Using Combined Data Encryption and Trusted Network Methods to Improve the Network Security of the Internet of Things Applications

Yudan Zhao
Network and Information Center, Guangzhou Open University, Guangzhou, 510000, China

***Abstract*—With the integration of big data and artificial intelligence, the Internet of Things has rapidly developed as the foundation for collecting data. The data collected by Internet of Things devices is mostly sensitive information, but limited resources can easily lead to data leakage. Therefore, this study adopts a combination of data encryption and trusted networks to improve the network security of the Internet of Things. This study proposes an Internet of Things network security system based on an improved SM9 encryption algorithm and iTLS security protocol. The system uses a key generation center to generate and distribute keys to complete information encryption and decryption, and identity authentication is carried out through dynamic keys. The results indicated that the total time for key generation, encryption, and decryption based on the SM9-iTLS network security system was 3.63 seconds. The total time for key generation, signature, and signature verification in the system was 3.65 seconds, which is better than other Internet of Things network security systems, and it also had better network resource occupancy and latency than other systems. The Internet of Things network security system based on improved SM9-iTLS can not only improve the security of information transmission among Internet of Things devices but also optimize the efficiency of information transmission. The research results have a certain promoting effect on developing the Internet of Things information security field.**

***Keywords*—Security protocols; Encryption; Internet of Things; Resource constraints; SM9**

I. Introduction

The Internet of Things (IoT) brings great convenience to people's daily lives, enterprise production and supply, etc. However, IoT devices usually use ordinary software and hardware configurations, so they are prone to exposing significant data and network security risks in open environments [1-3]. There are numerous IoT devices, but these devices have limited resources and insufficient network security protection capabilities, leading to a large number of network attacks and information leakage problems. Therefore, improving the network security protection capability of the IoT under limited resources is the main direction of research. The use of intrusion detection systems as a network security method can detect and prevent potential attacks, but with an increase in IoT devices, the difficulty of device maintenance and maintenance becomes greater, and the cost is higher [4-6]. Data encryption (DE) is a low-cost security solution that can prevent device data from being accessed or cracked by intruders, protecting the security of IoT devices when transmitting data [7-8]. Security protocols (SPs) in trusted networks can establish security mechanisms such as identity verification, key exchange, encrypted communication, auditing, and logging in network communication, effectively ensuring the security of network communication [9]. DE and SPs are widely used to ensure network security. Many scholars and experts have conducted relevant research on DE, SPs, and IoT network security (IoT-NS). Leigh proposed an encryption algorithm based on NIST-AES to address the issue of low security of confidential data in third-party HPC systems. The running time of using the NIST-AES algorithm was basically the same as that of using an ordinary encryption algorithm, and the data security of the HPC system has been improved by 34.8%. Therefore, the NIST-AES algorithm solved the data security problem of third-party HPC systems [10]. To address the security issues of voice data in centralized cloud storage, Zhang and Zhao proposed a distributed voice DE storage scheme based on IPFS and CP-HABE. This solution had reliability, security, and scalability. DE could solve data security issues in distributed communication [11]. To address the issue of IoT devices being vulnerable to attacks due to limited resources and being at the edge of the network, He et al. proposed an anonymous and lightweight authentication and key exchange protocol for IoT devices. This protocol had higher security than other SPs, and lightweight protocol deployment on IoT was feasible [12]. Mvah et al. proposed a self-executing SP based on Nash equilibrium to address the issue of attackers using artificial intelligence for ARP spoofing attacks. The simulation results showed that compared with other methods, this method could better prevent, detect, and recover ARP spoofing attacks. This protocol could solve the problem of ARP spoofing attacks and effectively ensure network security [13]. Rathee et al. designed a dynamic Pub/Sub communication mechanism to address communication trust and security issues during message publishing and subscription processes in environmental intelligence. This mechanism was more secure than traditional security measures [14]. Chowdhury et al. developed a network security tool based on device fingerprints to address the vulnerability of IoT devices to Mirai botnet and spoofing attacks in communication. The recognition accuracy of this scheme for the UNSW dataset was 99.81%, and the tool could enhance network security in heterogeneous network environments [15].

The above research indicates that in the case where traditional intrusion detection systems have application limitations, many scholars have conducted relevant research on DE, SPs, and IoT-NS. However, there is a paucity of research exploring the integration of DE and SPs for IoT-NS. Based on existing research, it is known that both DE algorithms and SPs have the effect of strengthening network security protection. Therefore, this study combines the two to construct a lightweight and high-security IoT-NS protection model. This study designs an improved SM9 encryption algorithm and iTLS SP for IoT network communication and constructs an IoT communication system with dual layer security protection of communication encryption and identity verification. The innovation of the research lies in the cancellation of certificate authentication for both encryption algorithms and SPs. The encryption algorithm uses a private key generator (PKG) to generate and distribute private keys and completes encryption and decryption through private key calculation. The SP completes the handshake protocol through dynamic password calculation. This improved encryption and identity authentication mechanism not only enhances network security but also increases communication speed and occupies fewer resources.

The research content mainly consists of five sections. Section I is the introduction, which analyzes the research achievements in IoT-NS and briefly describes the proposed IoT-NS protection model. Section II is based on the algorithm and mechanism of the improved SM9-iTLS IoT-NS system. Section III tests the research model. Section IV discusses the experimental results. Section V summarizes the research findings.

II. Methods and Materials

The SM9 encryption algorithm uses highly secure asymmetric encryption technology, but the certificate mechanism of SM9 is prone to vulnerabilities. This study proposes an SM9 algorithm for certificate free encryption and signature. In addition, a lightweight TLS communication protocol has been introduced and optimized, using dynamic keys in the handshake protocol to enhance the security of IoT information transmission through identity authentication. This study combines the improved SM9 encryption algorithm with iTLS SP to construct a secure and low-latency IoT network system [16].

