is  $(e,n)$ . The private key  $d$  is an integer that satisfies  $e \times d \equiv 1 \pmod{\phi(n)}$ . For plaintext  $m$  ( $m < n$ ), the encryption process is  $c=me \pmod{n}$  and The decryption process is referred to equation 2.1 below,

$$m = cd \pmod{n} \quad \dots(2.1)$$

The enormous computational cost of the RSA method, particularly when encrypting and decrypting huge amounts of data, will surely result in significant performance bottlenecks for IoT devices with limited resources. With the steady advancement of quantum computing technologies and the quick increase in processing power, the Shor algorithm in quantum computing has been proved to be able to decompose large integers in polynomial time. Once the quantum computer is practical, the encryption system of RSA algorithm will face the risk of collapse (Popoola et al.). In the identity authentication architecture of smart home devices proposed by Santa, RSA algorithm is cleverly used to build a trust bridge between devices and servers. Upon connecting to a device, the smart home hub server encrypts certain challenge data using the device's public key before transmitting it to the device. Following receipt of the encrypted data, the device decrypts it using its own private key and sends the

decryption result or the response produced in accordance with the decryption result back to the server. By confirming that the answer is accurate, the server confirms the legitimacy of the device identify (Santa & Ariza 2019).

To increase security in the IoT environment, this article presents an improved user authentication technique created especially for smart homes. It is based on the RSA algorithm and combines biometrics. In the RSA-B-ASH-S scheme, the security protocol of RSA is inherited and a third verification layer - biometric verification such as fingerprint or facial recognition - is added using the capabilities of the smartphone, thus achieving three-factor authentication. This solution not only satisfies all traditional security features, but also emphasizes perfect forward secrecy, that is, even if the private key is leaked for a long time, the previously transmitted encrypted information will not be cracked. Although RSA provides both public and private key pairs, it is more resource-intensive than symmetric encryption from a computational performance perspective. Since the scheme relies primarily on asymmetric encryption, compute-intensive problems can be encountered during decryption, especially when dealing with large number operations (Bagha et al. 2020).

The use of the RSA algorithm in a smart home setting is covered in this study., especially in the inverter system connected to the power grid. This paper points out that in the proposed system, RSA algorithm is mainly used to ensure the security of data transmission in IoT environment and prevent information leakage and tampering. The RSA protocol is implemented through Python programming to ensure the security of data communication from the inverter to the client, including the encryption and decryption process of data packets. The experimental findings demonstrate that even in the event that RSA encryption attacks the network, the attacker cannot obtain valid information because the data is encrypted. In order to provide high-strength security, RSA requires large key lengths, which slows down the performance of the system. The solution proposed in this paper simplifies the configuration by deploying RSA algorithm in the application layer instead of the data link layer, and can run on ordinary single chip microcomputer, reducing the difficulty and cost of implementation (Ahmed & Khan 2021).

The paper's improved RSA method seeks to increase data security in the context of IoT smart homes., increase the complexity of factorization by introducing five prime numbers instead of the traditional two, and replace the original key and module with a pseudo-public key and pseudo-module. The algorithm adds additional steps in the key generation phase, including calculating a pseudo-public index "f" as the product of the public key index "e" and a pseudo-private index "g" as the division result of the private key "d", in order to hide the actual public key "e", thus improving the ability to resist factorization attacks. Due to the use of more prime numbers and more complex factorization, the time cost of key generation, encryption and decryption increases compared with the original RSA algorithm. Taking into account the principle of security first, a longer encryption and decryption time is acceptable. The algorithm takes less time in the encryption and decryption process than the improved RSA algorithm based on Ivy et al. (2012), but more time than the original RSA algorithm by Rivest et al. (1978) (Ullah et al. 2022).

When discussing RSA algorithm for smart home security framework, an innovative security measure, namely RP2-RSA algorithm, is introduced in the literature. The algorithm is specifically designed for IoT -enabled healthcare systems. Traditional RSA encryption technology relies on public and private key pairs to ensure the security of data transmission, in the BBNSF framework proposed in this paper, the authors introduce the concept of inverted public and private keys to provide a higher level of security. RP2-RSA algorithm not only enhances the security of data encryption and decryption process, but also realizes the security level of 96.123%, which significantly improves the security of user authentication and medical data storage. The optimized RP2-RSA algorithm is introduced, which is an upgraded version based on the traditional RSA encryption method, and further enhances the encryption strength and efficiency. In this way, you can not only effectively defend against external hackers, but also prevent internal potential security threats, such as unauthorized access attempts from inside your home network. The application of RP2-RSA algorithm enables medical information in smart home systems to maintain a high level of security and integrity even in the face of complex man-in-the-middle attacks. (Kumar et al. 2022).

Table 2.5 Summary of RSA literature review

Author	Application scenario	Advantages	Disadvantages
Nasim Ahmed, Md. Ziaur Rahman Khan	IoT-based grid-connected inverter	Ensures data security and connectivity, reduces harmonics	Implementation complexity, cybersecurity concerns
Mohit Kumar et al.	IoT-enabled healthcare systems with blockchain	Combines RP2-RSA for secure framework, ASR-ANN technique	Complexity of integrating blockchain with RSA, resource consumption
Bashart Ullah et al.	Data security in the IoT	Enhanced security using five prime numbers, uses phony modules/exponents	Increased key generation time, computational cost
Amir Mohammadi Bagha	Smart-homes user authentication via smartphone	Three-factor authentication, forward secrecy, user anonymity	Potential incompatibility with conventional methods, complex setup

### 2.3.2 ECC

The elliptic curve theory serves as the foundation for the asymmetric encryption method known as ECC (Elliptic Curve Encryption). Elliptic curve equations are often specified over finite fields, most frequently  $GF(p)$  (where  $p$  is a prime number) or  $GF(2^m)$  (where  $m$  is a positive integer). The elliptic curve equation on the finite field  $GF(p)$  is usually expressed as  $y^2 \equiv x^3 + ax + b \pmod{p}$ , where  $a, b \in GF(p)$  and  $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ . This condition is to ensure that the curve is smooth and free of singularities.

In a smart home environment, the communication between low-power sensors and gateways has strict requirements on security and resource consumption. Because of the small amount of computation and the relatively fast speed of key generation and encryption and decryption, ECC algorithm can operate effectively on sensor devices with limited resources and ensure the security of data transmission (Lohachab 2018). In some intelligent environmental monitoring systems, temperature, humidity, air quality and other sensors need to securely transmit the collected data to the gateway or cloud platform. Popoola et al. show that under the same security level, the encryption and decryption time of ECC algorithm on a smart sensor node is shortened by about 74% compared with RSA, and the power consumption is reduced by 67%, which enables the

sensor to run stably for a long time under battery power supply and ensure data security (Popoola et al. 2024). In the remote control scenario of smart home devices, such as users remotely control smart appliances through mobile phone applications, ECC can be used to establish a secure channel between the device and the mobile phone to ensure the safe transmission of control instructions and prevent instructions from being hijacked or tampered with.

ECC is well-liked in resource-constrained IoT smart home scenarios because to its lightweight security features. The foundation of ECC is the elliptic curve discrete logarithm problem (ECDLP), which asserts that even if  $P$  and  $Q$  are known, it is extremely challenging to compute  $k$  if  $Q=kP$ , where  $P$  and  $Q$  are points on an elliptic curve and  $k$  is a scalar. This feature ensures the security of the data, while ECC only requires 384-bit keys to provide the same highest level of security as RSA with 7680-bit keys. This makes ECC ideal for smart devices with limited storage space and low computing power, as they are able to perform encryption operations with less energy consumption. ECC also supports many types of elliptic curve forms that are defined over different finite fields and rely on fundamental operations such as point addition or scalar point multiplication on elliptic curves. With the development of quantum computers, quantum computers can easily crack ECC encryption algorithms based on mathematical puzzles (Lohachab 2018).

The usage of ECC in the IoT is covered in this study, with a focus on digital signature use cases in smart home settings. According to the article, ECC is a small key size, lightweight encryption technique that outperforms RSA-driven technology in terms of strength at the same security level. ECC also provides better security and confidentiality in RFID systems than non-ECC secure architectures, as its point addition and replication processes are faster, and it is superior to RSA in generating digital signatures, but slightly slower in verifying digital signatures. Although RSA may be a better choice in cases where message verification is more frequent than signature generation, ECC is more practical and efficient for handheld applications given the advantages of computing power and key ratio. According to the NIST specification, ECC provides a significant security boost, especially when security requirements are more stringent and processor performance is more powerful, and to remain ahead of

other implementations, a slight increase in key length is all that is needed. The goal of this paper's suggested ECC-based digital signature mechanism is to increase IoT system security, especially for the security challenges of smart home access control systems. The experimental evaluation proves that ECC has better performance than RSA method in selected IoT use cases (Zahan et al. 2020).

It is pointed out that the proposed scheme realizes the mutual authentication among users, gateways and device nodes by using ECC, hash function and bit-by-bit or operation cryptographic primitives, and successfully establishes the session key. This scheme not only ensures the anonymity of the user, the forward security of the session key, but also can resist all the known attacks shown in this paper. In addition, the performance comparison compared with the latest designs shows that this solution is competitive in terms of storage, network communication and computing costs, especially in terms of users and gateways, requiring only 352 bits and 320 bits. The scheme offers some improvements in safety and efficiency, but it also inherits some limitations. Earlier schemes often ignored factors such as forward security, mutual authentication and user anonymity, which may lead to the security of session keys being threatened. In this paper, a new idea of key exchange is used to calculate the session key based on ECC after four interactions between three entities, which ensures the security of the session key, protects the privacy and maintains the effective performance of the scheme (Zou et al. 2021).

This paper (Islam et al. 2021) reviews the secure communication protocols based on ECC for smart home. With the development of IoT technology, the interconnection and data sharing between smart home devices become possible. This paper proposes a lightweight ECC encryption scheme, which not only ensures the security requirements of authentication, confidentiality, integrity and key negotiation, but also has the ability to resist common network attacks. The modified ECC algorithm used in the study enables efficient encryption with a smaller key size, which makes cracking the system an exponential time challenge for the attacker. The paper also compares the existing methods, showing that the proposed scheme has significant advantages in terms of storage and communication costs, and is suitable for application

to low-power devices Raspberry PI, and future work will be extended to implement the protocol on Arduino or lower power devices.

In this paper (Nyangaresi 2021), the author points out that the registry (RA) selects a particular form of elliptic curve based on a finite field and selects a generator point to generate the private and public keys of the system. The authors introduce long secret keys and hash functions, and ensure the transparency of the system by publishing these parameters publicly. During the registration process of users and smart devices (SD), the authors used ECC to calculate and verify security parameters to ensure the legitimacy of each user. When users log in, they enter their identity and password and generate a random number nonce, which is combined with ECC encryption or signature to ensure the authenticity and integrity of the data. Compared with traditional public key encryption algorithms, ECC provides a more efficient encryption method, which is especially suitable for resource-constrained devices in smart home environments. They also emphasize that by not relying on timestamps or validation tables, the protocol reduces the risk of clock asynchronization and prevents the risk of authentication data being breached. The security evaluation shows that the protocol can resist many kinds of attacks.

This study (Wang et al. 2022) suggests a strong and private three-factor authentication system based on ECC that is tailored for smart home settings. The purpose of this approach is to address the issues with node capture attacks, offline password guessing attacks, and the inability to ensure forward security that plagued Yu et al.'s three-factor anonymous authentication scheme. It is pointed out that traditional cryptographic systems cannot provide lightweight authentication due to the limited computing resources of devices in smart home networks. On the other hand, because ECC uses less bit space and CPU power, it is a very effective solution that is perfect for smart homes. The suggested authentication scheme combines ECC, hash function, and XOR operation to achieve session key forward security, user anonymity, and untraceability. It also conducts formal and informal security analysis, demonstrating its security properties with Burrows-Abadi-Needham logic and Scyther simulation tools. and highlight the benefits of efficiency in order to attain a fine balance between security and performance that is better suited for the real smart home setting.

In the simplified network topology constructed by the experiment, the ECC based security mechanism is deployed with commercial off-the-shelf devices such as Raspberry Pi, and the results show that the security measure can increase the confidentiality of communication from 29.35% to 70.65%. It shows that ECC can effectively increase the information confidentiality in the process of MQTT protocol publication/subscription communication. Although ECC provides strong security and ensures the security of each data transfer by personalizing the key generation process according to the MAC address of the device, this process also increases the computational burden, especially during the key generation phase, and can cause performance bottlenecks for devices with limited computing resources. While the implementation of ECC improves data confidentiality and integrity, it does not change the MQTT protocol itself, meaning that other standard security features such as SSL/TLS still need to be combined to secure the transport layer (Yusoff et al. 2022).

Table 2.6 Summary of ECC literature review

Author	Application scenario	Advantages	Disadvantages
Zou et al.	Smart home user authentication, WSNs	High security (anonymity, forward secrecy), efficient communication/calculation	Large database for storing information could be inefficient
Koziel, Azarderakhsh, etc.	IoT security, Quantum Key Distribution	Lightweight, secure against quantum attacks, mutual authentication	Implementation complexity on constrained devices
Azrin Zahan et al.	IoT security, smart-door system	Smaller key size, more strength than RSA, lightweight	Higher complexity and costs for RSA-based systems, power requirements
Zainatul Yusoff et al.	MQTT-based smart home systems	Lightweight, efficient security, small key sizes	Implementation complexity
Xiong Wang et al.	Smart home environments	Resistance to various attacks, user anonymity, session key forward security	Higher computational costs than some alternatives
Vincent Omollo Nyangaresi et al.	Smart homes	Low communication and computation overheads, robust against common attacks	Requires certificate authority, potential single point of failure
Towhidul Islam, Ravina Akter Youki, Bushra Rafia Chowdhury	Lightweight cryptographic scheme for securing IoT device communication in Smart Homes	Reduces processing and data exchange time; is effective with reduced key sizes, and is resistant to cyberattacks.	Computationally complex compared to traditional algorithms, may not be suitable for all resource-constrained devices

## 2.4 OTHER ENCRYPTION ALGORITHM

### 2.4.1 ASCON

ASCON has a unique authentication and encryption structure, which can achieve data encryption and integrity verification functions at the same time. The underlying mathematics involves elaborate wheel functions and key arrangements (Rahul et al. 2024). Through a series of complex nonlinear transformation and linear diffusion operations, the round function processes the data and the key in multiple rounds, so as to achieve a high degree of confusion and diffusion effect, and ensure the security of encryption. In terms of key arrangement, ASCON uses a specific algorithm to generate the sub-keys required for each round of encryption, which have complex dependencies, further enhancing the security of the encryption system (Ul Islam et al. 2024). In the initial stage of data encryption, the input data will first undergo specific XOR operations with the initial key, and then enter multiple rounds of round function processing, and the output of each round of round function will be used as one of the inputs of the next round, while combining with the corresponding sub-keys to transform until the entire encryption process is completed.

