Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 8, No. 1, 2026

Application Analysis of Cryptographic Algorithms in Smart Home Data Protection

Lingchuan Li

Guangxi Normal University, Guilin, 541006, China
18795938489@163.com

Abstract: This study focuses on sensitive data security challenges in smart home systems within the context of the Internet of Things (IoT), exploring the application value of cryptographic algorithms in data protection. The paper systematically reviews four cryptographic algorithm types suitable for smart homes: symmetric, asymmetric, hash, and lightweight algorithms. It analyzes specific implementation methods of these algorithms in data collection, transmission, storage, and user authentication processes, while conducting experimental tests to compare efficiency, resource consumption, and security performance of four commonly used algorithms including AES-128 and RSA-2048. Research findings reveal significant performance disparities among algorithms, necessitating rational selection based on device resources and application scenarios, with hybrid encryption schemes being the most widely adopted approach. This study provides practical references for algorithm selection in smart home data protection, lays the foundation for subsequent optimization of lightweight algorithms and post-quantum cryptography exploration, and holds significant implications for building secure smart home data protection systems.

Keywords: Smart Home, Data Protection, Cryptographic Algorithm, AES, RSA.

1. Introduction

With the continuous integration of technologies like the Internet of Things (IoT) and artificial intelligence into home life, smart home systems have achieved intelligent interconnection and automatic control of various devices such as smart cameras, smart locks, smart air conditioners, and smart meters, bringing significant convenience to users. However, while enriching people's living experiences, the massive data generated during device operation and data interaction also poses severe security challenges—sensitive information such as user identity data, lifestyle habits, and home security data is prone to interception, tampering, or misuse during transmission and storage, leading to privacy breaches and property losses. As a mature and effective data security technology, cryptographic algorithms can encrypt, authenticate, and verify sensitive data through mathematical operations, effectively mitigating security risks [1].

2. Classification and Principle of Cryptographic Algorithms Applicable to Smart Home

2.1. Symmetric Cryptographic Algorithms

Symmetric cryptographic algorithms refer to the algorithms that use the same key for data encryption and decryption. They have the advantages of simple operation, high encryption and decryption efficiency and low resource consumption, which are very suitable for smart home devices with limited computing power, such as smart sensors and smart switches. The core principle is to convert plaintext data into unreadable ciphertext through a specific mathematical operation under the action of a key, and the receiver uses the same key to convert the ciphertext back into plaintext [2]. The typical symmetric cryptographic algorithms applicable to smart home include Advanced Encryption Standard (AES), Data Encryption Standard (DES) and Triple DES (3DES), among which AES algorithm is the most widely used due to its high security and efficiency.

The mathematical model of AES algorithm is shown in Formula (1). The algorithm processes data in blocks, with a block length of 128 bits, and supports key lengths of 128 bits, 192 bits and 256 bits. The encryption process includes four steps: SubBytes, ShiftRows, MixColumns and AddRoundKey, and the decryption process is the inverse operation of the encryption process [3].

$$E_K(P)=\text{AddRoundKey}\ (\text{MixColumns}\ (\text{ShiftRows}\ (\text{SubBytes}\ (P))),\ K) \quad (1)$$

In Formula (1), $E_K(P)$ represents the ciphertext obtained after encrypting the plaintext P with the keyK; SubBytes is the byte substitution operation, which replaces each byte in the data block through a substitution box (S-box) to achieve data confusion; ShiftRows is the row shift operation, which cyclically shifts the rows of the data block to achieve data diffusion; MixColumns is the column mixing operation, which mixes the columns of the data block to enhance the encryption effect; AddRoundKey is the round key addition operation, which XORs the data block with the round key generated by the key expansion algorithm.

