PLOS One

RESEARCH ARTICLE

Enhancing security in instant messaging systems with a hybrid SM2, SM3, and SM4 encryption framework

He-Jun Lu[1,2]*, **Roben A. Juanatas**[1], **Mideth B. Abisado**[1]

1 College of Computing and Information Technologies, National University, Manila, Philippines,
2 The School of Big Data and Artificial Intelligence, Anhui Xinhua University, Hefei, Anhui, China

* luhejun@axhu.edu.cn

Abstract

With the rapid integration of instant messaging systems (IMS) into critical domains such as finance, public services, and enterprise operations, ensuring the confidentiality, integrity, and availability of communication data has become a pressing concern. Existing IMS security solutions commonly employ traditional public-key cryptography, centralized authentication servers, or single-layer encryption, each of which is susceptible to single-point failures and provides only limited resistance against sophisticated attacks. This study addresses the research gap regarding the complementary advantages of SM2, SM3, and SM4 algorithms, as well as hybrid collaborative security schemes in IMS security. This paper presents a hybrid encryption security framework that combines the SM2, SM3, and SM4 algorithms to address emerging threats in IMS. The proposed framework adopts a decentralized architecture with certificateless authentication and performs all encryption and decryption operations on the client side, eliminating reliance on centralized servers and mitigating single-point failure risks. It further enforces an encrypt-before-store policy to enhance data security at the storage layer. The framework integrates SM2 for key exchange and authentication, SM4 for message encryption, and SM3 for integrity verification, forming a multi-layer defense mechanism capable of countering Man-in-the-Middle (MITM) attacks, credential theft, database intrusions, and other vulnerabilities. Experimental evaluations demonstrate the system's strong security performance and communication efficiency: SM2 achieves up to 642 times faster key generation and 2.2 times faster decryption compared to RSA-3072; SM3 improves hashing performance by up to 11.5% over SHA-256; and SM4 delivers up to 22% higher encryption efficiency than AES-256 for small data blocks. These results verify the proposed framework's practicality and performance advantages in lightweight, real-time IMS applications.

OPEN ACCESS

Citation: Lu H-J, Juanatas RA, Abisado MB (2025) Enhancing security in instant messaging systems with a hybrid SM2, SM3, and SM4 encryption framework. PLoS One 20(9): e0332665. https://doi.org/10.1371/journal.pone.0332665

Editor: Vincent Omollo Nyangaresi, Jaramogi Oginga Odinga University of Science and Technology, KENYA

Received: April 6, 2025

Accepted: September 2, 2025

Published: September 15, 2025

Data availability statement: All relevant data are within the manuscript.

Funding: This work was supported by the Project (Grant Nos. 2020xfxm27, 2015ckjh117, 2016mooc190, 2019sysxx03).

Competing interests: The authors have declared that no competing interests exist.

Introduction

During the digital transformation of Chinese society, IMS have progressively evolved from their initial social functions to encompass public service domains such as financial transactions, government services, and enterprise organizational management. Furthermore, they have catalyzed the emergence of new market models like community group buying. As a vital component of digital infrastructure for social governance, an essential driving force behind economic transformation, and a key facilitator of public welfare services, IMS has demonstrated irreplaceable value in contemporary society.

With the rapid surge in China's internet user base and the widespread adoption of IMS, information security concerns have become increasingly severe. According to the Statistical Report on Internet Development in China, by December 2024, the number of internet users in China had reached 1.108 billion, with IMS users accounting for 1.081 billion, representing 97.6% of the total internet population [1]. During IMS usage, users frequently encounter security threats, including stolen account credentials [2,3], leaked personal information [4], and intercepted or compromised chat records [5,6], highlighting the inadequacy of current security mechanisms in mitigating emerging cyber threats [7]. With the increasing complexity of IMS functionalities and the diversification of their applications, traditional encryption techniques struggle to meet emerging security demands, particularly in defending against MITM attacks, data tampering, and unauthorized access. Most IMS security solutions available on the market today primarily rely on centralized encryption mechanisms, which can make servers prime targets for cyberattacks and pose a risk of single points of failure [8,9]. In contrast, decentralized security solutions reduce dependence on a central server, thereby mitigating potential security risks. Therefore, the development of an efficient and secure encryption framework to enhance IMS data protection has emerged as a critical and urgent challenge.