ASCON has proven excellent suitability for resource-constrained devices, where some small smart sensor networks have very limited computing power and storage resources. The ASCON algorithm can run efficiently on such devices to provide security protection for environmental data (such as temperature, humidity, and air quality) collected by sensors, preventing data from being stolen or tampered with during transmission (Magyari & Chen 2024). The main function of smart tags is to store and transmit a small amount of identifying information, and ASCON can ensure the security of this information without taking up too many resources. The ASCON algorithm's performance tests on a variety of low-power IoT devices show that it performs well in terms of storage space occupancy and computing resource consumption, and also shows high security in cryptanalysis tests, which is expected to become a key solution for lightweight encryption in the future smart home.

In terms of compatibility with existing encryption infrastructure, ASCON may need to adapt and transform the existing system to some extent due to its relatively

novel algorithm design, which undoubtedly increases the difficulty of promotion (Alharbi et al. 2024).

In particular, the article describes the implementation of the Ascon algorithm on resource-constrained IoT devices in the smart home. With the development of Quantum computers, traditional encryption methods face the risk of being cracked, so it is necessary to introduce PQC, which is resistant to quantum attacks, to ensure the security of smart home networks. Ascon-Sign, as a lightweight PQC signature scheme, is first implemented on resource-constrained FPGA for node verification in wireless sensor networks. This solution not only achieves twice the speed of comparable area-constrained devices, but also reduces the operating power consumption by 33%, and proposes modifications to the Ascon-Sign specification to reduce processing time and memory requirements (Magyari & Chen 2024).

ASCON is used in conjunction with hash functions in a new security authentication framework, ESCI-AKA, which aims to establish a secure channel between user devices and smart homes to ensure the security of data exchange over the public Internet. With ASCON, ESCI-AKA not only simplifies the login and password change process with a single encryption operation, but its security is supported by Random Prophecy Model (ROM) analysis and verified by the Scyther tool. Performance evaluations show that compared with other security mechanisms, ESCI-AKA reduces computing and communication costs by 61.01% to 86.27% and 40.51% to 65.94%, respectively, demonstrating significant cost effectiveness. The ASCON encryption technique is used to encrypt every message sent during the AKA phase. This ensures the security and effectiveness of the communication between devices in the smart home environment by reducing the security benefit of breaking the AEAD scheme, according to ASCON's security definition (Alasmary & Tanveer 2023).

The advantage of Ascon is that it is designed for lightweight IoT devices, providing high performance and low power consumption, which is ideal for wireless sensor nodes in smart home environments. It takes advantage of a simple hardware architecture to avoid energy-intensive resources such as DSPs and large amounts of BRAM, while supporting AEAD (authentication encryption with associated data) and

signature models, and integration via the AXI-Stream bus simplifies the implementation of other design solutions. In addition, Ascon-Sign combines with SPHINCS+, the only NIST-approved PQC signature mechanism based on stateless memory hashing, to provide a non-interactive digital signature capability that guarantees the authenticity of the identity of the message sender. While AScon-Sign performs well in specific application scenarios, it may not be optimal for applications that require extremely high throughput, as increasing throughput often means increasing resource consumption, which is contrary to the miniaturization and energy saving goals AScon pursues (Magyari & Chen 2024).

The evaluation of ASCON algorithms in IoT security applications is mentioned in the literature, particularly in terms of resistance to side channel attacks (SCA) in lightweight implementation environments. The literature review noted that successful statistical invalidation fault Analysis (SIFA) attacks were demonstrated on the Artix-7 field Programmable Gate Array (FPGA) for lightweight implementations by using voltage burrs to attack ASCON's supply pins. This indicates the vulnerability of ASCON to both passive and active SCA attacks (Sarker et al.). The NIST Lightweight Cryptography Standardization Competition's second round authentication password, ASCON, is intended to offer effective security and works well on systems with little resources (Rana et al. 2024).

It is pointed out that Ascon adopts authentication encryption and Associated Data operation, and uses sponge structure to support encryption/decryption and hashing operation, which meets the requirements of low power consumption and small area occupation of smart home devices while ensuring efficient performance. The design of lookup table (LUT6) based replacement boxes as part of a cipher replacement block for processing medical data such as electrocardiogram signals shows that Ascon is not only suitable for smart homes, but also extends to secure communication of medical-related IoMT devices. The algorithm is optimized through iterative design methods and finite state machine control to accommodate different types of 7 Series FPGA devices, resulting in increased operating frequency and reduced power consumption (Alharbi et al. 2024).

Table 2.7 Summary of ASCON literature review

Author	Application scenario	Advantages	Disadvantages
Alasmary, H.et al.	Smart Home Environment for secure communication using Authenticated Key Agreement Framework	Lightweight cryptography, robust security against various attacks, efficient in resource-constrained environments	Easy to be limited by the extreme resources of devices in a smart home environment
Magyari, A.; Chen, Y.	Resource-constrained FPGAs for wireless sensor networks verification, including weather detection systems	Fast implementation, reduced power consumption, shortened processing time and lower memory requirements	Increase in BRAM usage and static/active power if more complex operations are performed
Adel R. Alharbi et al.	IoT, embedded systems, mobile technologies, IoMT, securing communications in small-scale devices like wearable tech, smart home devices, sensors in industrial and healthcare settings, smart grid infrastructure	Lightweight, robust cryptographic solution; Low-area footprint; Minimum power consumption; Reasonable throughput; High operating frequency; Unified architecture for Ascon-128 and Ascon-128a variants	Interface adjustment is required to adapt to different application scenarios
Arun Kumar Rana et al.	Smart home applications, IoT-based environments	Secure sessions start faster with SHA 224; Energy-efficient hash function for CoAP protocol; Enhances security and speed	Focuses more on CoAP protocol rather than ASCON directly

#### 2.4.2 SHA-256

SHA-256 belongs to the SHA-2 family of hash functions, whose hash value is 256 bits long. Its calculation process consists of several key steps. The first is data padding, filling the original message so that its length meets a specific requirement. In SHA-256, the message is first appended with a '1' and then filled with several '0's' until the message length module 512 equals 448. The message is then extended to divide the filled message into 512-bit blocks, and each block is further processed to generate a series of words that will be used in subsequent compression functions. The compression function is the core calculation part of SHA-256, which iterates on the 512-bit message block and the 256-bit intermediate hash value, each round involves the use of complex logical

functions and constants, and after 64 rounds of iteration, the final 256-bit hash value is obtained (Lukesh et al. 2024).

During the startup process of smart devices, SHA-256 can verify firmware integrity. The device determines the locally stored firmware's SHA-256 hash value after turning on. The firmware's matching SHA-256 hash value will also be made public by the manufacturer upon the firmware's formal release. The official hash value and the one determined at device initialization are compared. The device can start and operate correctly if the two are consistent, which means that the firmware has not been altered during transfer or storage. If the hash values do not match, it most likely means that the firmware has suffered a malicious modification, and the device may stop booting and alert the user or administrator (Fotohi & Aliee 2021). This system can successfully stop malicious firmware from being installed, guarantee the device's regular operation, and protect user data.

SHA-256 has significant features. Its hash value is irreversible, that is, it is almost impossible to derive the original message from a given hash value. This feature ensures that even if an attacker obtains the hash value, it is difficult to restore the original sensitive information. Since any slight alteration to the data would cause a significant change in the hash value, SHA-256 is very collision resistant, making it very difficult to identify two distinct messages (Santos Jr et al. 2024). But with the continuous development of quantum computing technology, although there is no effective quantum attack method to crack SHA-256, its security is also constantly being studied and evaluated by the cryptography community.

The main disadvantages of SHA-256 are its relatively high computational overhead and long startup time, which is particularly evident on resource-constrained IoT devices, as these devices typically have low processing power and limited energy supply. Although SHA-256 can offer a high degree of security for smart homes, researchers are also looking into alternatives like SHA-224 for real-world applications to balance security and efficiency, which can establish secure sessions faster while ensuring sufficient security, and is better suited for low-power IoT devices. The article also mentions the adoption of Chacha stream ciphers as a measure to further improve

the security of communication between IoT devices to address the power consumption and performance challenges of traditional encryption methods (Rana et al. 2024).

Through hardware implementations like FPGA technology, SHA-256 algorithms improve the security and functionality of IoT devices in the context of smart homes. An FPGA-based SHA-256 hardware implementation strategy is suggested for IoT devices in the blockchain context, as noted in the literature, which utilizes the cluster core of parallel execution to improve processing power and save power consumption. This implementation not only improves the data processing speed, but also significantly reduces the dynamic power consumption, which is around 1,000 times lower than what was previously documented in the literature, making it especially appropriate for real-world issues with IoT devices. Furthermore, the SHA-256 algorithm is extensively utilized in several security procedures and technologies, such as proof of work, Merkle trees, HMAC, PKI, TLS, IPSec, etc., and is critical to ensuring the security of communication in a smart home environment. SHA-256 For resource-constrained IoT devices, how to optimize power consumption and hardware usage efficiency while maintaining high performance is an ongoing research challenge. The proposed design achieves a throughput of approximately 1.4 Gbps using a Xilinx Virtex 6 FPGA (Santos Jr et al. 2024).

SHA-256 technology can protect smart home devices from code penetration, and the use of the technology in a security framework can improve the security of the system against external cyber attacks, especially brute-force attacks. The article highlights how SHA-256 strengthens security measures by transferring cryptographic tokens to multiple devices. It also points to the ability to build private servers and apply advanced security features such as SHA-256 to avoid the costs associated with long-term use of public cloud services, as well as the ability to customize servers. From the literature, it can be seen that for resource-constrained devices, the use of high-strength hash functions such as SHA-256 can present performance challenges, because such functions are computation-intensive and may affect the response speed or power consumption of the device (Phuc et al. 2020).

The identity-based signature and SHA-256 algorithm become the key strategies to improve the scalability and security of the system. In order to ensure effective authentication and secure communication between devices, as well as data integrity and immutability, it is particularly important to adopt device identity as public key. This approach not only simplifies the complex certificate management process in traditional public key infrastructure (PKI), but also significantly reduces the overhead of each node resource, thereby improving the throughput of the entire network. Under this framework, SHA-256, as a cryptographic hash function, provides a strong guarantee for data integrity. Its core advantage is that any slight change to the input data will result in a large change in the output hash value, which makes it extremely effective at detecting data tampering. Since the generation of digital signatures depends on the correct private key, only the entity with the corresponding private key can create a legitimate signature, further strengthening the security of information. This mechanism supports non-repudiation, that is, once the message has been signed and sent by the sender, the sender cannot deny the action. As the number of network nodes increases, it becomes more and more difficult to perform complex hashing operations, which challenges the scalability of the system. These issues can be mitigated to some extent by well-designed data distribution and verification strategies, as well as by leveraging technologies such as edge computing to reduce the burden on central nodes (Fotohi & Aliee 2021).

The article (Ullah et al. 2023) mentioned that SHA-256 is mainly used in the encryption algorithm of MQTT protocol in the smart home environment. SHA-256 is used to provide security for data sent from the Raspberry Pi to the smart gateway, ensuring secure data transmission within the smart home system. When comparing SHA-256 with two hashing algorithms, SHA-3 and Keccak-256, SHA-256 failed to provide the maximum Avalanche Effect, which is the ability of small changes in inputs to cause large changes in outputs. This shows that SHA-256's output changes slightly in the face of changes in input bits, showing a significant difference of only 5%. In addition, in the case of fixed keys and flat text variation, the MQTT encryption algorithm using SHA-256 performs poorly compared to SHA-3, which is used by the Hyperledger Fabric blockchain. Although SHA-256 plays a role in smart home safety, experimental results show that it is not as robust as SHA-3 and Keccak-256 in some respects, especially in achieving adequate avalanche effects.

Table 2.8 Summary of SHA-256 literature review

Author	Application scenario	Advantages	Disadvantages
Arun Kumar Rana et al.	Secure sessions between transmitter and recipient nodes in IoT applications using CoAP protocol	Faster secure session start (SHA-224), Energy-efficient, Increases security and speed	Start a secure session slower than SHA-224
Tran Anh Khoa et al.	Authentication process in smart home devices over a wireless network	Provides strong security framework, Prevents unauthorized access to smart devices, Protects against cyber attacks	The implementation has strong complexity
Reza Fotohi, Fereidoon Shams Alice	Block transmission in a blockchain for IoT devices	Secure information exchange, Improved scalability, Reduced complexity	Consumes energy for block validation, Not suitable for all real-time applications due to time spent on validation
Abrar Ullah et al.	Comparing hash algorithms (SHA-3, Keccak-256, SHA-256) in smart home security	Better performance with fixed keys and plain text variation compared to MQTT	Unable to provide the maximum Avalanche Effect; Performed worse than SHA-3 and Keccak-256
Le Mai Bao Nhu et al.	Ensuring secure authentication for smart home control system components (hardware, server, web app)	Enhances security efficiency, Safeguards system information, Makes smart home safe and secure	An efficient security framework that includes SHA-256 can add complexity and cost to the system

## CHAPTER III

### METHODOLOGY

#### 3.1 INTRODUCTION

To evaluate the performance of different encryption algorithms in the IoT smart home. In this paper, a set of key indicators are analysed, including quantitative parameters such as throughput, delay, accuracy and packet loss rate. These metrics were chosen because they provide a comprehensive assessment of the algorithm's effectiveness, accuracy, and utility in resource-constrained smart home environments. Throughput reflects the amount of data that an encryption algorithm can process per unit time, which is critical to ensuring fast information exchange between smart home devices. The delay metric measures the time required for the entire process from data encryption to decryption, and low delay is the basis for achieving an immediate response to user instructions. The ability of the encryption and decryption processes to preserve the data's integrity without error is what accuracy is all about, and any mistake might lead to an intelligent system breakdown. The packet loss rate refers to the proportion of packets that fail to be delivered successfully in the process of network transmission, which is directly related to the reliability and stability of communication.

Given that smart homes are often characterized by limited computing power and storage space, a rigorous evaluation of each of these performance indicators becomes particularly important. This aids in identifying the encryption technique that works best in a certain application environment, but also has a significance for optimizing the overall security and user experience of smart home systems. Through such an assessment, we can provide guidance to smart home device manufacturers and service providers to choose the most appropriate encryption technology to safeguard users' privacy and data security.