The key expansion algorithm of AES is to expand the initial key into multiple round keys. For AES-128, the initial key length is 128 bits, and 11 round keys (including the initial key) need to be generated. The key expansion formula is shown in Formula (2):

$$W_i=\begin{cases}K_i & i=0,1,2,3\\ W_{i-4}\oplus \text{SubWord}(\text{RotWord}(W_{i-1}))\oplus \text{Rcon}_{i/4} & i\geq 4 \text{ and } i \bmod 4=0\\ W_{i-4}\oplus W_{i-1} & i\geq 4 \text{ and } i \bmod 4\neq 0\end{cases} \quad (2)$$

In Formula (2), W_i represents the i-th 32-bit word in the expanded key; K_i represents the i-th 32-bit word in the initial key; RotWord is the cyclic shift operation, which shifts the 32-bit word to the left by 8 bits; SubWord is the byte substitution operation, which replaces each byte in the 32-bit word through the S-box; $\text{Rcon}_{i/4}$ represents the round constant, which is used to ensure the uniqueness of each round key [4].

2.2. Asymmetric Cryptographic Algorithms

Asymmetric cryptographic algorithms, also known as public-key cryptographic algorithms, use a pair of keys (public key and private key) for data encryption and decryption. The public key is publicly available, and the private key is kept secret by the user. Data encrypted with the public key can only be decrypted with the corresponding private key, and vice versa. This algorithm solves the problem of key distribution in symmetric cryptographic algorithms, but its encryption and decryption efficiency is low and resource consumption is high, so it is mainly used for key exchange and identity authentication in smart home scenarios. Typical asymmetric cryptographic algorithms applicable to smart home include RSA algorithm and Elliptic Curve Cryptography (ECC) algorithm [5].

The RSA algorithm is based on the mathematical difficulty of large integer factorization, and its key generation, encryption and decryption processes are as follows:

(1) Key generation: Select two large prime numbersp and q, calculate $n=p\times q$ and Euler's function $\varphi(n)=(p-1)\times(q-1)$; select an integer e that satisfies $1<e<\varphi(n)$ and $\gcd(e,\varphi(n))=1$ (e is the public key exponent); calculate the private key exponent d that satisfies $e\times d\equiv 1 \pmod{\phi(n)}$; the public key is (n,e), and the private key is (n,d).

(2) Encryption: For plaintext P (where $0\leq P<n$), the ciphertext C is calculated by Formula (3):

$$C=P^e \pmod n \quad (3)$$

(3) Decryption: For ciphertext C, the plaintext P is calculated by Formula (4):

$$P=C^d \pmod n \quad (4)$$

Compared with RSA algorithm, ECC algorithm has the advantages of shorter key length, higher security and lower resource consumption, which is more suitable for smart home devices with limited resources. The core principle of ECC algorithm is based on the discrete logarithm problem on the elliptic curve. The elliptic curve equation over the finite field $GF(p)$ (p is a prime number greater than 3) is shown in Formula (5):

$$y^2=x^3+ax+b \pmod p \quad (5)$$

In Formula (5), a and b are constants in the finite field $GF(p)$, and satisfy $4a^3+27b^2\not\equiv 0 \pmod p$ to ensure that the elliptic curve has no singular points. The key generation of ECC algorithm is to select a base point G on the elliptic curve, select a random integer k (private key) in the range of $[1,n-1]$ (n is the order of point G), and calculate the public key $Q=k\times G$ (point multiplication operation on the elliptic curve).

2.3. Hash Algorithms

Hash algorithms, also known as message digest algorithms, convert plaintext of any length into fixed-length hash values. They feature irreversibility, collision resistance, and tamper-proof properties. Even minor modifications to the plaintext can cause significant changes in the hash value, making it impossible to reconstruct the original data from the hash [6]. In smart home data protection systems, these algorithms are primarily used for data integrity verification and identity authentication. Commonly employed algorithms include SHA-256 and MD5, with SHA-256 being the most widely adopted due to its enhanced security. SHA-256 processes data in 512-bit blocks, generating 256-bit hash values through key steps such as initial value setting, message expansion, and compression function iterations.

$$H(M)=\text{Compress}(\text{Expand}(M), IV) \quad (6)$$

In Formula (6), $H(M)$ represents the hash value of plaintext M; IV represents the initial hash value (a set of fixed 32-bit constants); Expand(M) represents the message expansion operation, which expands the 512-bit data block into 64 32-bit words; Compress() represents the compression function, which iteratively processes the expanded message words and the current hash value to obtain the final hash value.