At present, there is an important gap in the research of security schemes for IMS: No complete solution that can give full play to the collaborative advantages of SM2, SM3, and SM4 algorithms has emerged yet. The current literature lacks an in-depth exploration of end-to-end IMS security architectures. Consequently, this article proposes an efficient and secure hybrid encryption security framework for IMS, which integrates decentralization, certificateless authentication, and end-to-end encrypted transmission based on the complementary strengths of the SM2, SM3, and SM4 algorithms. This framework will be capable of effectively mitigating multiple security threats, such as data breaches, MITM attacks, and database intrusions, thereby significantly enhancing the security and reliability of IMS.

The main contributions of this study are summarized as follows:

- **Decentralized security framework with certificateless authentication:** We propose a novel security framework based on a decentralized architecture, which fully delegates encryption and decryption operations to client devices. This approach eliminates reliance on a central server and reduces the risk of single-point failures inherent in traditional architectures.

- **End-to-end hybrid encryption:** The framework implements a hybrid encryption mechanism throughout the transmission chain, ensuring data confidentiality, integrity, and availability during communication.
- **Encrypt-before-store principle:** All data are encrypted prior to storage in the database, thereby enhancing security at rest.
- **Hybrid encryption scheme for IMS:** To meet the real-time and low-latency requirements of Instant Messaging Systems (IMS), we implement a hybrid encryption and decryption scheme combining the SM2, SM3, and SM4 algorithms.
- **Performance evaluation against standard algorithms:** We evaluate the performance of SM2, SM3, and SM4 against conventional algorithms (RSA-3072, SHA-256, and AES-256) for small data blocks (less than or equal to 128 KB). Experiments were conducted using the Bouncy Castle cryptography library in a standardized environment with hardware acceleration for AES-NI and SHA-NI disabled.

The remainder of this paper is organized as follows: The Related work section systematically reviews the research progress on the application of SM2, SM3, and SM4 algorithms in the field of information security, and at the same time conducts a literature review on the relevant research achievements of IMS network security protection technology. The Methods section presents the proposed methodology based on SM2, SM3, and SM4 algorithms, introducing the overall security framework and the comprehensive security architecture alongside a hybrid encryption scheme for data communication and its implementation approach. The Results section details the experimental results obtained from this study. The Discussion section provides an analysis and discussion of the experimental findings. The Conclusions section concludes this paper by summarizing the key findings and contributions presented herein.

Related work

China has made significant progress in cryptographic research, with SM2 [10], SM3 [11], and SM4 [12] recognized as international standards. Li et al. [13] proposed a side-channel-resistant SM2 point multiplication, while Zhai et al. [14] developed a distributed SM2 decryption scheme for IoT. Cao et al. [15] applied SM2 to trusted metrological data, and Zhang et al. [16] introduced traceable ring signatures. Wu et al. [17], Xu et al. [18], and Zhao et al. [19] optimized SM2-based authentication and key exchange. Jayakumari et al. [20] employed ECC for multimedia protection, and Prabhu et al. [21] enhanced cloud storage security. Compared with RSA, SM2 achieves superior efficiency, with a 256-bit key providing security equivalent to a 3072-bit RSA key [22,23]. However, these studies primarily focus on isolated improvements rather than integrated frameworks ensuring confidentiality, integrity, and availability.

Research on hash algorithms highlights SM3's role as a robust alternative. Zheng et al. [24] designed a low-power SM3 implementation for IoT, and Han et al. [25] proposed a CUDA-based optimization to improve throughput. Stevens et al. [26] demonstrated chosen-prefix collisions for MD5, while Leurent and Peyrin [27] reported the first chosen-prefix collision on SHA-1, rendering legacy schemes unsuitable for secure applications. Nevertheless, prior work remains limited to algorithmic optimization or cryptanalysis, without system-level integration into low-latency secure communication frameworks.

Studies on SM4 have primarily focused on efficiency and adaptability. Guo et al. [28] extended SM4 into tweakable block ciphers, Zhang et al. [29] achieved record-breaking bit-sliced performance on x86, and Hu et al. [30] analyzed the SM algorithm family and software performance trade-offs. These contributions confirm SM4's potential but do not explore system-level integration for latency-sensitive applications.