### 3.1.1 Simulation Environment

A MATLAB simulation was created to demonstrate the reaction times of the three cryptographic algorithms under consideration (AES, RSA, and ECC) for an IoT smart house. Key performance characteristics related to rate, delay, accuracy, and packet loss are captured by this simulation under various network setup parameters, including the number of IoT devices, encryption size, and communication hop. To evaluate the performance of AES, RSA, and ECC under different settings, produce values for the selected metrics displayed in the following series of graphs using theoretical models specifically created for IoT systems. The performance evaluation of this study is based on the MATLAB code. The script focuses on four important performance indicators while simulating the behaviour of three cryptographic algorithms—AES, RSA, and ECC—under various IoT network conditions:

1. Throughput (bps): Measures the data transmission efficiency.
2. Delay (ms): Reflects the time required for end-to-end communication.
3. Accuracy (%): Indicates the reliability of data decryption.
4. Packet Loss Ratio (%): Assesses communication reliability by measuring lost data.

The visualization function of MATLAB is excellent. In the performance evaluation of encryption algorithms, experimental results need to be visually displayed through charts, and MATLAB provides a powerful drawing function, which can quickly generate high-quality visual results, which is easy to analyse and compare the performance of different algorithms.

There are more compelling reasons to choose MATLAB than Python and NS3. Although Python has strong extensibility and rich library resources, for large matrix calculations and complex numerical simulations, Python usually needs to rely on NumPy, SciPy and other libraries, and the performance of these libraries may not be as good as MATLAB in some scenarios. MATLAB provides an integrated development environment that avoids the complexity of Python's need to install and configure multiple libraries, and performs better in mathematical modelling and algorithm

optimization. In contrast, NS3 is a tool focused on network protocol and communication behaviour simulation. Although it has advantages in network simulation, its support for encryption algorithms is weak, and it cannot fully meet the needs of this study in the performance evaluation of encryption algorithms.

When designing MATLAB simulation scheme, AES, RSA and ECC encryption algorithms are selected for performance simulation based on their unique advantages and applicable scenarios. AES, as an efficient symmetric key encryption algorithm, is selected for its fast data encryption and decryption capability and wide range of security applications. It is especially suitable for handling large amounts of data transmission, and thanks to the hardware acceleration support of modern processors, it can provide high strength security with low computing overhead. As a representative of asymmetric encryption algorithm, RSA is applicable to the establishment and authentication of secure connections, ensuring the security of the initial communication between the device and the server, and ensuring the authenticity and integrity of messages through digital signatures. ECC stands out for its resource optimization features, achieving the same level of security as RSA at a shorter key length, not only reducing the amount of computation, but also speeding up the speed, especially suitable for resource-constrained IoT devices, ECC is also seen as a potential solution to the challenges of future quantum computers.

AES, RSA and ECC together cover the requirements from efficient data encryption to secure session establishment and resource optimization, and cover different application scenarios and technical features, so they can provide a rich basis for research comparison.

In the simulation results, the number of IoT devices, encryption sizes, and hops are the X-axis parameters. The Y-axis showed the relevant performance measures, and these were changed to mimic actual IoT scenarios. The script produced 12 graphs where each graph presents the results of AES, RSA, and ECC for one metric against one parameter. Chapter IV provides a detailed discussion of the simulation findings, and these visualizations aid in deciphering the trade-off between security and performance.

### 3.1.2 Dynamics of the Simulation Process

The steps followed in a simulation are linear and well defined for purposes of replication and to independently and systematically assess algorithm performance. The workflow diagram illustrates the step-by-step progression of the simulation:

1. **Start Simulation:** Start the simulation environment in MATLAB, where more specific preparatory actions are to be undertaken next.
2. **Define Parameters:** The fundamental variables were defined as; the number of IoT devices, size of the encryption and number of hops.
3. **Initialize Network and Devices:** The smart home network can be only modelled and visualized in terms of IoT devices and their connection paths.
4. **Apply Encryption Algorithms:** To protect the communication links between various devices, RSA, AES, and ECC should be used.
5. **Simulate Data Transmission with Defined Hops:** To replicate real-world multi-device settings, securely send data via many hops.
6. **Decrypt Data and Evaluate Metrics:** Encrypt the transmitted data at the transmitter end and decrypt transmitted data at the receiver end and observe the FE and performance specifications like average end-to-end throughput, delay, accuracy and packet loss.
7. **Output Results:** It can result in graphical and numerical form and all the results as per comparison needed to be saved.

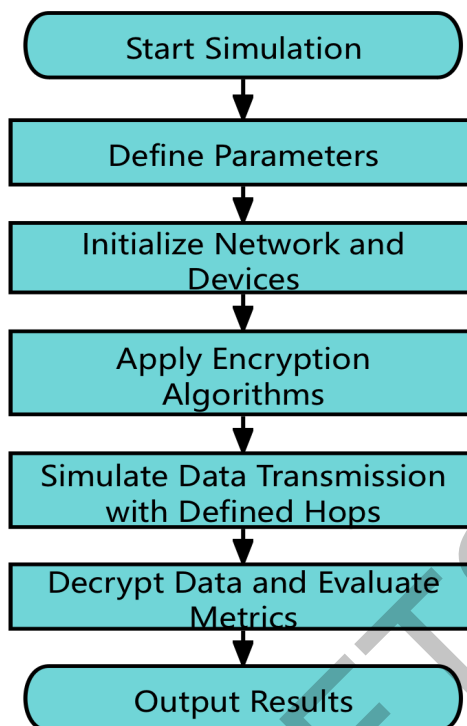


Figure 3.1 Workflow of the Simulation Environment

This Figure shows the various activities carried out in the course of the simulation process. Right from the commencement of the simulation, the data undergoes parameters definition, network and devices configuration, encryption algorithms implementation, transmission, decryption, and the assessment of the performance of the metrics. First, the simulation target is defined and four key performance parameters (throughput, delay, accuracy and packet loss rate) are selected to comprehensively measure the computational efficiency, real-time performance, data integrity and communication reliability of the encryption algorithm. In the simulation environment configuration phase, a real usage scenario is built by simulating the resource limitations of IoT devices (such as processing power, memory size, and network bandwidth) and the topology of the communication network (such as the number of devices, transmission distance, and relay hops), and the specific parameters of the encryption algorithm are set. Then, a set of simulated data packets are generated and encrypted and decrypted by MATLAB to ensure the integrity of data packets during transmission. In the aspect of performance evaluation, the throughput, delay, accuracy and packet loss rate are calculated by recording and analysing the simulation data, and the relationship between these indicators and the number of devices, key length and relay hops is generated by the visualization function of MATLAB. In order to verify

the reliability of the simulation results, the simulation was repeated with different parameters and compared with theoretical data or literature results to ensure the rationality and accuracy of the results. Finally, according to the simulation results, the advantages and disadvantages of the three algorithms are summarized, their applicability in smart home environment is analysed, and suggestions for improvement are put forward.

### 3.2 RSA ALGORITHM AND IMPLEMENTATION

The RSA algorithm is created based on the principle of public and private key and bases its cryptographic processing on the fiction of large numbers. Smart home networks use it in secure data transmission. RSA uses two large prime numbers,  $p$  and  $q$  to arrive at the generation of a public key and a private key. The values of  $e$  and  $n$  have a public domain while  $d$  and  $n$  have a private domain.

RSA encryption algorithm is an asymmetric encryption technique that starts the key generation process by selecting two large prime numbers  $p$  and  $q$ , first calculating their product  $n=p \times q$ , and using the Euler function  $\phi(n) = (p-1) \times (q-1)$  to determine the number of positive integers that are less than  $n$  and are prime to  $n$ . Next, an integer  $e$  smaller than  $\phi(n)$  and having a mutual prime of  $\phi(n)$  is selected as part of the public key, satisfying  $\gcd(e, \phi(n))=1$  and  $1 < e < \phi(n)$ , and the private key  $d$  is obtained by solving the congruence equation  $d \cdot e \equiv 1 \pmod{\phi(n)}$ , that is,  $d$  is the multiplicative inverse of  $\phi(n)$  in mode  $e$ . This method of key generation ensures the security of the key, as it is mathematically extremely difficult to find  $p$  and  $q$  backwards from the public  $n$ . Anyone with a public key can encrypt the message by converting it from message  $M$  to ciphertext  $C$  using the formula  $C = M^e \pmod{n}$ , where  $e$  and  $n$  are public. The original message  $M$  is recovered during decryption by using the private keys  $d$  and  $n$  in accordance with the formula  $M = C^d \pmod{n}$ . The security and secrecy of the communication are ensured since only those with the valid private key may finish this action.

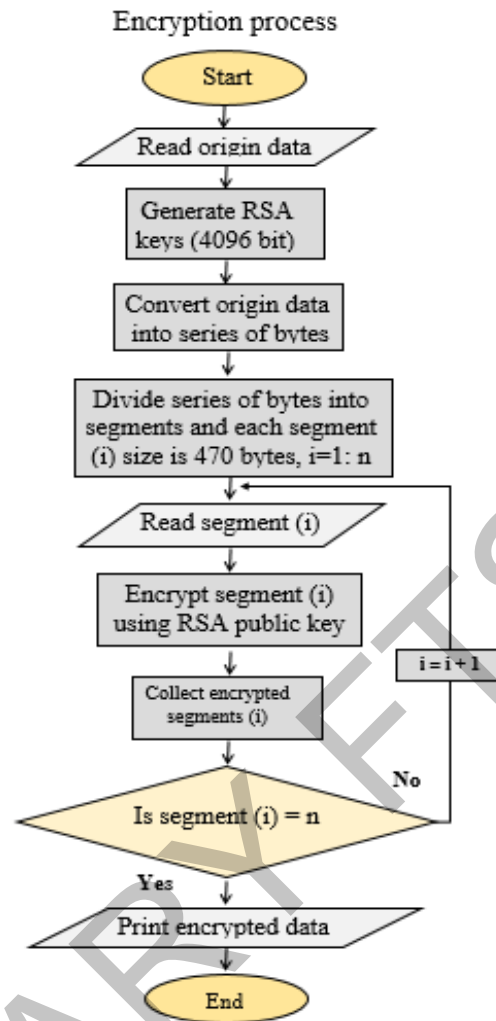


Figure 3.2 RSA Algorithm Flow Diagram

### 3.3 AES ALGORITHM AND IMPLEMENTATION

The applied encryption scheme known as AES maintains a set of cycles, each includes particular steps in order to enhance the system's integrity. AES is a type of symmetric key algorithm and the mode is highly praised for high computational efficiency and flexibility. In smart home networks, it can be employed for encrypting personal information that is in the networks.

In each round of processing of the AES algorithm, an S-Box SubBytes transformation is applied to each byte in the state matrix, where each byte is replaced with a new value according to a predefined replacement table, which enhances the nonlinear characteristics of the algorithm. By iterating the rows of the state matrix to

the left by varying offsets, the row shift transform improves the diffusion effect. The first row remains unchanged, the second row moves one position to the left, the third row moves two positions to the left, and the fourth row moves three positions to the left. The MixColumns transformation follows, where each state matrix column is multiplied by a fixed polynomial  $B(x)$  and represented as a polynomial in the finite field  $GF(2^8)$ , and the result is modulo  $x^4+1$ , which ensures that a change in a single byte affects all the bytes in the entire state matrix. Increased encryption security. Finally, the round key AddRoundKey transformation, when the state matrix and the round key corresponding to the round by the bit XOR operation, so that the key information of the round into the state, to ensure that each iteration has a different output until the predetermined number of rounds is completed.

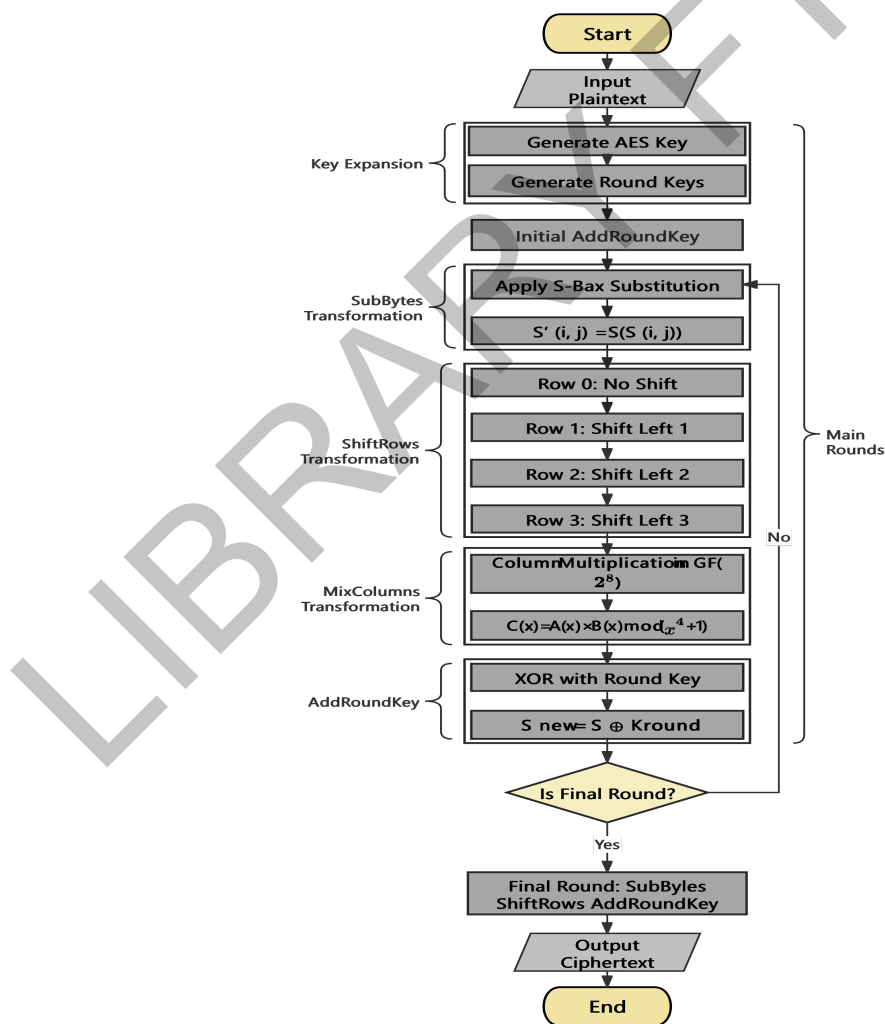


Figure 3.3 AES Algorithm Flow Diagram

AES algorithm encrypts the data in fixed-size blocks of data so the block size of data is 128 bits. First there is the Key Expansion, this is used to derive round keys from the core encryption key. The encryption process is subdivided to rounds which contain transformation that includes sub bytes of byte, shift rows, mix columns, and round key addition. It states that these procedures enhance the data security since they create confusion and dispersion. AES is widely employed in smart homes to encode the messages and protect the data because to the algorithm's high performance and minimal computational complexity. In smart homes, AES is employed where it's used to provide secure communication by encrypting the data between devices.