A. Design of Improved SM9 Encryption Algorithm

The identity-based cryptography (IBC) system is a cryptographic technique that facilitates the deployment of asymmetric cryptographic systems. SM9 is a cryptographic algorithm in IBC. SM9 mainly includes encryption, key encapsulation, key exchange, and digital signature. The logic of SM9 utilizes the Abel group discrete logarithmic difficulty formed by the points of elliptic curves over a finite field to achieve encryption, decryption, and digital signature [17-18]. The cracking difficulty of the 3072-bit factorization algorithm is equivalent to that of the 256-bit elliptic curve algorithm, indicating that the SM9 encryption algorithm has sufficiently high security. The PKG of SM9 only generates decryption private keys and uses identity information as the public key, which is beneficial for forming lightweight network encryption. However, this form of encryption that completely entrusts the key to a third party is vulnerable to network attacks. Therefore, this study improves SM9 by designing a certificate-free encryption and signature-based SM9 encryption algorithm. In improving SM9, user public keys are no longer verified through certificates but are generated and distributed using a key generation center, which can improve the network security of IoT devices [19]. In the certificate free encryption SM9 algorithm, the encryption process is completed using two parts: a public key and user identity, while the user decryption operation is decrypted using the generated key. The PKG in improved SM9 calculates the decryption private key for the encryption task. The calculation formula for decrypting the private key is Eq. (1).

$$sk_{SM9} = s(Hash(ID \| hid) + s)^{-1} P_2 \tag{1}$$

In Eq. (1), sk_{SM9} is the decryption private key, and P_2 is the generator of the addition group G_2. PKG calculates and decrypts the private key, and the complete encryption and decryption process is shown in Fig. 1.

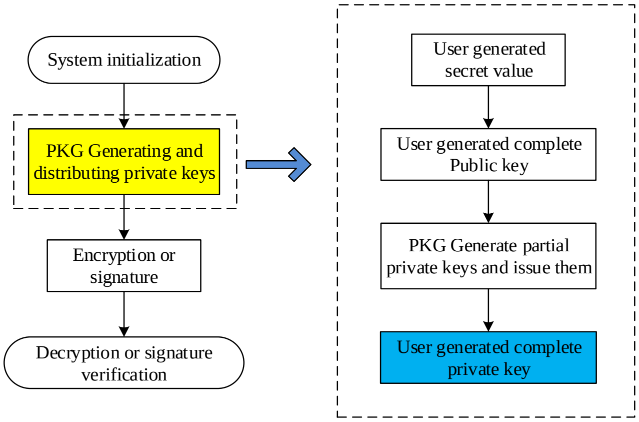

System initialization
PKG Generating and distributing private keys
Encryption or signature
Decryption or signature verification
User generated secret value
User generated complete Public key
PKG Generate partial private keys and issue them
User generated complete private key

Fig. 1. The process of PKG calculation and decryption of private keys and complete encryption / decryption.

In Fig. 1, the improved SM9 first performs system initialization, using multiplication group G_1, addition group G_2, and addition group G_2 of prime number q order, and then generates the distributed private key sk_{SM9} by PKG. The improved SM9 is encrypted based on the private key sk_{SM9}, and the encryptor completes the encryption through user ID and the public key calculated by ID. The user calculates the decrypted private key through the private key sk_{SM9} for decryption. The calculation formula for decrypting sk_{SM9} by the user is Eq. (2).

$$sk_{SM9-CLE} = x_{ID}^{-1} s(Hash(ID \| PK \| hid) + s)^{-1} P_2 \tag{2}$$

In Eq. (2), $sk_{SM9-CLE}$ is the decryption private key calculated by the user. P_2 is the generator of the additive group G_2. $Hash$ is the decryption private key generated by the user. x_{ID} is a random value. hid represents the user's key. ID represents the user's identity. s is the primary signature private key of PKG. PKG calculates the public key in the encrypted information based on the user's identity, and the calculation of the public key is Eq. (3).

$$PK = (x_{ID} P_1, x_{ID} P_{pub-e}) \tag{3}$$

In Eq. (3), PK is the public key, which is completely public, and P_1 is the generator of the additive group G_1. After receiving the public key, the sender encrypts the ciphertext based on the public key and user identity. After receiving the ciphertext, the user decrypts it through calculation to obtain the plaintext. The most important thing in decryption is to verify whether the packaging information of the encrypted calculation is consistent with the packaging information of the decrypted calculation. If they are not consistent, decryption cannot be performed. The expression to determine whether the encapsulated information is consistent is Eq. (4).

$$\begin{aligned} w' &= e(C_1, SK) \\ &= e(rx_{ID}(h_{ID}+s)P_1, {x_{ID}}^{-1}s(h_{ID}+s)^{-1}P_2) \\ &= e(P_1,P_2)^{r(h_{ID}+s)x_{ID}x_{ID}{}^{-1}}s(h_{ID}+s)^{-1} \\ &= e(P_1,P_2)^{rs} \\ &= g^r \\ &= w \end{aligned} \tag{4}$$

In Eq. (4), w and w' are the encapsulated information for encryption and decryption calculations. C_1 is ciphertext. The improved SM9 algorithm without certificate encryption has an encapsulation mechanism that can meet the IND-CCA security definition and avoid internal and external attacks in terms of security. In addition, when internal and external adversaries have attack advantages, the algorithm can use the advantages of the adversary to solve difficult problems, indicating that the improved SM9 with certificate free encryption has extremely high security. The data integrity and source verification in IoT data transmission are improved through certificate free signature algorithms, which maintain the computational speed of the SM9 encryption algorithm while ensuring the security of DE to better build lightweight IoT. The calculation of the private key of the certificate free signature algorithm is Eq. (5).

$$sk_{SM9} = s(Hash(ID \| hid)+s)^{-1}P_1 \tag{5}$$

In Eq. (5), PKG uses generator P_1 to calculate the private key. The formula for decrypting sk_{SM9} by the user is Eq. (6).

$$sk_{SM9-CLS} = x_{ID}^{-1}s(Hash(ID \| PK \| hid)+s)^{-1}P_1 \tag{6}$$

In Eq. (6), the user uses the generator P_1 to calculate the decryption private key. The signer signs by transmitting information and the decryption private key $sk_{SM9-CLE}$ calculated by the user. The verifier verifies the signature using the public key and user identity calculated by PKG, and the calculation expression of the public key is Eq. (7).