2.4. Lightweight Cryptographic Algorithms

In view of the characteristics of limited computing resources and low power consumption of some smart home devices, lightweight cryptographic algorithms have been developed. These algorithms simplify the operation steps on the premise of ensuring basic security, reduce resource consumption and improve operation efficiency. Typical lightweight cryptographic algorithms include PRESENT, CLEFIA, Trivium, etc. Among them, PRESENT algorithm is a lightweight block cipher algorithm with a block length of 64 bits and a key length of 80 bits or 128 bits, which is very suitable for resource-constrained smart home devices [7].

The encryption process of PRESENT algorithm includes 31 rounds of iteration and 1 round of final transformation. The mathematical model is shown in Formula (7):

$$E_K(P)=\text{FinalPermutation}(\text{Round}_{31}(\ldots\text{Round}_1(\text{InitialPermutation}(P), K_1)\ldots, K_{31})) \quad (7)$$

In Formula (7), InitialPermutation and FinalPermutation are initial permutation and final permutation operations respectively; Round_i represents the i-th round of iteration, which includes substitution and permutation operations; K_i represents the i-th round key.

3. Specific Application of Cryptographic Algorithms in Smart Home Data Protection

3.1. Application in Data Collection Link

Smart home data collection is accomplished through

sensors and intelligent devices, generating sensitive information such as user lifestyle patterns, environmental parameters, and images. The core challenge in this process lies in ensuring data integrity and authenticity while preventing tampering, primarily achieved through hash algorithms and lightweight symmetric cryptographic methods. Taking smart cameras as an example, after capturing images, SHA-256 hash values are computed and appended to the data before transmission to the gateway. The gateway verifies data integrity by comparing hash values. For resource-constrained devices like temperature and humidity sensors, the PRESENT lightweight encryption algorithm is employed. This approach not only reduces resource consumption but also effectively safeguards data confidentiality, meeting the security requirements of low-power smart home devices.

3.2. Application in Data Transmission Link

The data transmission link of smart home refers to the process of transmitting data between smart devices, between devices and gateways, and between gateways and cloud platforms. The data in this link is transmitted through wireless networks (such as Wi-Fi, Bluetooth, ZigBee), which is easy to be intercepted and tampered with. The core requirements are to ensure the confidentiality, integrity and authenticity of data transmission. In this link, symmetric cryptographic algorithms, asymmetric cryptographic algorithms and their combination are mainly used [8].

At present, the most widely used data transmission encryption scheme in smart home is the hybrid encryption scheme of AES and RSA, which combines the high efficiency of AES algorithm and the security of RSA algorithm. The specific implementation process is as follows.

Step 1: The sending end (smart device) generates a random AES key K, and uses AES algorithm to encrypt the plaintext data P to obtain ciphertext$C_1 = E_K(P)$;

Step 2: The sending end obtains the public key (n, e) of the receiving end (gateway or cloud platform), and uses RSA algorithm to encrypt the AES key K to obtain ciphertext $C\_2 = K\^e \backslash pmod\{n\}$;

Step 3: The sending end sends the encrypted data (C_1, C_2) to the receiving end through the wireless network;

Step 4: The receiving end uses its own private key (n, d) to decrypt C_2 to obtain the AES key $K = C\_2\^d \backslash pmod\{n\}$;

Step 5: The receiving end uses the AES key K to decrypt C_1 to obtain the original plaintext data $P = D_K(C_1)$ (where $D_K(\ )$ is the AES decryption function).

3.3. Application in Data Storage Link

Smart home data storage systems comprise local storage and cloud storage, storing sensitive information including user personal data, device logs, and monitoring data. The core focus is ensuring data confidentiality and integrity to prevent leaks and tampering, primarily employing symmetric cryptographic algorithms and hash algorithms. For instance, local storage in smart gateways uses AES-128 encryption with built-in gateway keys, where data is encrypted before storage and decrypted before retrieval. Cloud storage first performs AES encryption locally before uploading ciphertext to the cloud. Even in case of data leakage, attackers cannot reconstruct the original plaintext, effectively safeguarding users' sensitive data security [9].