### 3.4 ECC ALGORITHM AND IMPLEMENTATION

ECC is a public key cryptography which Emphasizes on offering increased security using reduced key length that is very beneficial to IoT devices with limited resources.

ECC is a public-key cryptography technique based on elliptic curve equations over finite fields, whose formula is as follows:

$$y^2 = x^3 + ax + b \text{ mod } p \quad \dots(3.1)$$

Where a and b are the parameters that define a particular elliptic curve and p is a large prime number. The key exchange in ECC generates the public key Q by multiplying the private key d with a production point G, that is,  $Q = d \times G$ , which ensures the security of the private key because calculating the discrete logarithm problem is extremely difficult. In the encryption process of message P, a random number k is first selected and  $C1 = k \times G$  is calculated, and then  $C2 = P + k \times Q$  is calculated, where Q is the public key of the receiver; Therefore, the encrypted message consists of two parts, C1 and C2. When decrypting, the receiver uses its own private key d to recover the original message via the formula  $P = C2 - d \times C1$ , where the operation involved is point addition and its inversion on an elliptic curve.

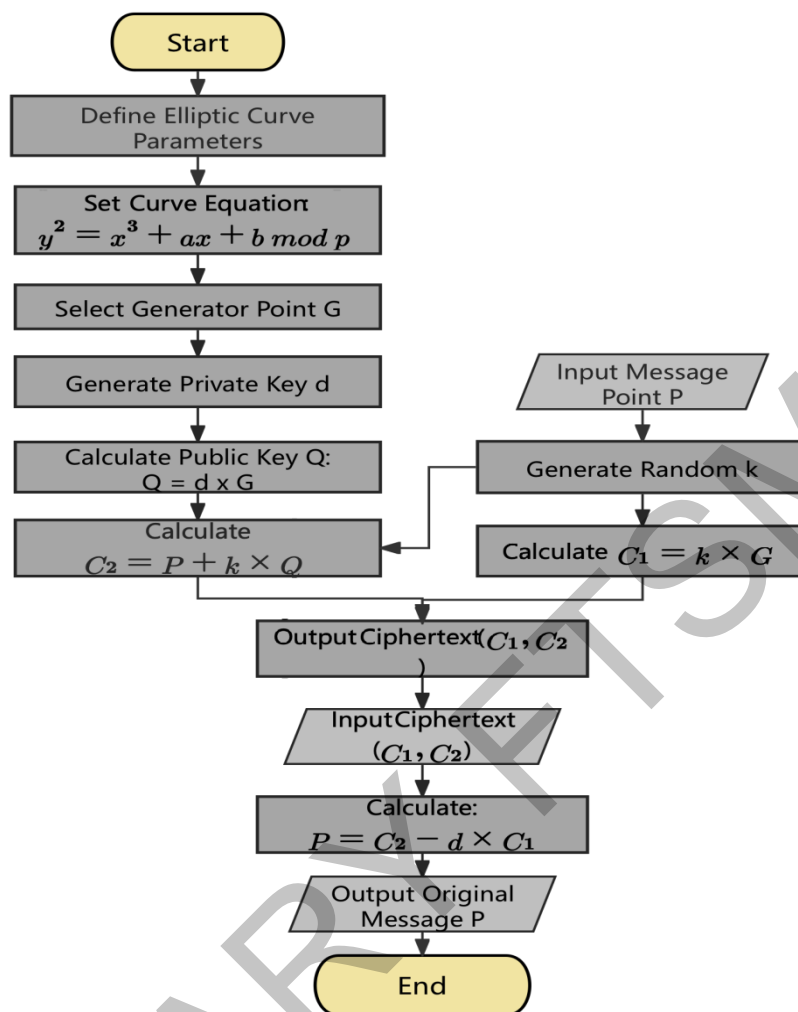


Figure 3.4 ECC Algorithm Flow Diagram

The ECC technique applies an asymmetric encryption system that stays comparatively protective while executing with minor key lengths, making the ECC technique suitable for devices with restricted resources such as the IoT devices. It starts with Key Generation, which means that with the help of elliptic curve Equations public and private keys are going to be created. In encryption the sender computes shared values in terms of elliptic curve point to secure the data. The Key Exchange stage enable devices to securely exchange the public keys, and when the Decryption has been made using the private key shall pull the original data. As it will be discussed in this paper, ECC is highly useful in smart home networks to serve the purpose of secure communication and device authentication. Smart home systems are another example of how ECC is utilized in order to render voice communication secure and to authenticate devices.

### 3.5 PERFORMANCE PARAMETERS

Four crucial metrics are used to assess the performance of the encryption algorithms (RSA, AES, and ECC): throughput, delay, accuracy, and packet loss ratio. Only throughput, delay, accuracy, and packet loss rate were selected as research parameters based on their ability to fully and efficiently evaluate the actual performance of encryption algorithms in a smart home environment, while being highly relevant to key requirements in IoT application scenarios (Chesuh et al. 2024). These four parameters together constitute a comprehensive framework for evaluating the performance of encryption algorithms in practical applications.

These four parameters are highly matched to the actual needs of the smart home environment (Rana et al. 2024). In the smart home scenario, encryption algorithms need to run on resource-constrained devices that have strict requirements on computing resources, power consumption, and communication stability. Efficient computing (corresponding to throughput) ensures that the system can respond quickly to large amounts of real-time data, real-time (corresponding to delay) ensures that security alarms or home automation can be triggered in a timely manner, data integrity (corresponding to accuracy) prevents sensitive information (such as authentication and control commands) from being compromised or lost. Communication reliability (corresponding to packet loss rate) reduces data transmission problems caused by limited network bandwidth or signal interference. The selection of these four parameters can avoid parameter redundancy and focus on the core performance indicators of the encryption algorithm. Although other parameters, such as power consumption or algorithmic complexity, also have some research significance, too many metrics can lead to increased analysis complexity. These four parameters are sufficient to fully reflect the comprehensive performance of the encryption algorithm in the communication and computation dimensions, while avoiding the focus of the research.

#### 3.5.1 Throughput

Throughput means the achievement of the amount of information that is handled or passed through the system. The time for communication propagation must be considered together with the additional security burden that comes with encryption.

Through put can be expressed as bits per second (Bps) or packets per second (pps). It indicates the flow of transmission of information using an encryption algorithm without forming traffic jams in transmission.

1. Measurement in Experiment
  - a. Throughput in the simulated network during the simulation is given by the overall amount of data transmitted to the recipient end within a given time frame while being limited by the overhead of encryption
2. Parameters Required
  - a. Total Data Transmitted (Dt) in bits.
  - b. Total Transmission Time (Tt) in seconds.
  - c. Unit: Bits per second (bps) or packets per second (pps).
3. Mathematical Equation

$$\text{Throughput}(\text{bps}) = \frac{\text{Total Data Transmitted (bits)}}{\text{Total Transmission Time (seconds)}} \quad \dots(3.2)$$

### 3.5.2 Delay

Delay is the time required to encode, transmit and decode data between the Equipment. Reducing the amount or length of such delays is paramount to sustaining real-time performance in smart homes. Postponement is calculated in millisecond (ms) and can encompass encryption, decryption and transfer of data.

1. Parameters Required
  - a. Encryption Time (Te) in milliseconds.
  - b. Transmission Time (Tt) in milliseconds.
  - c. Decryption Time (Td) in milliseconds.
2. Mathematical Equation

$$\text{Delay}(\text{seconds}) = Te + Tt + TD \quad \dots(3.3)$$

### 3.5.3 Accuracy

The degree to which data is delivered and decrypted error-free is known as availability accuracy. High accuracy therefore indicates dependable and safe communication between the linked devices. The ratio of successfully transferred to correctly decoded data packets is used to calculate accuracy. High accuracy should be maintained for it is very important to minimize loss of data, Protection of message content, maintaining communication media appropriateness, and guaranteeing that encryption algorithms do not compromise data quality.

1. Parameters Required
  - a. Correctly Received Packets ( $P_c$ ).
  - b. Total Transmitted Packets ( $P_t$ ).

2. Mathematical Equation

$$\text{Accuracy}(\%) = \left( \frac{\text{Correctly Received Packets}}{\text{Total Transmitted Packets}} \times 100 \right) \quad \dots(3.4)$$

### 3.5.4 Packet Loss Ratio

The number of data packets lost during transmission is known as the packet loss ratio. Packet loss in a smart companion causes security problems in a home system and lowers communication quality. High packet loss may cause repeated transmission, delays, and its' devastating impacts include separating parts of the IP address at different nodes and jumping over several other parts before rejoining the address susceptibility to attacks.

1. Impact on Security
  - a. Lost Packets ( $P_l$ ).
  - b. Total Transmitted Packets ( $P_t$ ).

2. Mathematical Equation

$$\text{Packet Loss}(\%) = \left( \frac{\text{Lost Packets}}{\text{Total Transmitted Packets}} \times 100 \right) \quad \dots(3.5)$$

## CHAPTER IV

### RESULTS AND DISCUSSION

#### 4.1 RESULTS

The following is a discussion of the smart house IoT environment simulation findings in this chapter. MATLAB was used to run the simulation. The performance of the AES, RSA, and ECC algorithm scenarios was compared. In each case, draw a line graph where applicable. For example, in the simulation scenario studying throughput versus the number of IoT devices, the X-axis is the number of IoT devices, the X-axis is the number of 10 to 100, and the Y-axis is the performance statistics. Through such simulations, we were able to analyse the impact of the number of devices under different encryptions on IoT network security and overall throughput.

##### 4.1.1 Throughput performance results

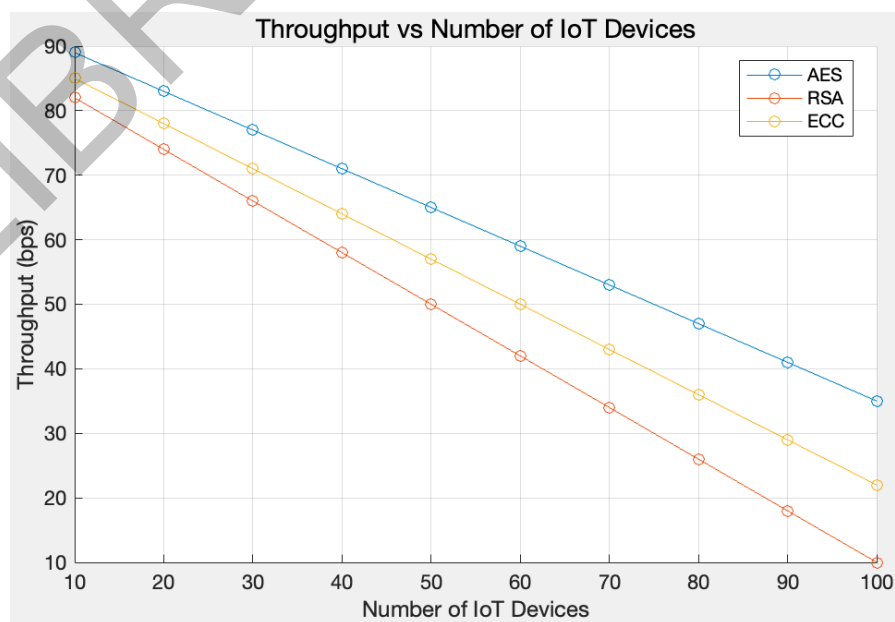


Figure 4.1 Throughput vs Number of IoT Devices

AES, RSA, and ECC throughput varies with the number of IoT devices. The number of IoT devices is shown on the horizontal axis (from 10 to 100), and the vertical axis represents throughput (in packets per second, bps). Initially, AES had the highest throughput at close to 90 bps, ECC was second at close to 85 bps and RSA had the lowest at close to 80 bps. As the number of devices increases, the throughput of all three algorithms decreases linearly, but at different rates. AES experienced the slowest decline with a final throughput of about 35 bps, RSA the fastest with a final throughput of about 10 bps, and ECC fell somewhere in between with a final throughput of about 20 bps. This shows that AES shows better performance stability when dealing with a large number of IoT devices, which is suitable for application scenarios requiring high throughput, RSA is more suitable for scenarios with high security requirements but a small number of devices, ECC provides a compromise solution to balance security and performance. As the number of IoT devices increases, throughput declines mainly because more devices join the network and compete for limited bandwidth and processing power, leading to bottlenecks. The increase in network traffic causes transmission delay, increases packet loss rate, and requires retransmission, which further reduces the effective throughput. Encryption algorithms such as RSA and ECC will significantly slow down the data processing speed as the number of devices increases due to high computing costs.

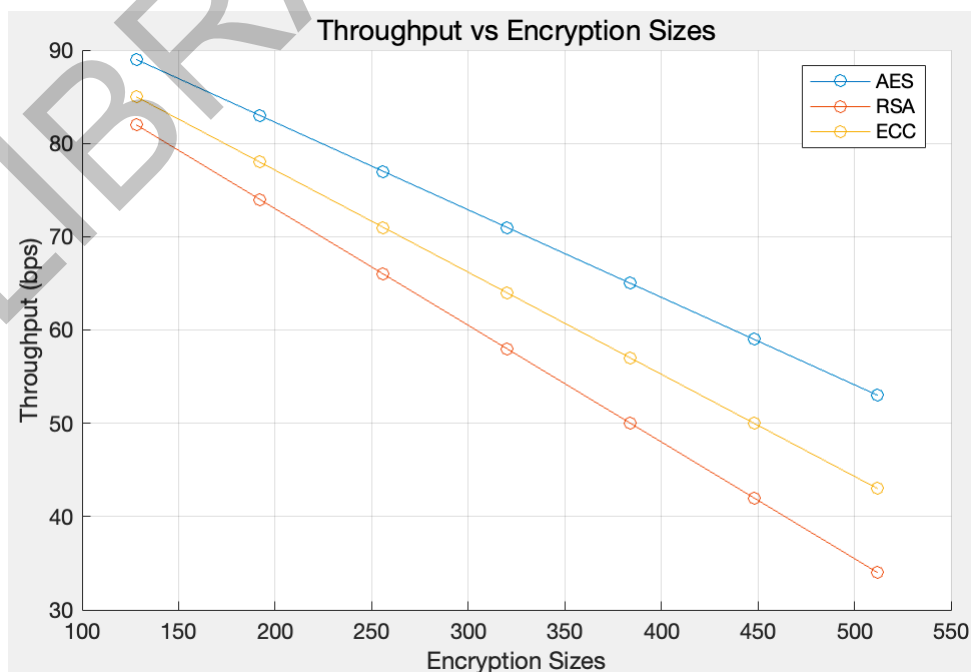


Figure 4.2 Throughput vs Encryption Sizes

The throughput change of AES, RSA, and ECC when the encryption key size increases. As the encryption size increases, throughput decreases mainly because larger encryption keys require more computing resources and time to process. Encryption algorithms such as AES, RSA, and ECC increase the time required to encrypt and decrypt data in the face of larger key sizes, which results in slower data transmission and, in turn, reduced throughput. The horizontal axis represents the size of the encryption key (from 100 to 500 bits), and the vertical axis represents throughput (in packets per second, bps). Initially, AES had the highest throughput at close to 90 bps, ECC was second at close to 85 bps, and RSA had the lowest at close to 80 bps. As the key size increases, the throughput of all three algorithms decreases linearly. Specifically, when the key size is 128 bits, AES has a throughput of about 85 bps, RSA about 75 bps, and ECC about 80 bps. When the key size is 192 bits, the throughput of AES drops to about 75 bps, RSA to about 65 bps, and ECC to about 70 bps. When the key size is 256 bits, AES throughput is about 65 bps, RSA drops to about 55 bps, and ECC drops to about 60 bps. When the key size is 512 bits, the throughput of AES drops to about 50 bps, RSA to about 35 bps, and ECC to about 45 bps. This shows that AES has better performance stability when dealing with large keys, and is suitable for application scenarios requiring high throughput, while RSA is more suitable for scenarios requiring high security but small key size.