$$PK = (x_{CD}P_2, x_{ID}P_{pub-s}) \tag{7}$$

The most important thing in signature verification is to verify whether the bilinear pairing results are consistent. If they are the same, it means the signature verification is successful. The expression to determine whether the bilinear pairing results are consistent is Eq. (8).

$$\begin{aligned} w &= u*t \\ &= e(S,P)*g^h \\ &= e(P_1,P_2)^{\frac{l*x*s(h1+s)}{(h1+s)x}} * e(P_1,P_2)^h \\ &= e(P_1,P_2)^{l*s+h*s} \\ &= e(P_1,P_2)^{(l+h)s} \\ &= e(P_1,P_2)^{rs} \\ &= g^r \end{aligned} \tag{8}$$

In Eq. (8), the digital signatures are (h,S), $u=e(S,P)$, $t=g^h$, and $g=e(P_1,P_{pub-s})$. In terms of security, the improved SM9 without certificate signature adopts a step-by-step protocol to complete the verification. The flowchart for proving distribution conventions is shown in Fig. 2.

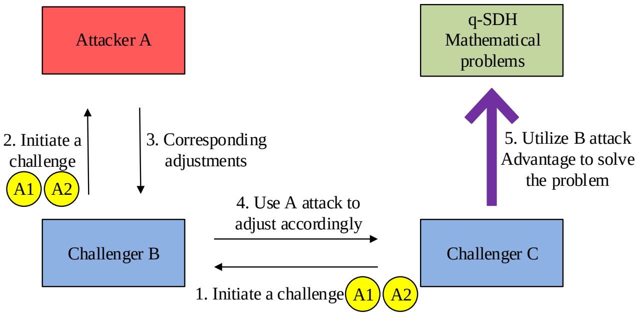

Attacker A
q-SDH Mathematical problems
2. Initiate a challenge
A1
A2
3. Corresponding adjustments
5. Utilize B attack Advantage to solve the problem
4. Use A attack to adjust accordingly
Challenger B
Challenger C
1. Initiate a challenge
A1
A2

Fig. 2. The process of proving distribution conventions.

In Fig. 2, the distribution specification sets an attacker A, as well as challenger B of A and challenger C of B. C launches an attack on B, B launches an attack on A, and A responds to B's attack. B utilizes A's response to challenge C, while C utilizes B's attack advantage to solve the q-SDH problem. The improved SM9 algorithm reduces the proof of unforgeability to solving the q-SDH problem to demonstrate the strong security of the algorithm. During the attack, the hash function constructed attack models for two attackers A1 and A2. Due to the forging signatures that can allow attackers to solve q-SDH mathematical problems, which are unsolvable, A1 cannot complete the attack. Similarly, forging signatures can solve the BDHI problem, which is also unsolvable, and A2's attack is also not feasible. Therefore, the improved SM9 without certificate signature is secure and reliable. The formula for forging signatures for A1 and A2 is Eq. (9).

$$\begin{cases} h = H_2(M \| w)S = (r-h)\dfrac{s}{x(h_* + s)}P_1 \\ h' = H_2'(M \| w)S' = (r-h')\dfrac{s}{x(h_* + s)}P_1 \end{cases} \tag{9}$$

In Eq. (9), x is a random unknown number. Challenger C in A2 calculates the solution to the BDHI problem as $e(P_1,P_2)^{1/x}$. The overall structure diagram of the improved certificate free encryption and signature SM9 algorithm is shown in Fig. 3.

In Fig. 3, the improved SM9 encryption algorithm key system structure includes five layers: business processing layer, password service layer, user key layer, PKM key layer, and data layer. This study improves the yellow section, including encryption and signature algorithms, key generation methods, and PKM calculation methods.

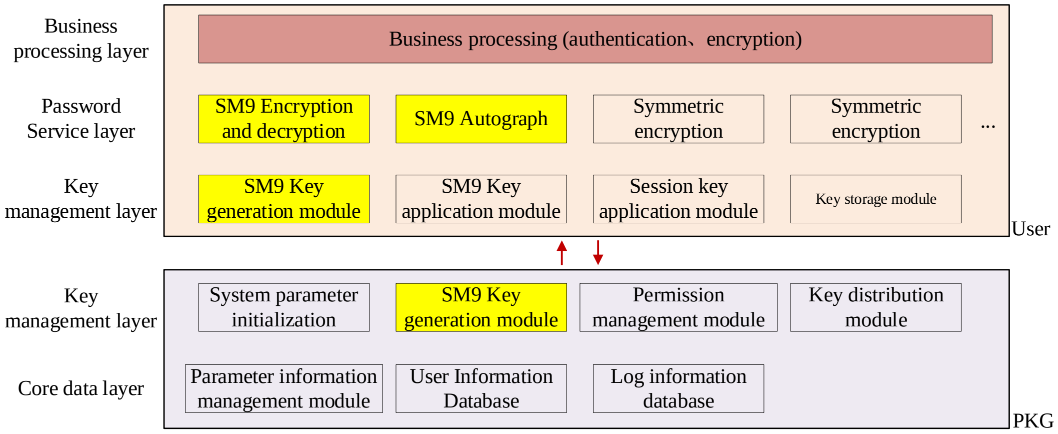

Business processing layer
Business processing (authentication、encryption)
Password Service layer
SM9 Encryption and decryption
SM9 Autograph
Symmetric encryption
Symmetric encryption
...
Key management layer
SM9 Key generation module
SM9 Key application module
Session key application module
Key storage module
User
Key management layer
System parameter initialization
SM9 Key generation module
Permission management module
Key distribution module
Core data layer
Parameter information management module
User Information Database
Log information database
PKG

Fig. 3. The overall structure of IoT SM9 key management system.

B. Construction of IoT-NS System Based on Improved SM9-iTLS

DE focuses on data protection, even if attackers obtain data, they cannot interpret the content of the data. The SP in a trusted network focuses on the rules of communication between devices, and communication can only be established based on these rules. This study integrates DE and SP to jointly build a more secure IoT communication network. IoT devices typically have low protection capabilities and are prone to interception and eavesdropping. This study introduces a lightweight TLS protocol to enhance network security [20]. The standard TLS protocol necessitates the authentication of certificates. In the case of a considerable number of IoT devices with limited computing capabilities, it requires a greater bandwidth, memory, and computational resources, which consequently increases network latency. Therefore, this study designs an improved iTLS protocol to ensure communication efficiency and further increase network security. The device establishes ecure communication throsugh a handshake protocol, and the handshake protocol of iTLS is shown in Fig. 4.