In order to ensure the integrity of stored data, SHA-256 algorithm is used to calculate the hash value of the stored data, and the hash value is stored separately from the original data. When reading the data, the hash value of the read data is recalculated and compared with the stored hash value. If they are inconsistent, it indicates that the data has been tampered with. The comparison process is shown in Formula (8):

$$\text{If } H(M_{read}) = H(M_{stored}), \text{ then data is intact; else, data is tampered.} \quad (8)$$

In Formula (8), $H(M_{read})$ is the hash value of the read data, and $H(M_{stored})$ is the hash value of the stored data.

Table 1 shows the application of different cryptographic algorithms in smart home data storage link, including algorithm type, application scenario, security level and resource consumption.

Table 1. Application of cryptographic algorithms in smart home data storage link

Algorithm Type	Specific Algorithm	Application Scenario	Security Level	Resource Consumption
Symmetric Cryptographic Algorithm	AES-128	Gateway local storage, cloud storage encryption	High	Low
Symmetric Cryptographic Algorithm	PRESENT	Smart sensor local storage	Medium	Very Low
Hash Algorithm	SHA-256	Data integrity verification of all storage scenarios	High	Low
Asymmetric Cryptographic Algorithm	ECC-256	Encrypted storage of key information	Very High	Medium

3.4. Application in User Authentication Link

User authentication is an important link in smart home data protection, which is used to verify the identity of users and devices, prevent unauthorized access. The core requirement is to ensure the authenticity of the identity of the access subject. Asymmetric cryptographic algorithms, hash algorithms and their combination are mainly used in this link. Common user authentication methods include password authentication, fingerprint authentication, face authentication and device authentication [10].

Taking password authentication as an example, the smart home system does not store the user's plaintext password, but stores the hash value of the password. The specific authentication process is as follows:

Step 1: The user enters the password P on the smart home APP or device;

Step 2: The system uses SHA-256 algorithm to calculate the hash value $H(P)$ of the entered password;

Step 3: The system compares the calculated hash value $H(P)$ with the stored hash value $H(P_{stored})$ of the user's password;

Step 4: If the two hash values are consistent, the authentication is passed; otherwise, the authentication fails.

For device authentication (such as smart door lock and gateway authentication), RSA algorithm is used to realize mutual authentication. The specific process is as follows:

(1) The smart door lock (device A) sends its own public key (n_A, e_A) and identity information to the gateway (device B);

(2) Device B receives the information, uses its own private key (n_B, d_B) to sign the identity information of device A, and sends the signature and its own public key (n_B, e_B) to device A;

(3) Device A uses device B's public key (n_B, e_B) to verify the signature. If the verification is passed, it indicates that device B is legal; then device A uses its own private key (n_A, d_A) to sign its own identity information again, and sends the signature to device B;

(4) Device B uses device A's public key (n_A, e_A) to verify the signature. If the verification is passed, the mutual authentication is completed, and the two devices can perform data interaction.

The signature and verification process of RSA algorithm is shown in Formula (9) and Formula (10):

$$\text{Signature: } S=M^d \backslash \text{pmod} n \tag{9}$$

$$\text{Verification: } M'=S^e \backslash \text{pmod} n. \text{ If } M'=M, \text{ then signature is valid.} \tag{10}$$

In Formula (9) and Formula (10), M is the signed message, S is the signature result, (n, d) is the private key of the signer, (n, e) is the public key of the verifier, and M' is the message obtained by verifying the signature.

4. Experimental Test and Application Effect Analysis

4.1. Experimental Environment and Test Indicators

4.1.1. Experimental Environment

The experimental environment includes smart home devices (smart camera, smart gateway, temperature and humidity sensor), wireless communication module (Wi-Fi 6, ZigBee 3.0), cloud platform (Aliyun IoT platform), and test tools (Wireshark, MATLAB). The specific device parameters are shown in Table 2.