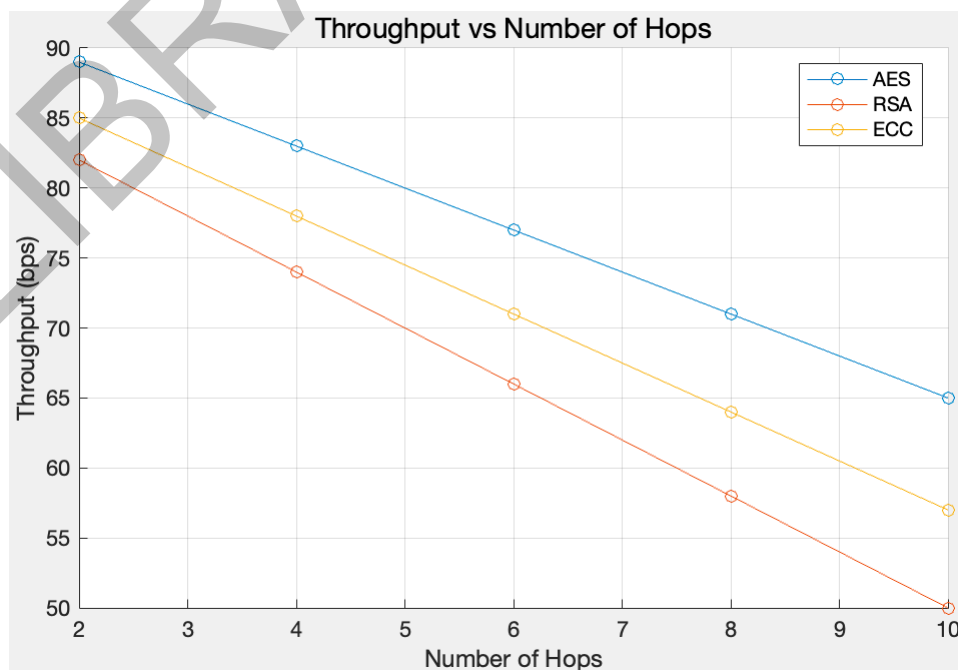


Figure 4.3 Throughput vs Number of Hops

The throughput change of AES, RSA, and ECC with the increase of hops during data transmission. The horizontal axis represents the number of hops (from 2 to 10), and the vertical axis represents throughput (in packets per second, bps). Initially, AES had the highest throughput at nearly 90 bps, ECC was second at nearly 85 bps, and RSA had the lowest throughput at nearly 83 bps. As the number of hops increases, the throughput of all three algorithms decreases linearly, but at different rates. Specifically, with a hop count of 2, AES has a throughput of about 90 bps, RSA about 83 bps, and ECC about 85 bps. With a hop count of 4, the throughput of AES drops to about 80 bps, RSA to about 76 bps, and ECC to about 78 bps. With a hop count of 6, AES throughput is about 70 bps, RSA drops to about 69 bps, and ECC drops to about 71 bps. With a hop count of 8, the throughput of AES drops to about 60 bps, RSA to about 62 bps, and ECC to about 64 bps. When the hop count is 10, the throughput of AES drops to about 50 bps, RSA to about 50 bps, and ECC to about 55 bps. As the number of hops increases, throughput decreases because each hop through introduces additional delay and processing time, which reduces the effective bandwidth. In a multi-hop network, packets need to be forwarded through multiple nodes, and each node will increase the transmission delay and data processing burden to a certain extent, which will cumulatively affect the overall data transmission rate. AES is better in this regard, it almost solves the problem caused by the increase in the number of hops, unlike RSA and ECC as the increase in the number of hops significantly increase the computing burden, so AES is more suitable for multi-hop networks in the IoT, such networks often need to transfer data from the source device through multiple hops to the target device.

#### 4.1.2 Delay performance results

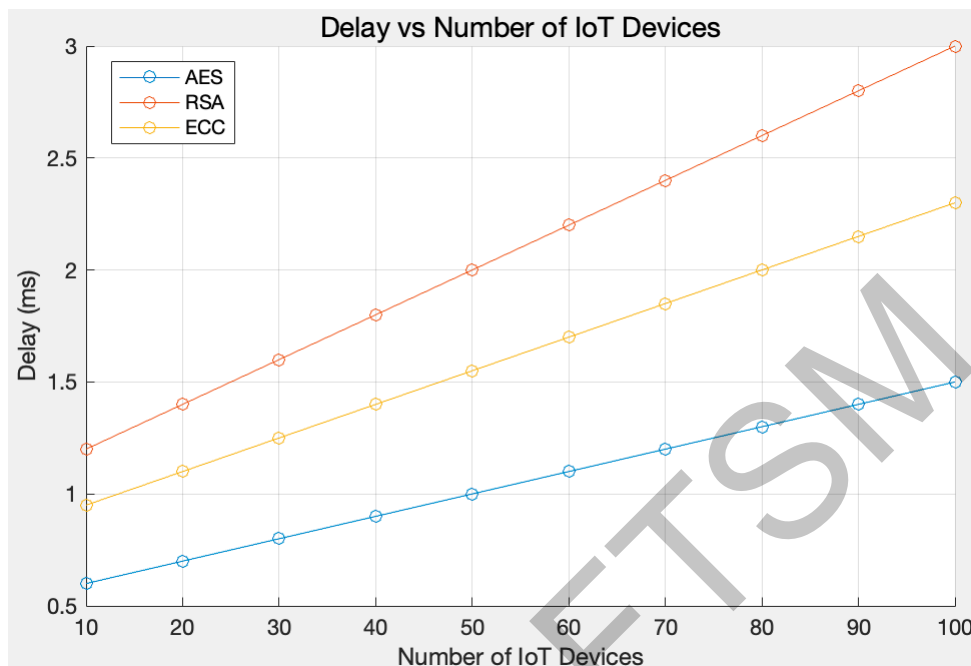


Figure 4.4 Delay vs Number of IoT Devices

Delay changes of AES, RSA, and ECC as the number of IoT devices increases. As the number of IoT devices increases, so does delay, as more devices mean greater competition for network traffic and resources. Each new device consumes bandwidth and introduces additional data processing requirements, causing network congestion. This not only increases the waiting time of packets during transmission, but also causes packet loss and retransmission, further increasing the overall delay. Initially, AES has the lowest delay, close to 0.5 ms, ECC is next, close to 0.9 ms, and RSA has the highest delay, close to 1.2 ms. As the number of devices increases, the delay of all three algorithms increases linearly, but at different rates. AES has the slowest rise with a final delay of about 1.5 ms, RSA has the fastest rise with a final delay of about 3.0 ms, and ECC is in between with a final delay of about 2.3 ms. This shows that AES shows low delay when dealing with a large number of IoT devices, which is suitable for application scenarios requiring low delay. RSA's delay increases the fastest with the increase of the number of devices, which may be more suitable for scenarios with low delay requirements but high security requirements. ECC provides a compromise solution to balance delay and security. Applicable to scenarios that require both delay and efficiency. This graph demonstrates AES's capacity to reduce delay and demonstrates

that it is appropriate for use in IoT systems that are delay-sensitive. While the RSA's delay values are quite large and render the method completely inappropriate for situations where speedy reaction times are crucial, ECC may be anticipated to deliver adequate delay values for moderate loads, albeit always being on the upper side.

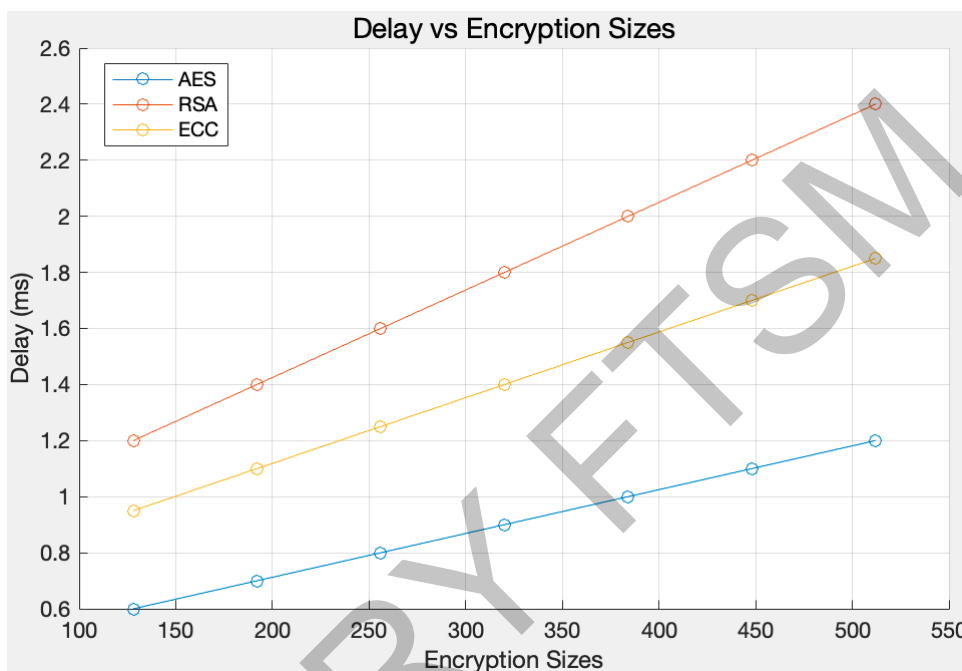


Figure 4.5 Delay vs Encryption Sizes

The delay change of AES, RSA, and ECC when the encryption key size increases. As the encryption size increases, so does the delay, because larger keys require more computing resources and time to complete the encryption and decryption process. As the key size increases, encryption algorithms such as AES, RSA, and ECC take longer to process data, resulting in higher transmission delays for each packet. The horizontal axis represents the size of the encryption key (from 100 to 500 bits) and the vertical axis represents the delay (in milliseconds, ms). Initially, AES had the lowest delay, close to 0.6ms, ECC followed by close to 0.9ms, and RSA had the highest delay, close to 1.2ms. Specifically, when the key size is 128 bits, the delay of AES is about 0.7 ms, RSA is about 1.4 ms, and ECC is about 1.1 ms. When the key size is 192 bits, the delay of AES increases to about 0.8 ms, RSA to about 1.6 ms, and ECC to about 1.3 ms. When the key size is 256 bits, the delay of AES is about 0.9 ms, RSA increases to about 1.8 ms, and ECC increases to about 1.5 ms. When the key size is 512 bits, the delay of AES increases to about 1.2 ms, RSA to about 2.4 ms, and ECC to about 1.8

ms. As the key size increases, the delay of all three algorithms increases linearly, but at different rates. AES has the slowest rise with a final delay of about 1.2 ms, RSA has the fastest rise with a final delay of about 2.4 ms, and ECC is in between with a final delay of about 1.8 ms. This indicates that AES has a low delay when processing large keys, and is suitable for scenarios that require low delay. RSA's delay increases the fastest as the key size increases, and it is more suitable for scenarios that require low delay but high security.

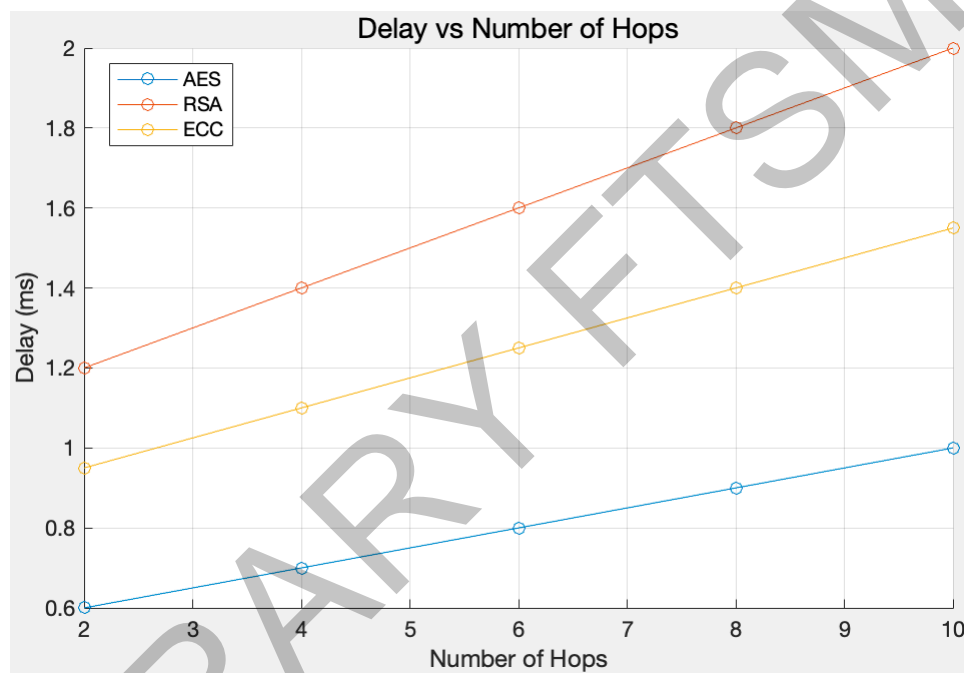


Figure 4.6 Delay vs Number of Hops

The delay change of AES, RSA, and ECC with the increase of hops during data transmission. The horizontal axis represents the number of hops (from 2 to 10), and the vertical axis represents the delay (in milliseconds, ms). As the number of hops increases, so does the delay, as each additional hop past introduces new transmission and processing time. In a multi-hop network, packets need to be forwarded through multiple nodes, and each node not only increases the time of physical transmission, but may also increase queuing, processing, and potential retransmission times, which all add up and cause overall delay to rise. Initially, AES has the lowest delay, close to 0.6 ms, ECC is next, close to 1.0 ms, and RSA has the highest delay, close to 1.2 ms. With the increase of the number of hops, the delay of all three algorithms shows a linear upward trend, but the rising speed is different. Specifically, when the hop count is 2, the delay of AES

is about 0.6 ms, RSA is about 1.2 ms, and ECC is about 1.0 ms. When the hop count is 4, the delay of AES increases to about 0.8 ms, RSA to about 1.4 ms, and ECC to about 1.2 ms. When the hop count is 6, the delay of AES is about 1.0 ms, RSA increases to about 1.6 ms, and ECC increases to about 1.4 ms. When the hop count is 8, the delay of AES increases to about 1.2 ms, RSA to about 1.8 ms, and ECC to about 1.6 ms. When the hop count is 10, the delay of AES increases to about 1.4 ms, RSA to about 2.0 ms, and ECC to about 1.8 ms.