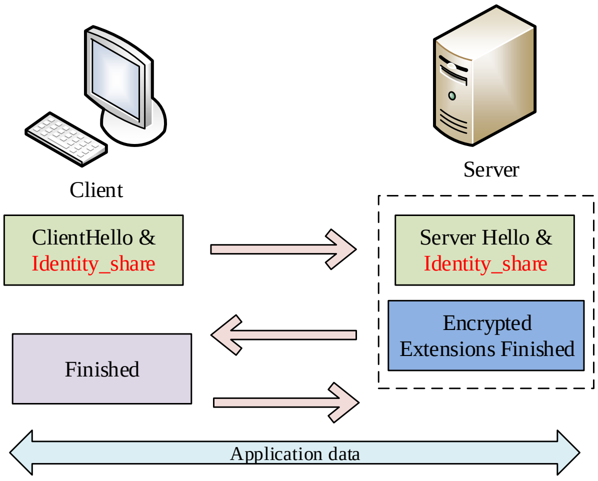

Client
Server
ClientHello & Identity_share
Server Hello & Identity_share
Encrypted Extensions Finished
Finished
Application data

Fig. 4. Completed handshake protocol process of iTLS.

In Fig. 4, an Identity share is added between the client and server to perform key exchange, and a shared key is established through this exchange mechanism. The server extends through Identity share and can provide multiple key exchange parameters, from which the server selects a key for negotiation. This approach increases the flexibility and security of the protocol. In addition, the entire handshake process of iTLS does not require certificate authentication, reducing a significant amount of authentication time. The calculation of the shared key by the server is Eq. (10).

$$shared\_secret = yEK_C \,||\, e(EK_C + H(ID_C), yP_{pub} + sk_S) \quad (10)$$

In Eq. (10), ID_C represents the identity of the client. EK_C is the temporary public key of the client. P is the generator. H is a hash function. sk_S is the private key generated by KGC on the server. y is the temporary private key of the server. The handshake key in the shared key is obtained using a key derivation function, and the formula for the key derivation function is Eq. (11).

$$handshake\_secret = \text{HKDF-Extract}(ek, shared\_secret) \quad (11)$$

In Eq. (11), HKDF is the key derived function. The client uses the server ID_S and key provided by ServerHello to calculate the shared key, and the expression for the shared key is Eq. (12).

$$shared\_secret = xEK_S \,||\, e(xP_{pub} + sk_C, EK_S + H(ID_S)) \quad (12)$$

In Eq. (12), EK_S is the temporary public key of the server, and sk_C is the private key generated by KGC for the client. This study aims to enhance protocol compatibility and reduce latency. A compatibility design is adopted to enable iTLS to establish a connection with TLS1.3. The 0-RTT mode of TLS1.3 is introduced, and dynamic keys (IDEK) are used in the 0-RTT mode to improve security. The expression for calculating IDEK on the client side is Eq. (13).

$$IDEK_C = \text{HKDF-Extract}(0, e(sk_C, H(ID_S))) \quad (13)$$

In Eq. (13), the client uses server identities ID_S and sk_C to calculate $IDEK_C$. The formula for calculating IDEK on the server is Eq. (14).

$$IDEK_S = \text{HKDF-Extract}(0, e(H(ID_C), sk_S)) \quad (14)$$

Both parties perform a 0-RTT handshake using the generated dynamic key to complete data decryption. The compatibility design of ITLS's 0-RTT handshake is shown in Fig. 5.

In Fig. 5, the protocol enhances compatibility with TLS1.3 by adding key_share extension and handshake negotiation with TLS1.3. In the 0-RTT mode, the client has added the early_data extension to prove that it will carry an Application data *. The protocol calculates dynamic passwords to protect the security of 0-RTT data, and the server determines whether to perform a handshake based on the properties of bilinear pairs. The bidirectional pair expression is Eq. (15).

$$e(sk_C, H(ID_S)) = e(H(ID_C), H(ID_S))^s = e(H(ID_C), sk_S) \quad (15)$$

The server and client complete handshake negotiation through the early_data extension of Eq. (15), achieving secure connection of IoT devices. This study improves the SM9 algorithm and TLS protocol, integrating the encryption algorithm and SP to construct an IoT-NS system based on the improved SM9-iTLS, as shown in Fig. 6.

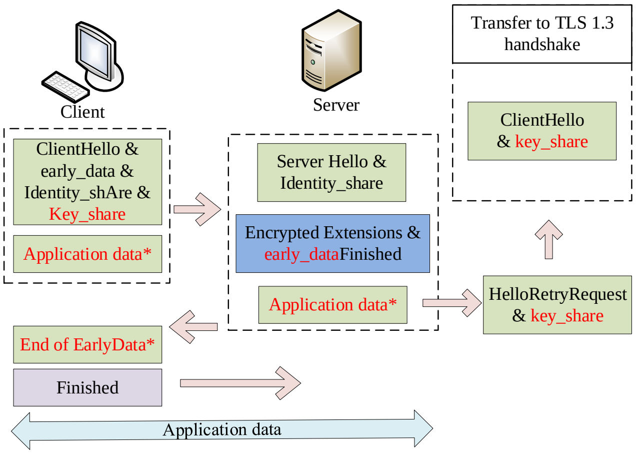

Client
Server
Transfer to TLS 1.3 handshake
ClientHello & early_data & Identity_shAre & Key_share
Application data*
Server Hello & Identity_share
Encrypted Extensions & early_dataFinished
Application data*
ClientHello & key_share
HelloRetryRequest & key_share
End of EarlyData*
Finished
Application data

Fig. 5. Compatibility design of ITLS's 0-RTT handshake mode.