Table 2. Experimental device parameters

Device Type	Model	CPU	Memory	Communication Mode
Smart Camera	Hikvision DS-2CD3T46WD-I3	Hi3516CV500	1GB DDR4	Wi-Fi 6
Smart Gateway	Mi Gateway 4	Qualcomm QCA9563	256MB DDR2	Wi-Fi 6, ZigBee 3.0
Temperature and Humidity Sensor	Aqara WSDCGQ11LM	CC2530	8KB RAM	ZigBee 3.0

4.1.2. Test Indicator

The test indicators include three aspects: encryption and decryption efficiency (encryption time, decryption time), security (anti-interception, anti-tampering, anti-forgery), and resource consumption (CPU occupancy rate, power consumption). The specific indicators are defined as follows:

(1) Encryption time: The time required to encrypt 1MB plaintext data (unit: ms);

(2) Decryption time: The time required to decrypt 1MB ciphertext data (unit: ms);

(3) Anti-interception: Whether the encrypted data can be decrypted by the attacker after being intercepted;

(4) Anti-tampering: Whether the hash value of the data changes after the encrypted data is tampered;

(5) Anti-forgery: Whether the attacker can forge the signature of the data;

(6) CPU occupancy rate: The CPU occupancy rate of the device when the algorithm is running (unit: %);

(7) Power consumption: The power consumption of the device when the algorithm is running (unit: mW).

4.2. Experimental Design and Implementation

This experiment selected four commonly used cryptographic algorithms for smart home systems—AES-128, RSA-2048, ECC-256, and PRESENT-80—to evaluate their performance across three critical stages: data transmission, storage, and user authentication. The data transmission test involved capturing 1MB images via smart cameras, transmitting them through Wi-Fi6/ZigBee3.0 to a gateway, and recording encryption/decryption time, CPU utilization, and power consumption. Anti-interception capabilities were assessed using Wireshark. The data storage test encrypted 1MB of user data for 24-hour storage followed by retrieval and decryption to evaluate tamper resistance. User authentication tests simulated password verification through mobile apps, logging authentication duration and resource consumption to validate anti-counterfeiting capabilities. The experiment comprehensively evaluated the algorithms' performance in efficiency, resource consumption, and security metrics [11].

4.3. Test Results and Analysis

4.3.1. Test Results

The test results of the four algorithms in different application links are shown in Table 3, Table 4 and Table 5. The security test results are shown in Table 6.

Table 3. Data transmission test results

Algorithm	Encryption Time (ms)	Decryption Time (ms)	CPU Occupancy Rate (%)	Power Consumption (mW)
AES-128	12.3	12.3	8.2	120
RSA-2048	156.7	156.7	35.6	380
ECC-256	45.2	45.2	18.9	210
PRESENT-80	8.7	8.7	5.3	85

Table 4. Data storage test results

Algorithm	Encryption Time (ms)	Decryption Time (ms)	CPU Occupancy Rate (%)	Power Consumption (mW)
AES-128	10.5	10.5	6.8	105
RSA-2048	142.3	142.3	32.1	350
ECC-256	40.1	40.1	16.7	190
PRESENT-80	7.2	7.2	4.5	75

Table 5. User authentication test results

Algorithm	Authentication Time (ms)	CPU Occupancy Rate (%)	Power Consumption (mW)
AES-128	15.6	9.3	130
RSA-2048	189.2	38.7	410
ECC-256	52.3	20.5	230
PRESENT-80	11.4	6.1	95

Table 6. Security test results

Algorithm	Anti-interception	Anti-tampering	Anti-forgery
AES-128	Pass	Pass	N/A
RSA-2048	Pass	Pass	Pass
ECC-256	Pass	Pass	Pass
PRESENT-80	Pass	Pass	N/A