#### 4.1.3 Accuracy performance results

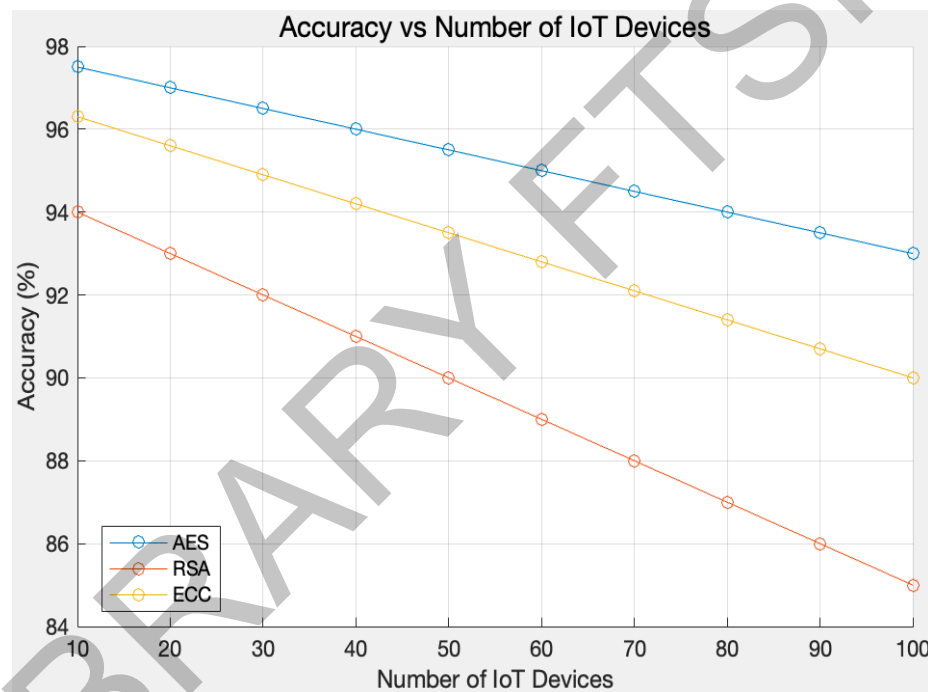


Figure 4.7 Accuracy vs Number of IoT Devices

Accuracy changes of AES, RSA, and ECC as the number of IoT devices increases. The horizontal axis represents the number of IoT devices (from 10 to 100), and the vertical axis represents accuracy (in percentage, %). As the number of IoT devices increases, accuracy decreases because more devices bring with them higher network traffic and encryption and decryption requirements, increasing the likelihood of error or loss of packets. When the number of devices increases, the network congestion intensifies, leading to the increase of data transmission delay and packet loss rate, which directly affects the integrity and accuracy of data transmission. Initially, AES had the highest accuracy at nearly 98%, ECC was next at nearly 96%, and RSA

had the lowest accuracy at nearly 94%. The accuracy of all three algorithms decreased linearly as the number of devices increased, but at different rates. AES declined the slowest, with a final accuracy of about 93%, RSA the fastest, with a final accuracy of about 85%, and ECC fell somewhere in between, with a final accuracy of about 90%.

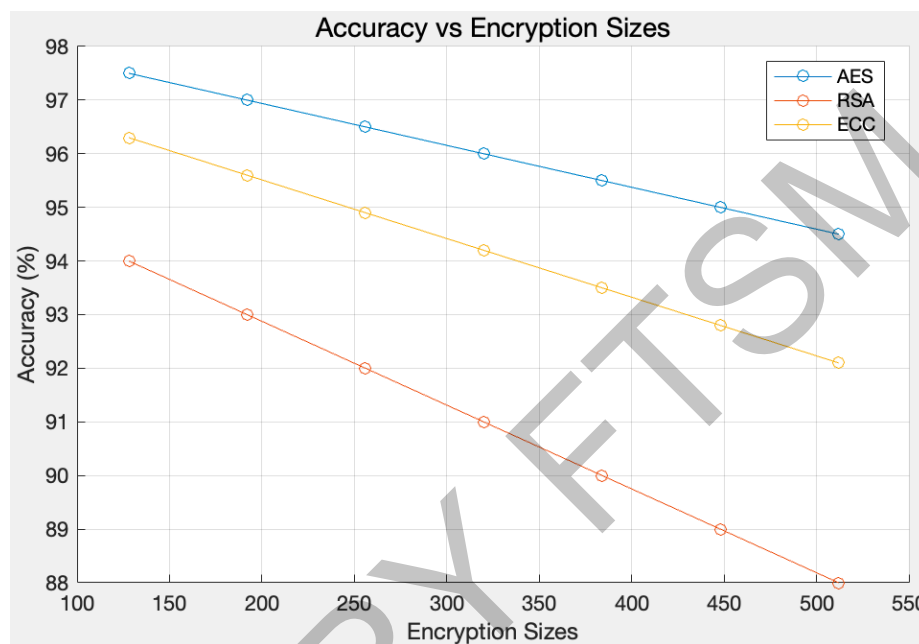


Figure 4.8 Accuracy vs Encryption Sizes

The accuracy change of AES, RSA, and ECC when the encryption key size increases. The horizontal axis indicates the size of the encryption key (from 100 to 500 bits), and the vertical axis indicates accuracy (in percentage, %). As the encryption size increases, accuracy decreases because larger keys require more computing resources and time to process, which increases the complexity of packet processing and the potential for errors. Initially, AES had the highest accuracy at nearly 97%, ECC was next at nearly 96%, and RSA had the lowest accuracy at nearly 94%. As the key size increases, the accuracy of all three algorithms decreases linearly. Specifically, when the key size is 128 bits, AES accuracy is about 96.5%, RSA is about 93.5%, and ECC is about 95.5%. When the key size is 192 bits, the accuracy of AES drops to about 95.5%, RSA to about 92.5%, and ECC to about 94.5%. When the key size is 256 bits, AES accuracy is about 94.5%, RSA drops to about 91.5%, and ECC drops to about 93.5%. When the key size was 512 bits, the accuracy of AES dropped to about 93%, RSA to about 88%, and ECC to about 92%. This indicates that AES shows high accuracy when

dealing with large keys, and is suitable for application scenarios requiring high accuracy. RSA's accuracy decreases the fastest as the key size increases, so it may be more suitable for scenarios requiring low accuracy but high security.

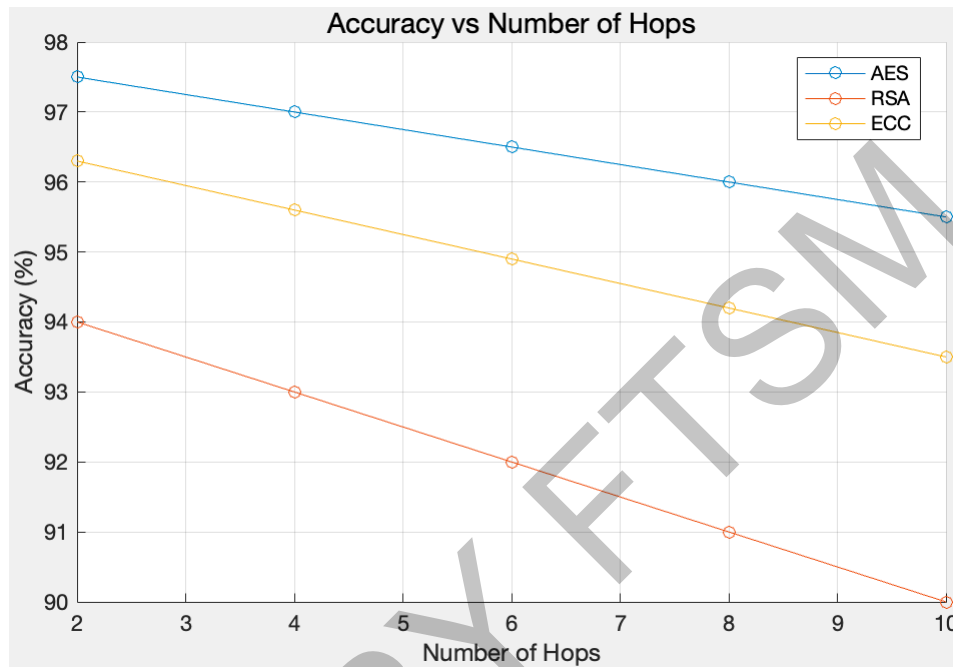


Figure 4.9 Accuracy vs Number of Hops

The accuracy change of AES, RSA, and ECC as the number of hops increases during data transmission. The horizontal axis represents the number of hops (from 2 to 10), and the vertical axis represents accuracy (in percentage, %). As the number of hops increases, accuracy decreases because the delay and processing burden increases with each node a packet passes through as it travels through a multi-hop network, which increases the likelihood of packet loss or error. Especially in multi-hop environments, cumulative transmission errors and potential network congestion can further affect data integrity. Initially, AES had the highest accuracy at nearly 98%, ECC was next at nearly 96.5%, and RSA had the lowest accuracy at nearly 94%. As the number of hops increases, the accuracy of all three algorithms decreases linearly, but at different rates. Specifically, when the hop count is 2, AES is about 98% accurate, RSA is about 94%, and ECC is about 96.5%. When the hop count is 4, the accuracy of AES drops to about 97%, RSA to about 93%, and ECC to about 95.5%. When the hop count is 6, the accuracy of AES is about 96%, RSA drops to about 92%, and ECC drops to about 94.5%. When the hop count is 8, the accuracy of AES drops to about 95%, RSA to

about 91%, and ECC to about 93.5%. When the hop count is 10, the accuracy of AES drops to about 94%, RSA to about 90%, and ECC to about 92.5%.

#### 4.1.4 Packet Loss Ratio performance results

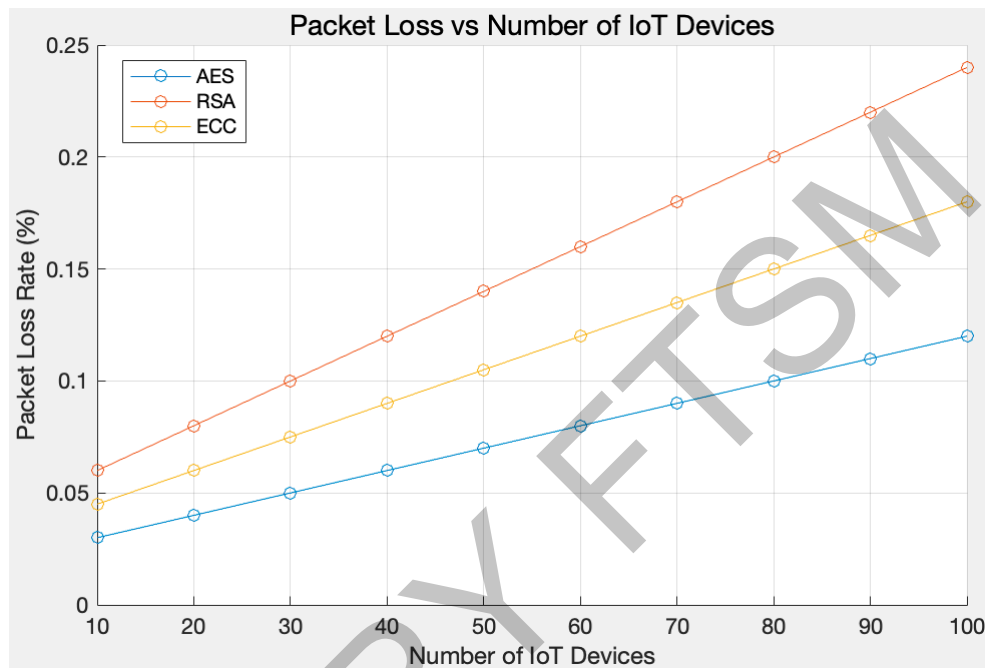


Figure 4.10 Packet Loss Rate vs Number of IoT Devices

Change of packet loss rate of AES, RSA, and ECC when the number of IoT devices increases. As the number of IoT devices increases, the packet loss rate increases because more devices are added to the network, increasing the possibility of network traffic and congestion, resulting in higher transmission delays and higher risk of packet loss. Each new device brings additional data processing requirements, which may cause competition and bottlenecks in network resources. The horizontal axis represents the number of IoT devices (from 10 to 100), and the vertical axis represents the packet loss rate (in percentage, %). Initially, the packet loss rate of AES is the lowest, close to 0.03%, ECC is the second, close to 0.05%, and RSA is the highest, close to 0.06%. As the number of IoT devices increases, the packet loss rate of all three algorithms increases linearly, but at different rates. Specifically, when the number of devices is 20, the packet loss rate for AES is about 0.04%, for RSA about 0.07%, and for ECC about 0.06%. When the number of devices is 40, the packet loss rate increases to about 0.05% for AES, about 0.10% for RSA, and about 0.08% for ECC. When the number of devices

is 60, the packet loss rate for AES is about 0.07%, RSA increases to about 0.13%, and ECC increases to about 0.11%. When the number of devices is 80, the packet loss rate increases to about 0.09% for AES, about 0.16% for RSA, and about 0.14% for ECC. When the number of devices is 100, the packet loss rate increases to about 0.11% for AES, about 0.24% for RSA, and about 0.18% for ECC. The findings highlight the significance of lowering packet loss by highlighting its effects on IoT networks. By taking longer to build the algorithm with the least amount of loss, AES sets itself apart from the other algorithms and is thus appropriate for reliable communication. As a result, ECC is a sensible choice, although RSA performs less than well.