In Fig. 6, the IoT device is connected to the device service enterprise and transmits communication data. The improved SM9 key system distributes keys to IoT devices and device service enterprises. During the communication process, IoT devices use SM9 keys to negotiate symmetric keys and then encrypt data based on the symmetric keys to decrypt information for device service enterprises. Using the iTLS protocol for device authentication in communication further enhances the security of information transmission. The SM9 key system includes a registration center and a key generation center PKG, mainly responsible for user registration and SM9 key generation and distribution. The IoT network system utilizes SM9 encryption technology and iTLS protocol to achieve identity authentication and encrypted data exchange.

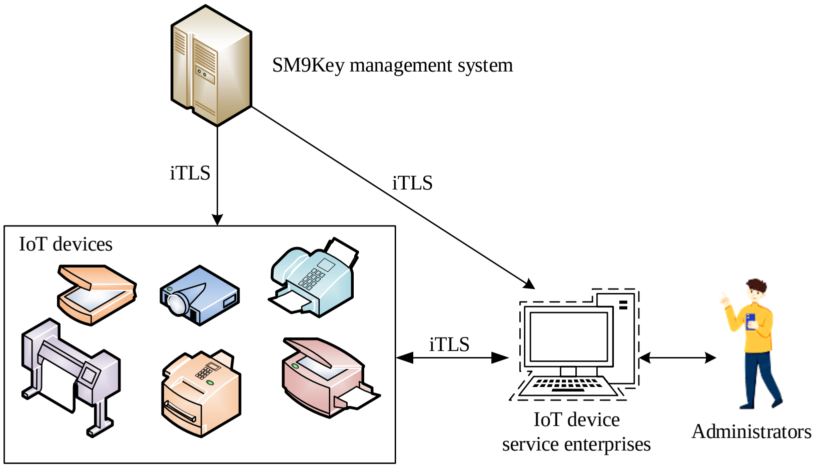


Fig. 6. IoT security management system based on improved SM9 encryption algorithm and iTLS SP.

III. Results

To verify the performance of the system, relevant experiments were conducted in this study. The experiment first conducted a comparative experiment on the improved SM9 encryption algorithm to test its computational efficiency. Then, comparative experiments were conducted on iTLS to test the network latency under iTLS SP. Finally, experimental analysis of the IoT-NS system based on improved SM9-iTLS showed that the proposed IoT-NS system not only enhanced security, but also could respond quickly under the improved encryption algorithm and SP, without causing IoT communication burden.

A. Experimental Environment and Parameter Settings

This study used the MIRACL library to test the improved SM9 encryption algorithm. The test code was written on ESP32, and the size of the encrypted information was selected as 120 bits. The elliptic curve of SM9 selected 256 bits to construct an addition group, with two addition groups of 64 and 128 bytes in length. 120 repeated tests were conducted to calculate the average running time of the algorithm. This study implemented SP iTLS experimental analysis in the WolfSSL library written in C language, with security levels of 112 and 128 bit. All experiments selected TLS_AES_128_CCM_SHA256 as the hash function and symmetric encryption cipher suite. The computer used in the experiment had 16GB of memory and an Intel i7 8700 processor. Table I shows the detailed parameter settings for each encryption algorithm in ESP32.

The experiment selected RSA3072, SM9, and improved SM9 for comparative testing, and TLS1.3 and iTLS with RSA, ECC, and 0-RTT authentication modes were used for comparative testing. A comparative experiment was conducted between the IoT-NS system based on ECC-iTLS, RSA-TLS, and ElGamal TLS and SM9 iTLS. The evaluation indicators used encryption, decryption, signature, and signature verification time as the computational efficiency evaluation indicators of encryption algorithms. Communication connection delay was used as an indicator of SP connection efficiency. Traffic overhead was utilized as an evaluation indicator for network resource occupancy.

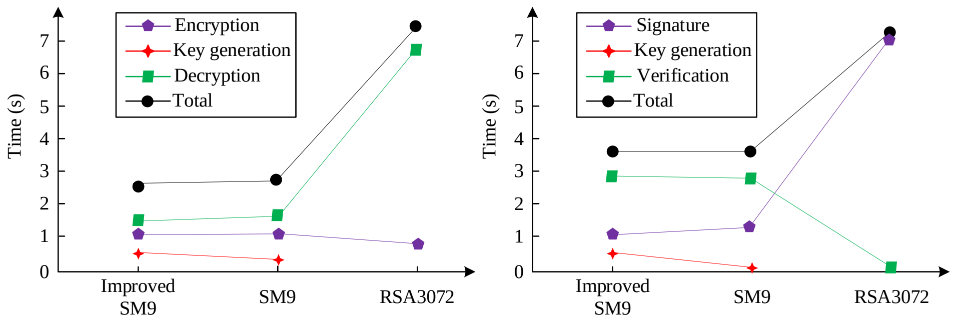
TABLE I. DETAILED INFORMATION ON PARAMETER SETTINGS FOR ALGORITHMS

/	Key escrow	Certificate	(Encryption algorithm) Public key	(Signature algorithm) Public key	(Encryption algorithm) Private key	(Signature algorithm) Private key
Improved SM9	No	No	ID+128	ID+256	128	64
RSA3072	No	Yes	384	384	384	384
SM9	Yes	No	ID	ID	128	64

B. Analysis of Improved SM9 Computing Efficiency and iTLS Communication Delay

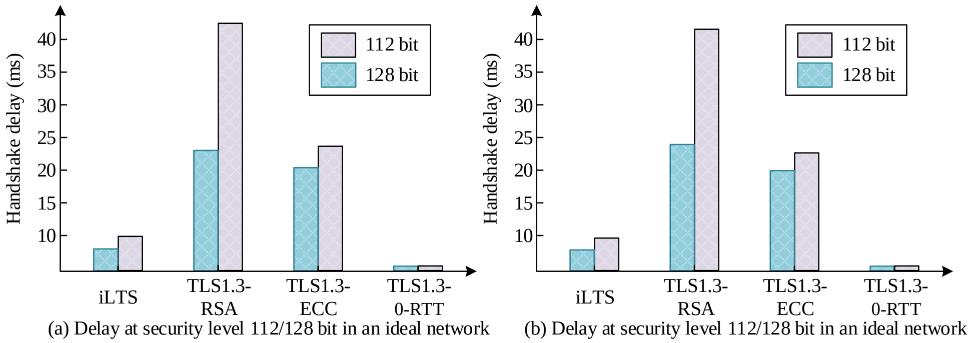
The experiment first compares the improved SM9 algorithm. The calculation time results for the complete encryption solution, encryption process, and complete signature and verification process are shown in Fig. 7.