In instant messaging (IM) security, Liu et al. [31] implemented a hybrid 3DES–RC4 scheme. Tajudeen et al. [32] reviewed AES-based techniques for message protection. Kasar et al. [33] investigated decentralized WebRTC-based messaging, and Zhou et al. [34] designed an enterprise IM system with hierarchical protection. While these studies improve security and performance, they are largely confined to single algorithms or enterprise-specific solutions, leaving a gap in decentralized, end-to-end secure frameworks.

To address this gap, this study proposes a decentralized hybrid encryption framework integrating SM2, SM3, and SM4 with certificateless authentication. The framework enforces encrypt-before-store, mitigates single-point failures, and provides confidentiality, integrity, and availability for real-time IM systems, bridging the gap between algorithmic enhancements and practical deployment.

Methods

SM2 algorithm principle

SM2 is a public key cryptosystem that relies on the computational hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Let Q and P be two points on an elliptic curve, where $Q = dP$ $(d \in Z)$, d represents the private key, while Q serves as the corresponding public key. The challenge of deriving d from Q and P is considered computationally infeasible, forming the foundation of the algorithm's security. The operational principles of the SM2 algorithm are depicted in Fig 1.

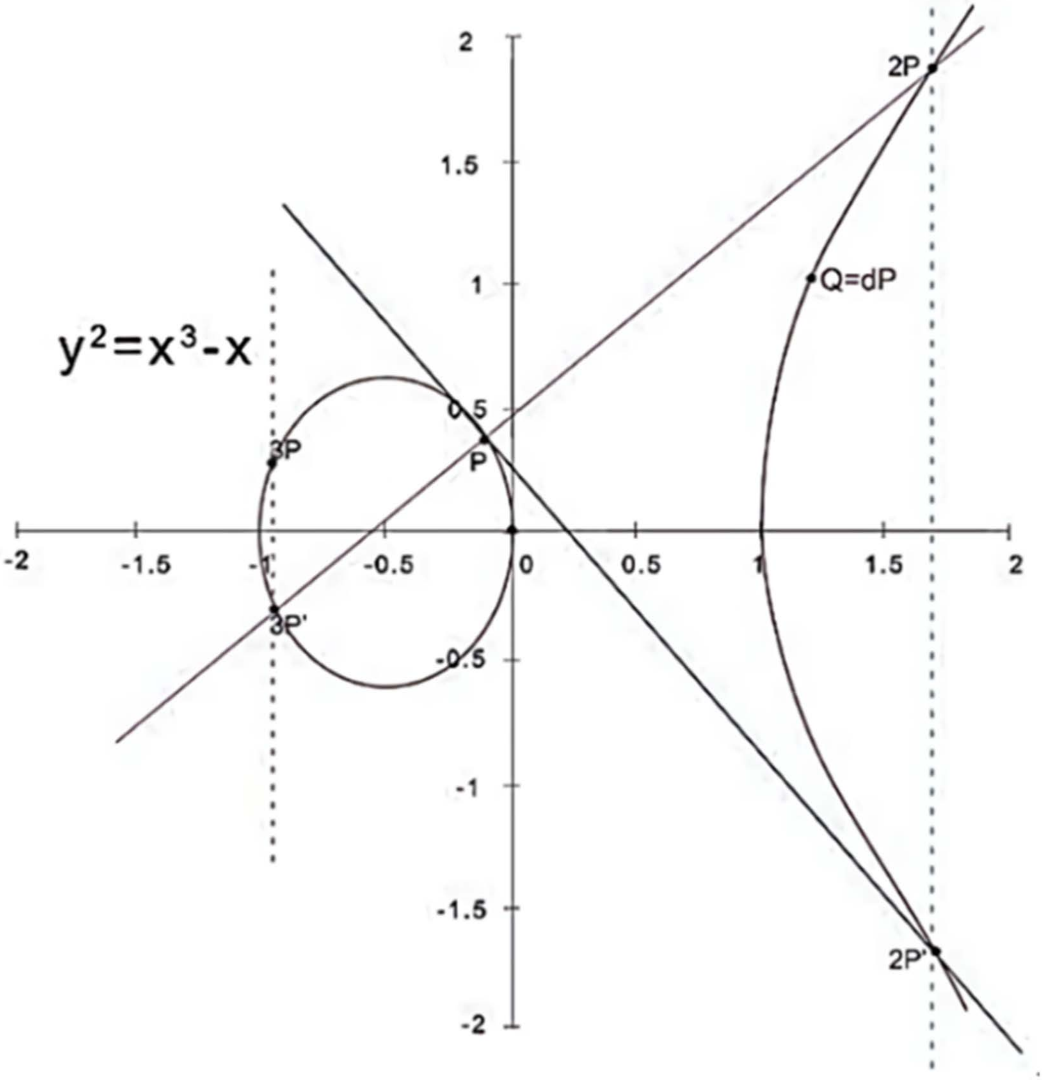

Fig 1. The principles of the SM2 algorithm.

https://doi.org/10.1371/journal.pone.0332665.g001

The encryption and decryption processes of the SM2 algorithm are defined as follows:

Consider a scenario where User A transmits a message M to User B, with L denoting the bit length of M. Let d be the private key, where $d \in [1,\ n-1]$, and d is a randomly generated 256-bit integer. The parameter n represents the order of the publicly known base point G on the elliptic curve, while P denotes the corresponding public key, as formulated in Eq (1).

$$P = d \times G \tag{1}$$

Step A1: User A receives the public key P_B from User B, where P_B denotes a point on the elliptic curve, as defined in Eq (2).

$$P_B = (x_B, y_B) \tag{2}$$

Step A2: User A selects a random integer *k*, where $k \in [1, n-1]$, and computes C_1, as defined in Eq (3).