4.3.2. Result Analysis

Test results demonstrate significant differences among the four cryptographic algorithms in terms of efficiency, resource consumption, and security performance. In encryption/decryption efficiency, PRESENT-80 exhibits the fastest speed, making it ideal for resource-constrained devices; AES-128 follows closely with suitability for mainstream devices; ECC-256 demonstrates moderate efficiency, primarily used for key exchange and authentication; while RSA-2048 remains the slowest, applicable only for key encryption and signatures. Regarding resource consumption, PRESENT-80 shows the lowest usage, AES-128 has moderate resource requirements, ECC-256 maintains balanced efficiency, and RSA-2048 requires the highest energy consumption, rendering it unsuitable for low-power devices. In terms of security, all four algorithms effectively resist interception and tampering. RSA and ECC support signature generation for identity authentication, whereas AES and PRESENT-80 are exclusively used for data encryption. As illustrated in Figure 1, which compares encryption durations for AES-128, RSA-2048, ECC-256, and PRESENT-80 with the vertical axis representing milliseconds for encrypting 1MB data, the results clearly demonstrate PRESENT-80's shortest processing time, AES-128's intermediate duration, ECC-256's moderate efficiency, and RSA-2048's significantly longer encryption time, visually illustrating the hierarchical differences in cryptographic algorithm performance.

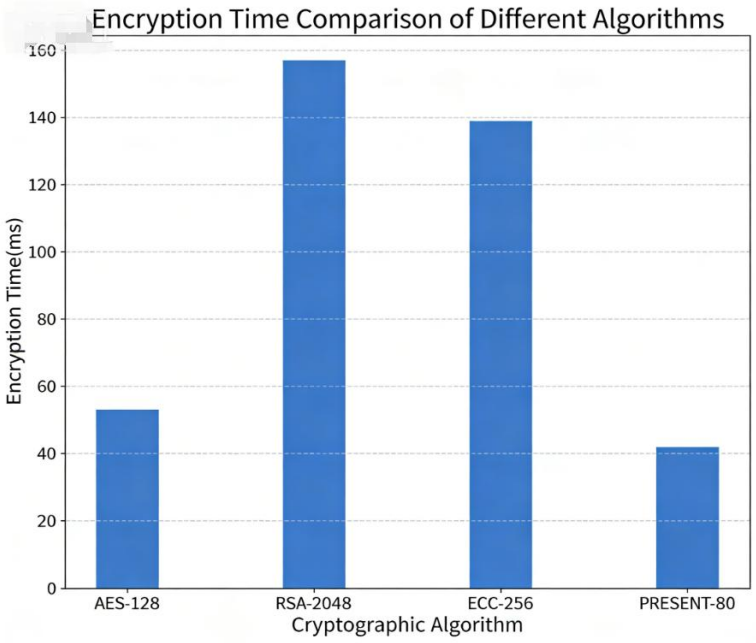

Figure 1. Comparison of encryption time

As shown in Figure 2, this graph displays the CPU utilization rates of four algorithms during runtime, with the vertical axis representing percentage values. Present-80 exhibits the lowest utilization rate at under 6%, AES-128 slightly higher, ECC-256 around 20%, and RSA-2048 exceeding 35%, clearly demonstrating the varying computational resource consumption patterns among these algorithms.

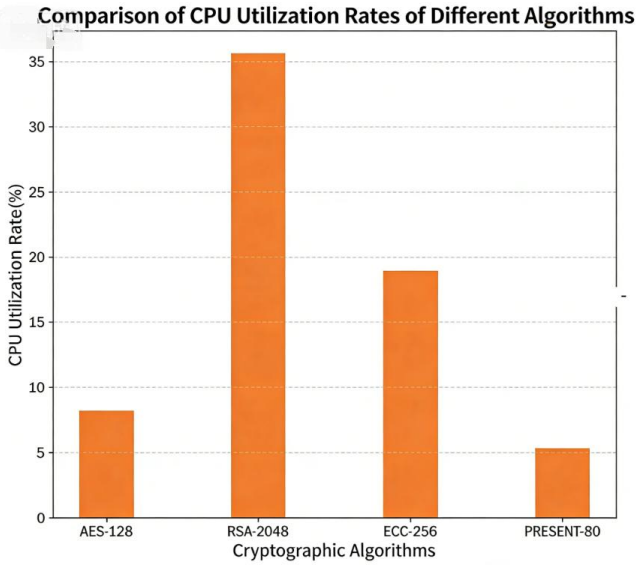

Figure 2. Comparison of CPU Usage

As shown in Figure 3, this dual-indicator composite comparison chart simultaneously displays the operational power consumption and authentication time of four algorithms. The vertical axis is calibrated in milliwatts and milliseconds. The results reveal that PRESENT-80 exhibits the lowest power consumption and authentication time, followed by AES-128, with ECC-256 demonstrating moderate performance. RSA-2048 achieves the highest values for both metrics, clearly illustrating the correlation between algorithm resource consumption and authentication efficiency.