In Fig. 7 (a), the key generation time of the research algorithm is 0.39s, the encryption time is 1.02s, the decryption time is 1.30s, and the total time is 2.71s. The encryption time of SM9 is lower than that of the improved SM9, but the decryption time and key time are longer, with a total time of 2.74s, which is higher than SM9. The key generation time of RSA3072 is lower than that of the research algorithm, but the decryption time is 6.78s, and the total calculation time is 7.31s. In Fig. 7 (b), the total time for key generation, signature, and signature verification of the research algorithm is 3.74s, and the total time for SM9 algorithm is 3.76s. The signature verification time of RSA3072 is the lowest, with a minimum value of 0.06s, but the key generation time takes 6.88s, resulting in a total time of 6.94s. The experiment sets up an ideal network with zero delay and no packet loss under wireless broadband, and tests the handshake delay of iTLS protocol and TLS1.3 protocol at security levels of 112-bit and 128-bit, as shown in Fig. 8.



(a) Comparison of encryption algorithms
(b) Comparison of signature algorithms

Fig. 7. Comparison of encryption and signature verification times of algorithms in ESP32.



(a) Delay at security level 112/128 bit in an ideal network
(b) Delay at security level 112/128 bit in an ideal network

Fig. 8. Full handshake delay for iTLS and TLS 1.3 under ideal network conditions.

In Fig. 8 (a), when the security levels are 112-bit and 128-bit, the TLS1.3 protocol delay in the 0-RTT mode is the lowest, with the lowest values of 1.4ms and 1.5ms. The delay of iTLS is 6.4ms and 9.9ms respectively, the delay of TLS1.3-RSA is 23.3ms and 43.9ms, and the delay of TLS1.3-ECC is 21.5ms and 23.7ms. Fig. 8 (b) shows the results of the second group of experiments. The delay of iLTS is 6.7ms at 112-bit security level and 9.7ms at 128-bit security level. The delay of iLTS is slightly increased compared to TLS1.3-0-RTT protocol, which is also improved by TLS, but it is still at a low latency level.

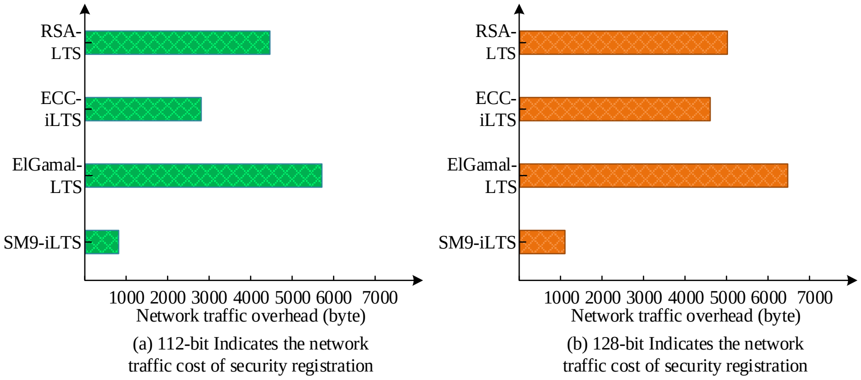
C. Performance Analysis of IoT-NS System Based on Improved SM9-iTLS

This study compares the improved IoT-NS system of SM9-iTLS with the aforementioned IoT security system. The network traffic overhead of each system under 112-bit and 128-bit security levels is shown in Fig. 9.

In Fig. 9 (a), when the security level is 112-bit, the network traffic overhead of the research system is the lowest, with a minimum value of 794 bytes, ElGamal-TLS of 5731 bytes, ECC-iTLS of 2788 bytes, and RSA-TLS of 4312 bytes. In Fig. 9 (b), when the security level is 128-bit, the network traffic overhead of the research system is the lowest, with a minimum value of 1127 bytes, ElGamal-TLS of 6395 bytes, ECC-iTLS of 4623 bytes, and RSA-TLS of 5018 bytes. In summary, the proposed IoT-NS system has the lowest network traffic overhead in tests at different security levels. The experimental results of the complete encryption and decryption time and signature verification time of each system are shown in Fig. 10.

In Fig. 10 (a), the total time for key generation, encryption, and decryption of SM9-iTLS is 3.63 seconds. The total time for ElGamal-TLS is 9.27, ECC-iTLS is 5.33 seconds, and RSA-TLS is 7.31 seconds. In Fig. 10 (b), the total time for key generation, signature, and verification of SM9-iTLS is 3.65 seconds. The total time of ElGamal-TLS is 10.97 seconds, ECC-iTLS is 9.27 seconds, and RSA-TLS is 11.37 seconds. Therefore, the IoT-NS system based on improved SM9-iTLS can ensure good efficiency in information encryption, decryption, and identity authentication during secure communication. To further test and study the performance of the system, comparative experiments are conducted on system communication delay under different network delays, network bandwidth, and packet loss rates, as shown in Fig. 11.

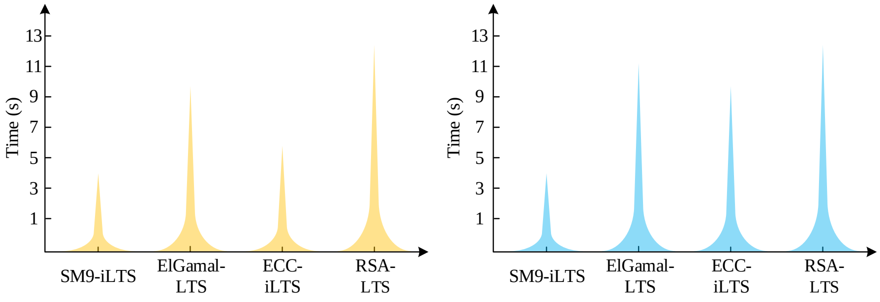
Fig. 11 (a) shows the experimental results under a network delay of 1-256ms. Under different network delays, SM9-iTLS has the lowest communication delay, while ElGamal-TLS has the highest communication delay. Fig. 11 (b) shows the experimental results under different network packet loss rates. When the network packet loss rate is below 15%, the communication delay of each system remains around 0, but as the packet loss rate increases, the communication delay continues to increase, especially ECC-iTLS, which shows exponential growth. When the packet loss rate is 25%, the communication delay of SM9 iTLS is 317ms, RSA-TLS is 5879ms, ElGamal-TLS is 1752ms, and ECC-iTLS is 11674ms. In Fig. 11 (c), under different network bandwidths, the communication delay of SM9-iTLS is the lowest, while ECC-iTLS has the highest communication delay. In summary, under different network quality communication environments, the SM9-iTLS IoT-NS system has the fastest response speed and the least burden on communication.