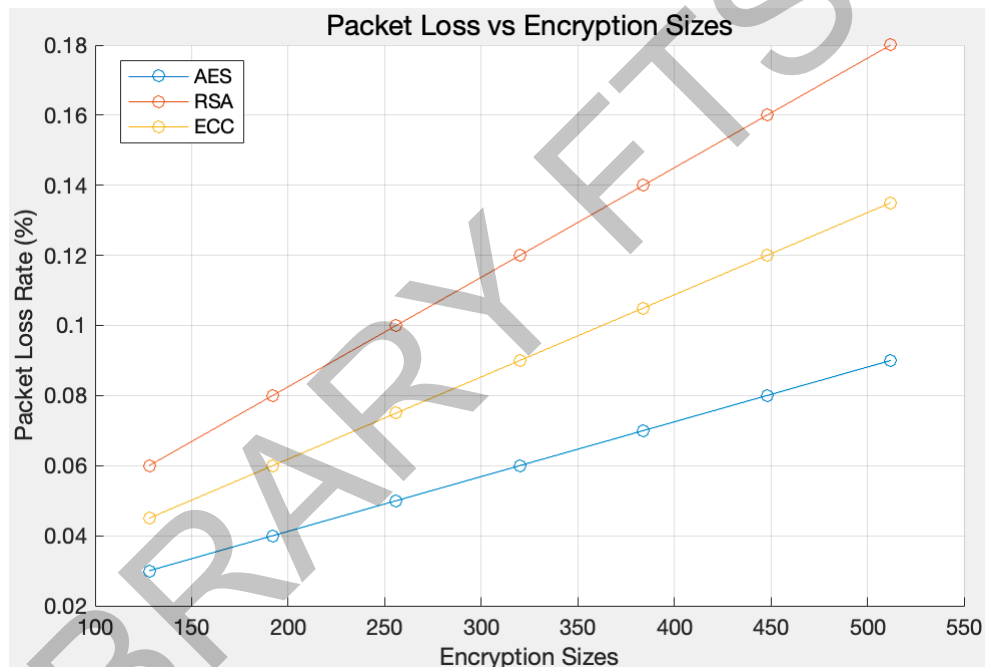


Figure 4.11 Packet Loss Rate vs Encryption Sizes

Change the packet loss rate of AES, RSA, and ECC when the encryption key size increases. The horizontal axis represents the size of the encryption key (from 100 to 500 bits), and the vertical axis represents the packet loss rate (in percentage, %). The packet loss rate increases as the encryption size increases because larger key sizes require more processing time and computing resources, resulting in increased delay for each node when encrypting and decrypting data. This additional processing burden increases queue times for packets when the network is congested, increasing the risk of timeouts and packet loss. Initially, the packet loss rate of AES is the lowest, close to 0.03%, followed by ECC, close to 0.045%, and RSA is the highest, close to 0.06%.

With the increase of the key size, the packet loss rate of all three algorithms increases linearly, but the rate of increase is different. Specifically, when the key size is 128 bits, the packet loss rate for AES is about 0.04%, for RSA about 0.075%, and for ECC about 0.055%. When the key size is 192 bits, the packet loss rate increases to about 0.055% for AES, 0.09% for RSA, and 0.07% for ECC. When the key size is 256 bits, AES has a packet loss rate of about 0.07%, RSA increases to about 0.105%, and ECC increases to about 0.085%. When the key size is 512 bits, the packet loss rate increases to about 0.09% for AES, about 0.17% for RSA, and about 0.135% for ECC. This shows that increasing the encryption size means that more processing time is required, which can easily lead to increased packet loss in a network environment. RSA algorithm, due to its complex computation process, has a significant decrease in processing efficiency in the face of large encryption size, which makes the packet loss more serious. This indicates that in environments with low requirements for packet loss, such as some non-critical data transmission scenarios in smart homes, RSA or ECC algorithms can be properly considered, but the risk of packet loss should be paid attention to. The AES algorithm, due to its efficient data processing capability, can better control the packet loss rate when processing different encryption sizes, and is suitable for scenarios with high data integrity requirements in smart homes, such as the transmission of important configuration information and the interaction of sensitive data, so as to ensure that the data reaches the destination accurately.

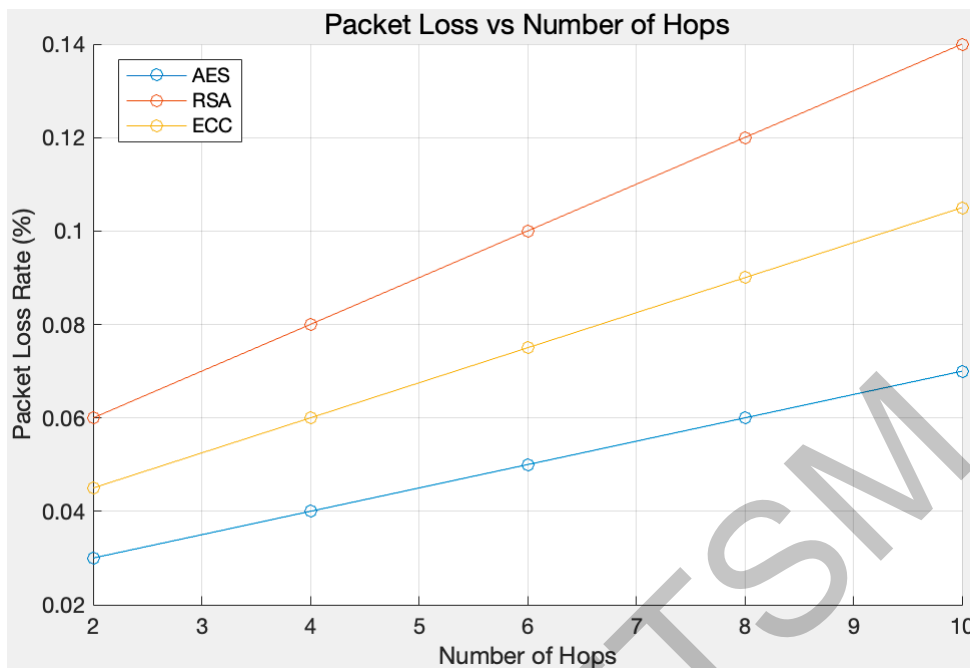


Figure 4.12 Packet Loss Rate vs Number of Hops

The packet loss rate of AES, RSA, and ECC changes with the increase of hops during data transmission. The horizontal axis indicates the number of hops (from 2 to 10), and the vertical axis indicates the packet loss rate (in percentage, %). The packet loss rate increases as the number of hops increases, because each additional hop introduces new transmission delays and processing times, increasing the risk of packet loss. In a multi-hop network, packets need to be forwarded through multiple nodes, each of which can result in packet loss due to processing burden, network congestion, or transmission errors. Initially, the packet loss rate of AES is the lowest, close to 0.03%, followed by ECC, close to 0.045%, and RSA is the highest, close to 0.06%. The packet loss rate of all three algorithms increases linearly with the increase of hop count, but the increase speed is different. Specifically, when the hop count is 2, the packet loss rate of AES is about 0.03%, RSA is about 0.06%, and ECC is about 0.045%. When the hop count is 4, the packet loss rate of AES increases to about 0.04%, RSA to about 0.08%, and ECC to about 0.06%. When the hop count is 6, the packet loss rate of AES is about 0.05%, RSA increases to about 0.10%, and ECC increases to about 0.075%. When the hop count is 8, the packet loss rate increases to about 0.06% for AES, about 0.12% for RSA, and about 0.09% for ECC. When the hop count is 10, the packet loss rate of AES increases to about 0.07%, RSA to about 0.14%, and ECC to about 0.105%.

## 4.2 ANALYSIS

### 4.2.1 Throughput Analysis

#### a. Throughput vs Number of IoT Devices

1. Observation: As the number of IoT devices increases, the throughput decreases for all methods. As the number of devices rises, AES maintains its superior efficiency over both ECC and RSA and is distinguished by greater throughput levels.
2. Reasoning: Because AES has a lower computational cost than RSA and ECC, it can process more data packets. As the number of devices increases, competition for resources intensifies, and network congestion outperforms the availability of many methods.
3. Implication: AES works best in situations with high demand, particularly in large IoT networks like industrial systems or applications for smart cities.

#### b. Throughput vs Encryption Sizes

1. Observation: It is also clear that a lesser throughput is attained as the encryption size grows. Among the four methods, AES remains the most effective, but RSA is the least effective in terms of processing complexity.
2. Reasoning: Data transmission slows down when encryption size increases because it takes longer to encrypt data. Throughput is significantly reduced by RSA implementations, whereas ECC offers an improvement but again does not give way to AES.
3. Implication: The throughput requirements of the particular application should be considered when selecting the kind of encryption; for instance, lower encryption sizes should be utilized in high throughput applications, particularly if RSA or ECC encryption is being employed.

**c. Throughput vs Number of Hops:**

1. Observation: This is due to the fact that each hop adds more throughput while moving it farther away from the initial optimal value. Naturally, AES exhibits the highest throughput, while ECC outperforms RSA in terms of throughput; nevertheless, RSA's throughput declines with a steeper curve.
2. Reasoning: Effective bandwidth is decreased as a result of the additional delay and processing time added by each hop. AES virtually solves the hop growth issue, unlike RSA and ECC, and adds computational weight with each hop, unlike RCAEZ.
3. Implication: AES is especially well-suited for IoT networks with several hops, such the direct sequence mesh network seen in smart home technologies, which transmits data from a source that passes through numerous hops before reaching the intended device.

**4.2.2 Delay Analysis**

**a. Delay vs Number of IoT Devices:**

1. Observation: The increase in the number of devices causes delays in all algorithms. RSA has the highest delay with respect to time, then comes ECC while AES has the least delay.
2. Reasoning: Because of the complexity of the method employed, the number of devices leads to more traffic and more encryption or decryption, particularly for RSA. Wait times will be shortened because to AES's lightweight design.
3. Implication: AES works well in situations that need for regular sampling, such a real-time monitoring system in the medical field or a crucial industrial operation.

**b. Delay vs Encryption Sizes:**

1. Observation: The outcome demonstrates how the size of the encryption affects the likelihood that the delay will increase as the encryption size

grows. ECC and AES only have a small increase in delay, whereas RSA has the most delay.

2. Reasoning: The wider encryption sizes also increase processing time, which makes the delay worse. The RSA technique primarily uses modular exponentiation, which makes the encryption size crucial in this situation.
3. Implication: Although AES has been shown to be suitable for applications with low delay, reasonable encryption sizes should be taken into account in order to get the best possible balance between security and speed.

**c. Delay vs Number of Hops:**

1. Observation: As the number of hops grows, so does the time. Out of all the algorithms, RSA has the most delay, followed by ECC in second place and AES in last.
2. Reasoning: Due to encryption and decryption at the subsequent nodes, each hop results in a processing delay. This overhead is not really noticeable in AES's architecture.
3. Implication: AES facilitates speedier communications in multi-hop contexts, such as IoT networks spread across wide regions, and is utilized in applications like environmental monitoring.

**4.2.3 Accuracy Analysis**

**a. Accuracy vs Number of IoT Devices:**

1. Observation: As the number of devices rises, the accuracy improvement for all algorithms decreases. The highest implementation accuracy is required by AES, which is followed by RSA and ECC.
2. Reasoning: Higher device numbers result in a somewhat higher degree of accuracy loss since they increase the likelihood that the packet may be incorrect or lost. Because AES's robustness facilitates this process, it is therefore more accurate than other models.

3. Implication: AES is suitable for all situations when data must be kept essentially unaltered, including when exchanging money or utilizing a digital communication line for health-related purposes.

**b. Accuracy vs Encryption Sizes:**

1. Observation: Accuracy is somewhat impacted by encryption size. AES continues to be the most accurate, while RSA's decrease is more pronounced.
2. Reasoning: Higher encryption size values result in higher processing demands, which, especially for RSA, might cause many mistakes or even packet drops.
3. Implication: In situations where accuracy is crucial, AES with moderate encryption sizes is often used.

**c. Accuracy vs Number of Hops:**

1. Observation: As the number of hops grows, the findings' accuracy decreases. RSA has the greatest degree of decreases, whilst AES is least impacted.
2. Reasoning: The likelihood that a packet may be lost or that several mistakes will occur is increased in multi-hop networks. Because AES operates so efficiently, it is less impacted in this area than RSA and ECC.
3. Implication: AES is the best algorithm for multi-hop networks in order to maintain high and consistent accuracy, such as in dispersed multiple sensor applications for the IoT in agriculture.

#### **4.2.4 Packet Loss Analysis**

**a. Packet Loss vs Number of IoT Devices:**

1. Observation: Simulations reveal that the degree of packet loss increases in direct proportion to the number of devices in use. When it comes to packet

loss, AES is the best, followed by ECC and RSA, with RSA recording the poorest outcome.

2. Reasoning: More devices on the network result in increased traffic and network collisions. Because it reduces delays and almost eliminates packet loss through retransmissions, AES is especially well-suited for low delay systems at frequencies.
3. Implication: In IoT settings like alarm systems and smart grid networks, where each packet must be delivered, AES is quite helpful.

**b. Packet Loss vs Encryption Sizes:**

1. Observation: The findings suggest that the rise in encryption sizes is the cause of the total packet loss. RSA has the most packet loss, whereas AES has the lowest.
2. Reasoning: More processing time is needed for higher encryption ponds, which makes packet losses worse during network congestion. Long response times are a result of RSA's inefficiency, which exacerbates this.
3. Implication: Higher encryption levels take longer to process, which makes packet losses worse when there is network congestion. RSA's inefficiency, which results in lengthy reaction times, exacerbates this.

**c. Packet Loss vs Number of Hops:**

1. Observation: As the number of layers increases, more packets are lost. On one end of the spectrum, RSA has the largest loss, while AES has the least.
2. Reasoning: Multi-hop networks have cumulative packet loss due to the intermediary processing of the data packets. The effectiveness of AES reduces these kinds of losses.
3. Implication: In multi-hop networks, such those required for smart agriculture or disaster relief, AES is the most reliable communication method.

#### 4.2.5 Overall Comparative Insights

AES performs well in various performance indicators, with high throughput, low latency and very low packet loss rate, while showing strong scalability with no significant performance degradation when the number of devices, encryption scale or relay hop number increases, and is ideal for most IoT applications. Although RSA provides higher security, its low throughput, high latency, and high packet loss rate make it difficult to meet the requirements of real-time or high-traffic scenarios, and it is more suitable for scenarios with high security requirements. ECC strikes a balance between performance and security, has low resource requirements, and is suitable for resource-constrained low-power devices and large-scale IoT networks. Because ECC is inferior to AES in throughput and speed, it is also not ideal in real-time or high-traffic scenarios. AES is suitable for application scenarios that require efficient data processing, RSA is suitable for specific tasks with high security requirements, and ECC provides a performance and security solution for resource-limited scenarios.