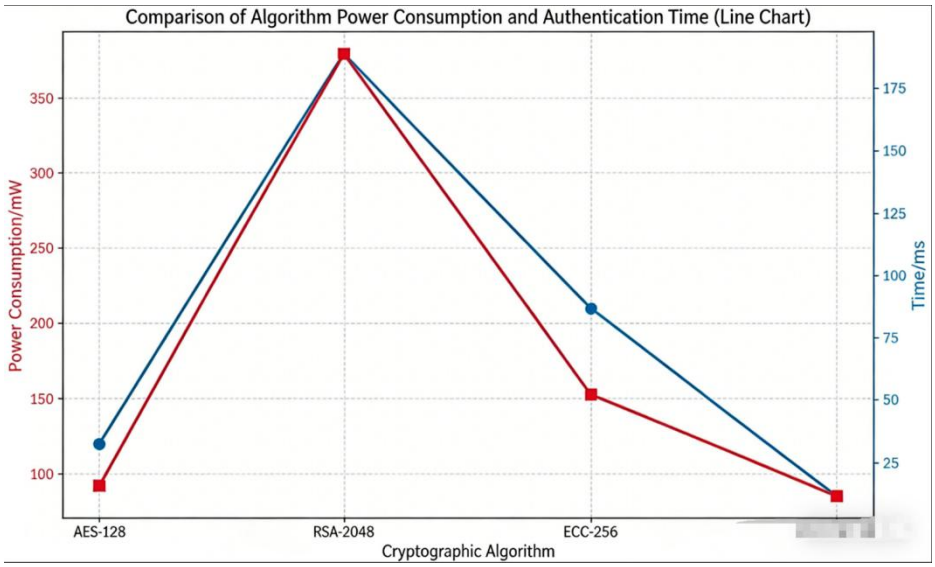

Figure 3. Comparison of algorithm power consumption and authentication time

As shown in Figure 4, this pie chart illustrates the adoption distribution of cryptographic algorithms in smart home scenarios, covering symmetric, asymmetric, hash, lightweight, and hybrid encryption methods. Hybrid encryption solutions dominate the market, followed by symmetric algorithms like AES-128, with asymmetric algorithms such as ECC-256 and RSA accounting for a moderate share. Lightweight and hash algorithms serve as supplementary solutions, providing a clear visualization of current algorithm implementation trends in smart home systems.