(a) 112-bit Indicates the network traffic cost of security registration

(b) 128-bit Indicates the network traffic cost of security registration

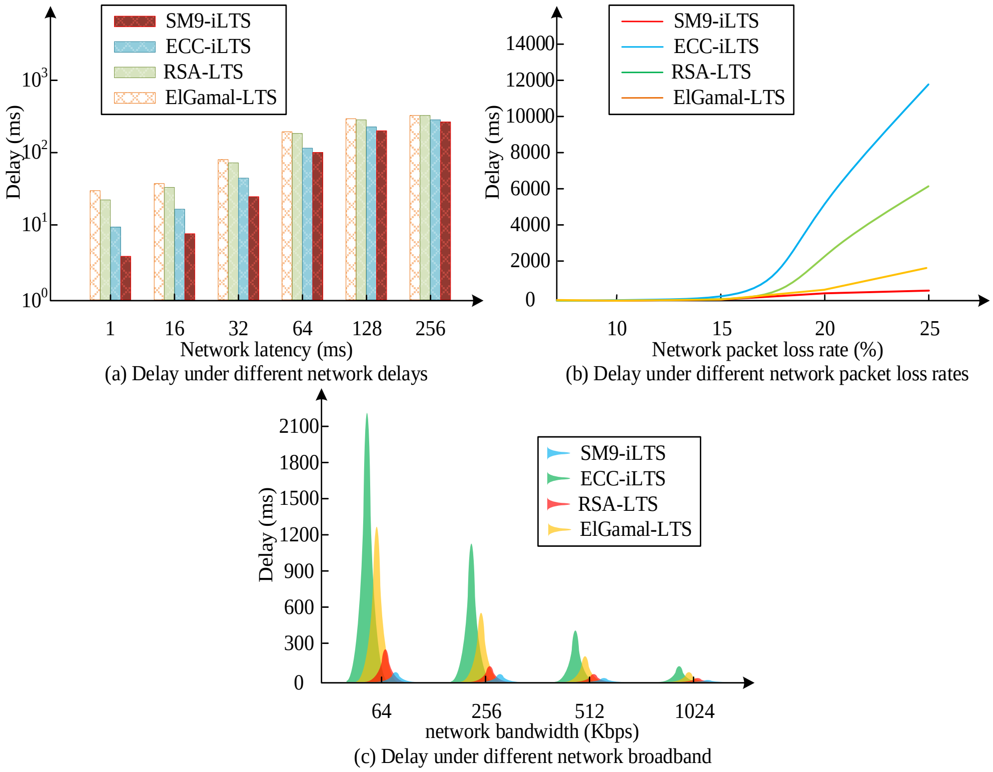
Fig. 9. Network traffic overhead for secure registration of 112 bit and 128 bit.



(a) Total time to complete encryption and decryption

(b) Total time to complete signature and verification

Fig. 10. The total travel time results of various model comparison experiments.



(a) Delay under different network delays

(b) Delay under different network packet loss rates

(c) Delay under different network broadband

Fig. 11. Communication latency of various systems under different latency, bandwidth, and packet loss rates.

IV. Discussion

This study compared and analyzed the computational efficiency and communication delay of the improved SM9 encryption algorithm and iTLS SP, and conducted comparative tests on the performance of the IoT-NS system based on the improved SM9-iTLS. The results of this study showed that the SM9 encryption algorithm took a total of 2.71s for key generation, encryption, and decryption, while the total time for key generation, signature, and verification was 3.74s, which is more efficient than other encryption algorithms. This result was similar to the research conclusion of Jing's team on improving the addition and multiplication sets of the SM9 algorithm [21]. The improved SM9 encryption algorithm not only had higher security but also improved computational efficiency. In the comparative experiment of iTLS, at the security levels of 112 and 128 bits, the latency of iTLS was 6.4ms and 9.9ms, slightly higher than TLS1.3-0-RTT, but at a low latency level. This result was similar to the conclusion of Zhang's team in designing SP based on La-TLS network [22]. Therefore, the improved SP could further enhance network security without increasing communication burden. Finally, in the performance analysis of the IoT-NS system based on improved SM9-iTLS, the network traffic overhead of SM9-iTLS was 794 and 1127 bytes at security levels of 112 and 128 bits, respectively. The total time for key generation, encryption, and decryption was 3.63s. The total time for generating, signing, and verifying dynamic keys was 3.65s. When the packet loss rate was 25%, the communication delay of SM9-iTLS was 317ms. Under different conditions, the network traffic overhead, encryption and decryption time, and latency of the research system were superior to other systems. This was similar to the conclusion obtained by Wang's team in the lightweight communication network based on SM9 and TLS improvements designed in 2024 [23]. This result indicated that the IoT-NS system based on improved SM9-iTLS not only enhanced the security capability of IoT device communication but also enhanced the efficiency of network communication.