Table 4.1 Comparison table of algorithm results

Algorithm	Throughput	Delay	Accuracy	Packet Loss Rate
AES	High efficiency, low resource consumption	Slow growth	Higher, even if the number of devices increases	Better control, even if the number of devices increases
RSA	High security, but not suitable for real-time or high-traffic networks	Rapid growth	The accuracy decreases significantly as the number of devices increases	The packet loss rate increases with the increase of the number of devices
ECC	Suitable for resource-constrained environments, but less efficient than AES	Moderate growth	Accuracy is less affected when the number of devices increases	Packet loss rate increases moderately

## **CHAPTER V**

### **CONCLUSION**

#### **5.1 INTRODUCTION**

This study systematically discusses the information security problems in smart home systems, comprehensively evaluates the encryption algorithms involved in IoT smart home, and focuses on the technical characteristics, application scenarios and performance of AES, RSA and ECC three mainstream encryption algorithms. Through theoretical analysis and experimental verification, the following conclusions are drawn. Through detailed technical analysis and experimental verification, this study provides a scientific basis for the future technical improvement and application of smart home system. The research results not only reveal the advantages and limitations of AES, RSA and ECC in the smart home environment, but also propose algorithm selection recommendations for different application scenarios, which will help promote the security, reliability and user trust of the smart home ecosystem.

#### **5.2 EXPERIMENTAL SUMMARY**

Through simulation experiments, the performance of three encryption algorithms in terms of throughput, delay, accuracy and packet loss rate is evaluated. AES algorithm has become the most widely used symmetric encryption algorithm in smart home system because of its efficient encryption performance and flexible key length options. Due to its high throughput and low delay, AES is ideal for applications that require real-time and efficient data processing, such as real-time control of smart home devices, high-speed transmission of sensor data, and encrypted transmission of video surveillance data. AES's efficiency and resource-friendly design make it superior in resource-constrained IoT devices. RSA is relatively inefficient at processing complex data, while ECC shows significant advantages in low-energy devices. RSA has high

delay due to its complex key generation and decryption process, while AES and ECC perform well in scenarios with low delay requirements. All three algorithms can guarantee high data accuracy, but ECC has the lowest packet loss rate in low-power environment. RSA algorithm is based on large prime number decomposition and has high security. It is widely used in scenarios that require strong identity authentication, such as user authentication and sensitive data transmission in smart homes. Its strong security and reliable authentication capabilities make it particularly suitable for use in scenarios where a secure connection needs to be established and verified, such as secure communication between smart home gateways and cloud services, user authentication, and the secure exchange of critical data. ECC algorithm achieves the same or even higher security than RSA with a smaller key length, while having lower computational complexity and power consumption, so it is especially suitable for devices with extremely limited resources, such as low-power sensors and wearable devices. ECC has significant advantages in distributed IoT applications due to the optimization of computing and storage requirements, but its real-time and performance are still slightly inferior to AES, so it should be selected according to the specific application needs. In addition, ECC's potential resistance to future quantum computing threats also makes it a strong choice for long-term security needs.

### **5.3 CHALLENGES AND COUNTERMEASURES**

The design of encryption algorithms faces multiple challenges, especially in resource-constrained environments such as the IoT and smart homes. Algorithms need to strike a balance between security and performance against brute force attacks, side-channel attacks, and quantum computing threats, while adapting to the computing power and energy consumption constraints of the device. With the continuous development of attack technology, algorithms need to be able to cope with emerging threats and solve security risks in key generation, storage, and distribution. In terms of scalability, the algorithm needs to be able to maintain performance as the number of devices and the amount of data increases, while ensuring interoperability across heterogeneous hardware and communication protocols. The algorithm also needs to comply with regulations and industry standards, meet the user's demand for efficient and convenient

experience, and ensure the reliability and practicality of the algorithm in various scenarios through comprehensive verification and testing.

The high interconnection and complex communication mode of the devices in the smart home system bring multiple security threats. The research concludes that it is very important to adopt hierarchical security strategy to protect the system comprehensively. At the transport layer, AES-256 or ECC can be selected to ensure the confidentiality and integrity of data during transmission. Implement fine-grained access control and multi-factor authentication mechanism at the application layer (such as RSA authentication scheme combining biometric and public-private key mechanism). Future research can begin with the development of quantum attack-resistant encryption algorithms that can withstand quantum computing threats (such as lattice theory or hash graph based encryption techniques) to deal with potential threats from future computing power increases. Optimize existing encryption algorithms, reduce computing overhead and energy consumption to adapt to more resource-constrained devices, and promote the development of lightweight encryption technology. On the premise of maintaining high security, we should explore how to reduce the complexity of user operations, optimize the user experience, and further improve the ease of use and user acceptance of smart home systems.

## REFERENCES

- Abdulla, L., Mahmood, M., Salih, A. & Karim, S. 2021. Analysis and evaluation of symmetric key ciphers for internet of things smart home. *Indonesian Journal of Electrical Engineering and Computer Science* 22(2):
- Abu-Tair, M., Djahel, S., Perry, P., Scotney, B., Zia, U., Carracedo, J. M. & Sajjad, A. 2020. Towards secure and privacy-preserving IoT enabled smart home: architecture and experimental study. *Sensors* 20(21): 6131.
- Acar, A., Fereidooni, H., Abera, T., Sikder, A. K., Miettinen, M., Aksu, H., Conti, M., Sadeghi, A.-R. & Uluagac, S. 2020. Peek-a-boo: I see your smart home activities, even encrypted! *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp.207-218.
- Ahmed, N. & Khan, M. Z. R. 2021. A secure IoT based grid-connected inverter using RSA algorithm. *2021 31st Australasian Universities power engineering conference (AUPEC)*, pp.1-5.
- Alasmary, H. & Tanveer, M. 2023. ESCI-AKA: Enabling secure communication in an iot-enabled smart home environment using authenticated key agreement framework. *Mathematics* 11(16): 3450.
- Alharbi, A. R., Aljaedi, A., Aljuhni, A., Alghuson, M. K., Aldawood, H. & Jamal, S. S. 2024. Evaluating Ascon Hardware on 7-series FPGA Devices. *Ieee Access*:
- Alzahrani, A. J. 2023. A New Compact-Data Encryption Standard (NC-DES) Algorithm Security and Privacy in Smart City. *International Conference on Micro-Electronics and Telecommunication Engineering*, pp.769-784.
- Bagha, A. M., Woungang, I., Dhurandher, S. K. & Traore, I. 2020. A rsa-biometric based user authentication scheme for smart homes using smartphones. *Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, pp.845-857.
- Chesuh, L. N., Fernández-Díaz, R. Á., Alija-Perez, J. M., Benavides-Cuellar, C. & Alaiz-Moreton, H. 2024. Improve quality of service for the Internet of Things using blockchain & machine learning algorithms. *Internet of Things* 26: 101123.
- Fadhil, M. S., Farhan, A. K. & Fadhil, M. N. 2021. A lightweight aes algorithm implementation for secure iot environment. *Iraqi Journal of Science*: 2759-2770.
- Fotohi, R. & Aliee, F. S. 2021. Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT. *Computer Networks* 197: 108331.
- Hasan, M. K., Shafiq, M., Islam, S., Pandey, B., Baker El-Ebiary, Y. A., Nafi, N. S., Ciro Rodriguez, R. & Vargas, D. E. 2021. Lightweight cryptographic algorithms

for guessing attack protection in complex internet of things applications. *Complexity* 2021(1): 5540296.

Islam, T., Youki, R. A., Chowdhury, B. R. & Hasan, A. T. 2021. An ECC based secure communication protocol for resource constraints IoT devices in smart home. *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*, pp.431-444.

Kavitha, A., Rao, B. S., Akthar, N., Rafi, S. M., Singh, P., Das, S. & Manikandan, G. 2022. A Novel Algorithm to Secure Data in New Generation Health Care System from Cyber Attacks Using IoT. *International Journal of Electrical and Electronics Research (IJEER)* 10(2): 270-275.

Kumar, M., Mukherjee, P., Verma, S., Kavita, Kaur, M., Singh, S., Kobielnik, M., Woźniak, M., Shafi, J. & Ijaz, M. F. 2022. BBNSF: Blockchain-based novel secure framework using RP2-RSA and ASR-ANN technique for IoT enabled healthcare systems. *Sensors* 22(23): 9448.

Lohachab, A. 2018. Using quantum key distribution and ECC for secure inter-device authentication and communication in IoT infrastructure. *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)*, pp.26-27.

Lukesh, S., Jagadesh, J. & Pandimurugan, V. 2024. Secure Smart Home Data by Using Blockchain Technology with Hash Algorithm. *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, pp.354-361.

Magyari, A. & Chen, Y. 2024. Securing the Internet of Things with Ascon-Sign. *Internet of Things* 28: 101394.

Mamvong, J. 2023. Efficient security algorithm for provisioning constrained internet of things (iot) devices.

Medileh, S., Laouid, A., Euler, R., Bounceur, A., Hammoudeh, M., Alshaikh, M., Eleyan, A. & Khashan, O. A. 2020. A flexible encryption technique for the internet of things environment. *Ad Hoc Networks* 106: 102240.

Mousavi, S. K., Ghaffari, A., Besharat, S. & Afshari, H. 2021. Security of internet of things based on cryptographic algorithms: a survey. *Wireless Networks* 27(2): 1515-1555.

Nyangaresi, V. O. 2021. ECC based authentication scheme for smart homes. *2021 International Symposium ELMAR*, pp.5-10.

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

- Phuc, C. H., Phuong, N. T. H., Van Dung, N., Nam, N. H., Chau, D. S. T. & Duc, D. N. M. 2020. Research Article Designing Efficient Smart Home Management with IoT Smart Lighting: A Case Study.
- Pirbhulal, S., Zhang, H., E Alahi, M. E., Ghayvat, H., Mukhopadhyay, S. C., Zhang, Y.-T. & Wu, W. 2016. A novel secure IoT-based smart home automation system using a wireless sensor network. *Sensors* 17(1): 69.
- Popoola, O., Rodrigues, M., Marchang, J., Shenfield, A. & Popoola, J. Hybrid Encryption for Smart Home Healthcare: Ensuring Data Confidentiality and Security.
- Popoola, O., Rodrigues, M. A., Marchang, J., Shenfield, A., Ikpehai, A. & Popoola, J. 2024. An optimized hybrid encryption framework for smart home healthcare: Ensuring data confidentiality and security. *Internet of Things* 27: 101314.
- Radhi, B. M. & Hussain, M. A. 2023. Smart Building Security using ESP32 based AES One Bio-key and Owner's Biometrics Encryption Technology. *J. Basrah Res.(Sci.)* 49(2): 30-47.
- Rahman, Z., Yi, X., Billah, M., Sumi, M. & Anwar, A. 2022. Enhancing AES using chaos and logistic map-based key generation technique for securing IoT-based smart home. *Electronics* 11(7): 1083.
- Rahul, R., Venkatesan, R. & Jebaseeli, T. J. 2024. Smart Farming with Improved Security using Ascon Encryption and Authentication. *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp.365-373.
- Rana, A., Rana, S., Bali, V., Das, R., Muduli, D., Dewan, R. & Singh, A. 2024. Comprehensive analysis of services towards Data Aggregation, Data Fusion and enhancing security in IoT-based smart home. *EAI Endorsed Transactions on Internet of Things* 10(1):
- Salim, K. G., Al-Alak, S. M. K. & Jawad, M. J. 2021. Improved image security in internet of thing (IoT) using multiple key AES. *Baghdad Science Journal* 18(2): 0417-0417.
- Santa, R. M. & Ariza, F. M. S. H. M. 2019. Secure information transmission device implemented on an embedded system using 3DES and AES algorithms. *International Journal of Engineering Research and Technology* 12(1950-1956): 32.
- Santos Jr, C. E., Silva, L. M. D., Torquato, M. F., Silva, S. N. & Fernandes, M. A. 2024. SHA-256 Hardware Proposal for IoT Devices in the Blockchain Context. *Sensors* 24(12): 3908.
- Setiawan, F. B. 2021. Securing data communication through MQTT protocol with AES-256 encryption algorithm CBC mode on ESP32-based smart homes. *2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE)*, pp.166-170.

- Singh, S., Sharma, P. K., Moon, S. Y. & Park, J. H. 2024. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*: 1-18.
- Tchagna Kouanou, A., Tchito Tchappa, C., Sone Ekonde, M., Monthe, V., Mezatio, B. A., Manga, J., Simo, G. R. & Muhozam, Y. 2022. Securing data in an internet of things network using blockchain technology: smart home case. *SN Computer Science* 3(2): 167.
- Tihanyi, N. 2022. Report on the first DES fixed points for non-weak keys: Case-study of hacking an IoT environment. *Ieee Access* 10: 77802-77809.
- Ul Islam, M., Nazish, M., Sultan, I. & Tariq Banday, M. 2024. ASCON Lightweight Security Standard for the Internet of Things Devices—A Study. *International Conference On Innovative Computing And Communication*, pp.503-517.
- Ullah, B., Mateen, A., Khalid, A., Ullah, S., Adnan, R. & Iqbal, J. 2022. Enhanced RSA Algorithm for Data Security in the Internet of Things.
- Ullah, R., Bazai, S. U., Aslam, U. & Shah, S. a. A. 2023. Utilizing blockchain technology to enhance smart home security and privacy. *Proceedings of International Conference on Information Technology and Applications: ICITA 2022*, pp.491-498.
- Wang, X., Teng, Y., Chi, Y. & Hu, H. 2022. A robust and anonymous three-factor authentication scheme based ECC for smart home environments. *Symmetry* 14(11): 2394.
- Yusoff, Z. Y. M., Ishak, M. K., Rahim, L. A. & Ali, O. 2022. Elliptic curve cryptography based security on mqtt system for smart home application. *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp.1-4.
- Zahan, A., Hossain, M. S., Rahman, Z. & Shezan, S. 2020. Smart home IoT use case with elliptic curve based digital signature: an evaluation on security and performance analysis. *International Journal of Advanced Technology and Engineering Exploration* 7(62): 11-19.
- Zeadally, S., Das, A. K. & Sklavos, N. 2021. Cryptographic technologies and protocol standards for Internet of Things. *Internet of Things* 14: 100075.
- Zhang, L. & Wang, L. 2024. A hybrid encryption approach for efficient and secure data transmission in IoT devices. *Journal of Engineering and Applied Science* 71(1): 138.
- Zou, S., Cao, Q., Wang, C., Huang, Z. & Xu, G. 2021. A robust two-factor user authentication scheme-based ECC for smart home in IoT. *IEEE Systems Journal* 16(3): 4938-4949.