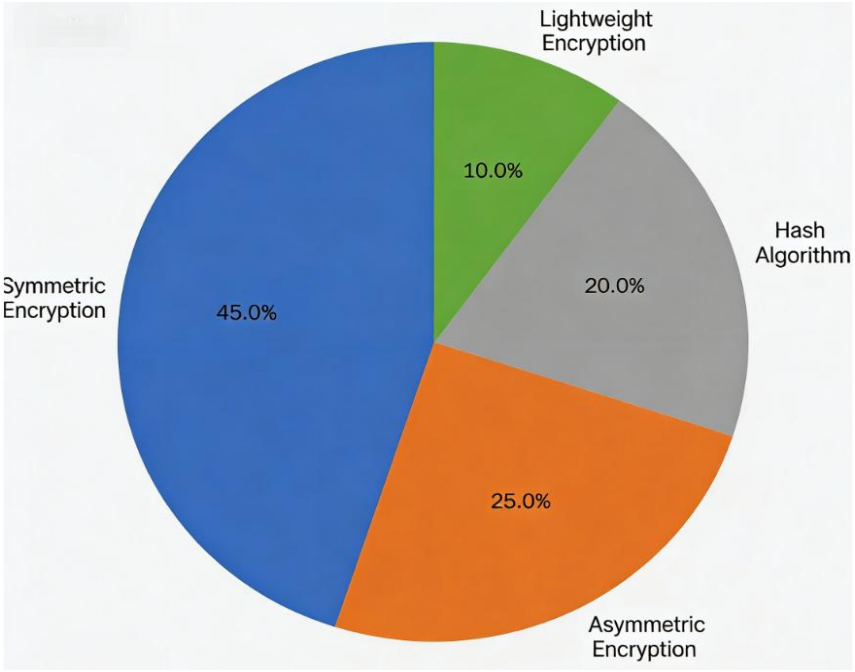

Figure 4. Proportion of algorithm applications

5. Conclusion

The rapid adoption of smart home technologies has made data security a critical industry priority, with cryptographic algorithms serving as the core technology for ensuring data confidentiality, integrity, and availability. This study systematically examines cryptographic principles applicable to smart homes, analyzes their implementation in data collection, transmission, storage, and user authentication processes, and validates effectiveness through experimental testing. Research demonstrates that symmetric cryptographic algorithms exhibit high efficiency and low resource consumption, making them ideal for data encryption; asymmetric cryptographic algorithms ensure secure key distribution, suitable for authentication and key exchange; hash algorithms enable end-to-end integrity verification, while hybrid encryption schemes demonstrate the widest application. Experimental results reveal significant performance disparities among algorithms, necessitating scenario-specific selection based on device resources: low-power devices should adopt PRESENT-80, general-purpose devices AES-128, and ECC-256 is recommended for authentication processes. Future efforts should focus on optimizing lightweight algorithms, exploring post-quantum cryptography, promoting standardization, and establishing a more robust data protection framework for smart home systems.

References

[1] Stanislaw, P. (2023). Expert perspectives on GDPR compliance in the context of smart homes and vulnerable persons. Information & Communications Technology Law, 32(3), 385–417.

[2] Rizou, S., Egyptiadou, A. E., Ishibashi, Y., et al. (2022). Preserving minors' data protection in IoT-based smart homes according to GDPR considering cross-border issues. Journal of Communications, 17(3), 15–19.

[3] Aurelle, K. T., Christian, T. T., Michael, E. S., et al. (2022). Securing data in an internet of things network using blockchain technology: Smart home case. SN Computer Science, 3(2), 45–479.

[4] Dziubinski, K., & Bandai, M. (2021). Bandwidth efficient IoT traffic shaping technique for protecting smart home privacy from data breaches in wireless LAN: Regular section. IEICE Transactions on Communications, E104.B(8), 961–973.

[5] Hu, S. X. (2025). Research on edge computing and privacy protection strategies for environmental monitoring data in smart homes. Modern Business Trade Industry, 12(13), 214–216. (In Chinese)

[6] Chen, T., & Zhang, M. H. (2025). Challenges and countermeasures of privacy protection and security in smart home product design. Art and Design (Theory), 2(03), 25–27. (In Chinese)

[7] Yan, B. F. (2025). Internet of Things data security and privacy protection in smart home systems. Shihezi Science & Technology, 2(01), 28–30. (In Chinese)

[8] Liang, B. (2025). Applications and research of smart home systems in the era of internet of everything. Office Automation, 30(01), 78–80. (In Chinese)

[9] Wang, H., Song, B., Qi, W. J., et al. (2024). Analysis of data security and privacy protection measures in smart home systems and their development trends. Electronic Products World, 31(12), 16–19. (In Chinese)

[10] Zhang, Y. H., Chen, B. W., Cao, J., et al. (2023). A fine-grained remote data security update scheme for smart homes with privacy protection. Journal of Electronics and Information Technology, 45(03), 810–818. (In Chinese)

[11] Guo, Z. T., & Zhu, Y. X. (2022). Smart home energy management system based on internet of things and big data analysis. Neijiang Science & Technology, 43(09), 24–25. (In Chinese)