V. Conclusion

The development of digitalization has brought about an increasing number of IoT devices, but these devices often have weaker self-protection capabilities. This study established a lightweight IoT-NS system by improving the SM9 encryption algorithm and combining iTLS SP with IoT devices. The experimental data demonstrated that the IoT-NS system, based on an enhanced SM9-iTLS, exhibited notable optimization in network resource utilization, computational efficiency, and communication delay, while simultaneously enhancing network security. The research system exhibited superior performance compared to other IoT-NS systems. Therefore, in situations where IoT device resources were limited, using the SM9 encryption algorithm to generate and distribute keys using PKG and the iTLS communication protocol to use dynamic keys for identity authentication, could improve the security of IoT networks. In practical environments, there are significant differences in the configuration of IoT devices. Some devices have strong computing power, but most devices have low computing power. In special environments, the time cost may be high and cannot meet the communication needs. Therefore, improving the hardware environment is the direction of later research. For example, despite the SM9 algorithm's robust security, the public key appends a sequence of inconsequential digits to the initial one, necessitating additional storage space and communication overhead. This is a problem that exists in certificate free systems, and further research can be conducted to improve it. The KGC of a single trust domain network is set by the network manager. However, in multi-trust domain networks, a single KGC greatly limits the scalability of iTLS. Future research will focus on studying cross domain authentication schemes based on identity passwords.

Acknowledgment

The research is supported by The 2023 Research Fund Project of Guangzhou Open University, titled "Research on the Model of Building a Digital University to Support the High Quality Development of Guangzhou Open University", with reference number: 2023KYZXYB09.

References

[1] Qian J, Li H, Huo Y, Xing X. Empowering IoT security: an innovative handover-driven node selection approach to tackle conscious mobile eavesdropping. International Journal of Sensor Networks (2), 2024, 44(2):84-98. circuit for iot security. IET Circuits, Devices & Systems, 2022, 16(1):40-52.

[2] Ipseeta Nanda, Rajesh De. THE STATE OF THE ART IN ECO-FRIENDLY IOT. Information Management and Computer Science, 2022, 5(1):18-22.

[3] Ahmad Muhammad Thantawi, Sri Astuti Indriyati. Conceptual Design Impacts in New Normal Era: The Use of Artificial Intelligence (AI) And Internet Of Things (IOT) (Case Studies: Class Room And Restaurant). Acta Informatica Malaysia. 2022; 6(2): 39-42.

[4] Maseno E M, Wang Z, Liu F. Intrusion Detection System in IoT Based on GA-ELM Hybrid Method. Journal of Advances in Information Technology, 2023, 14(4):625-629.

[5] Marzouk R, Alrowais F, Negm N, Alkhonaini M A, Hamza M A, Rizwanullah M, Yaseen I, Motwakel A. Hybrid deep learning enabled intrusion detection in clustered iiot environment. Computers, Materials & Continua, 2022, 1(8):3763-3775.

[6] Kiruba D G, Benita J. A Survey of Secured Cluster Head: SCH based Routing Scheme for IOT based Mobile Wireless Sensor Network.ECS transactions, 2022, 107(1):16725-16745.

[7] Yadav K, Jain A, Alharbi Y, Alferaidi A, Alkwai L M, Ahmed N M O S, Hamad S A S. A secure data transmission and efficient data balancing approach for 5g-based iot data using uudis-ecc and lsrhs-cnn algorithms.

[8] Senthilkumar M, Murugan BS. Enhancing The Security of An Organization From Shadow Iot Devices Using Blow-Fish Encryption Standard. Acta Informatica Malaysia. 2022; 6(1): 22-24.

[9] Rani D, Tripathi S. Design of blockchain-based authentication and key agreement protocol for health data sharing in cooperative hospital network. Journal of supercomputing, 2024, 80(2):2681-2717.

[10] Lapworth L. Parallel encryption of input and output data for HPC applications. International Journal of High Performance Computing Applications, 2022, 36(2):231-250.

[11] Zhang Q, Zhao Z. Distributed storage scheme for encryption speech data based on blockchain and IPFS. Journal of supercomputing, 2023, 79(1):897-923.

[12] He D, Cai Y, Zhu S, Zhao Z, Chan S, Guizani M. A lightweight authentication and key exchange protocol with anonymity for iot. IEEE transactions on wireless communications, 2023, 22(11):7862-7872.

[13] Mvah F, Tchendji V K, Djamegni C T, Anwar A H, Tosh D K, Kamhoua C. Gatebasep: game theory-based security protocol against arp spoofing attacks in software-defined networks. International Journal of Information Security(1), 2024, 23(1):373-387.

[14] Rathee G, Kerrache C A, Calafate C T. An Ambient Intelligence approach to provide secure and trusted Pub/Sub messaging systems in IoT environments. Computer networks, 2022, 218(9):1-9.

[15] Chowdhury R R, Idris A C, Abas P E. Identifying SH-IoT devices from network traffic characteristics using random forest classifier. Wireless networks, 2024, 30(1):405-419.

[16] Jinghua Z. Special Issue on Machine Learning and Big Data Analytics for IoT Security and Privacy (SPIoT2022). Neural computing & applications, 2024, 36(5):2119-2120.

[17] Mohanrasu S S, Udhayakumar K, Priyanka T M C, Gowrisankar A, Banerjee S, Rakkiyappan, R. Event-triggered impulsive controller design for synchronization of delayed chaotic neural networks and its fractal reconstruction: an application to image encryption. Applied mathematical modelling, 2023, 115(1):490-512.

[18] Wu S T. An Application of Keystream Using Cellular Automata for Image Encryption in IoT. Journal of Internet Technology, 2023, 24(1):149-162.

[19] Zhang Y, Wu Q, Wang P, Wen L, Luan Z, Gu C. Tvd-pb logic circuit based on camouflaging

[20] Yadav A K, Misra M, Pandey P K, Braeken A, Liyange M. An improved and provably secure symmetric-key based 5g-aka protocol. Computer networks, 2022, 218(9):1-13.

[21] Jing S, Yang X, Feng Y, Liu X, Hao F, Yang Z. Hardware Implementation of SM9 Fast Algorithm Based on FPGA. Atlantis Press, 2022, 12(27):797-803.

[22] Xinglong Z, Qingfeng C, Yuting L I. LaTLS:A Lattice-Based TLS Proxy Protocol. Chinese Journal of Electronics, 2022, 31(2)313-321.

[23] Wang D, Dong L, Tang H, Gu J, Liu Z, You X. SDN Security Channel Constructed Using SM9. International Symposium on Digital Forensics and Security, 2024, 4(12):1-5.