$$C_1 = k \times G = (x_1, y_1) \tag{3}$$

Step A3: Compute the shared secret key S, as defined in Eq (4).

$$S = k \times P_B = (x_s, y_s) \tag{4}$$

Step A4: Utilize the Key Derivation Function (KDF) to compute the encryption key t, as defined in Eq (5).

$$t = KDF(x_s \,||\, y_s, L) \tag{5}$$

Step A5: Compute the ciphertext C_2, as defined in Eq (6), where $\oplus$ denotes the XOR operation.

$$C_2 = M \oplus t \tag{6}$$

Step A6: Compute the ciphertext C_3, as defined in Eq (7).

$$C_3 = Hash(x_s \,||\, M \,||\, y_s) \tag{7}$$

Step A7: Compute the ciphertext C, as expressed in Eq (8).

$$C = (C_1 \,||\, C_2 \,||\, C_3) \tag{8}$$

Upon receiving the ciphertext C, User B extracts C_1, C_2, and C_3.

Step B1: User B computes the shared secret key S′ using their private key d_B, as defined in Eq (9).

$$S^{'} = d_B \times C_1 = ({x_s}', {y_s}') \tag{9}$$

Step B2: Apply the KDF to compute the key t′, as defined in Eq (10).

$$t' = KDF({x_s}' \,||\, {y_s}', L) \tag{10}$$

Step B3: Decrypt C_2 to obtain M′, as defined in Eq (11).

$$M' = C_2 \oplus t' \tag{11}$$

Step B4: Compute C_3' as defined in Eq (12). If C_3 is equal to C_3', then the plaintext M′ is successfully output.

$$C_3' = \mathrm{Hash}(x_s' \,||\, M' \,||\, y_s') \tag{12}$$

SM3 cryptographic hash algorithm

For a message m of length L (where $L < 2^{64}$ bits), the SM3 hash algorithm processes the message in blocks and iteratively compresses it to generate a hash value of 256 bits. The algorithm flow is shown in Fig 2.

The message padding process begins by appending a ‘1’ bit, followed by k ‘0’ bits where k is the integer satisfying Eq (13) and a 64-bit L’s binary representation, thereby generating a padded message m’ with a length multiple of 512 bits as specified in Eq (14). The message is divided into n blocks (n determined by Eq 15), each denoted as B_i. During computation, each block undergoes message expansion and is processed through the compression function CF in an iterative manner. As shown in Eq (16), V_n is the result of iterative compression, where V_0 is the 256-bit initial value IV, ultimately producing a 256-bit hash value.

$$L + 1 + k \equiv 448 \bmod 512 \tag{13}$$

$$m' = (B_0, B_1, \cdots, B_{n-1}) \tag{14}$$

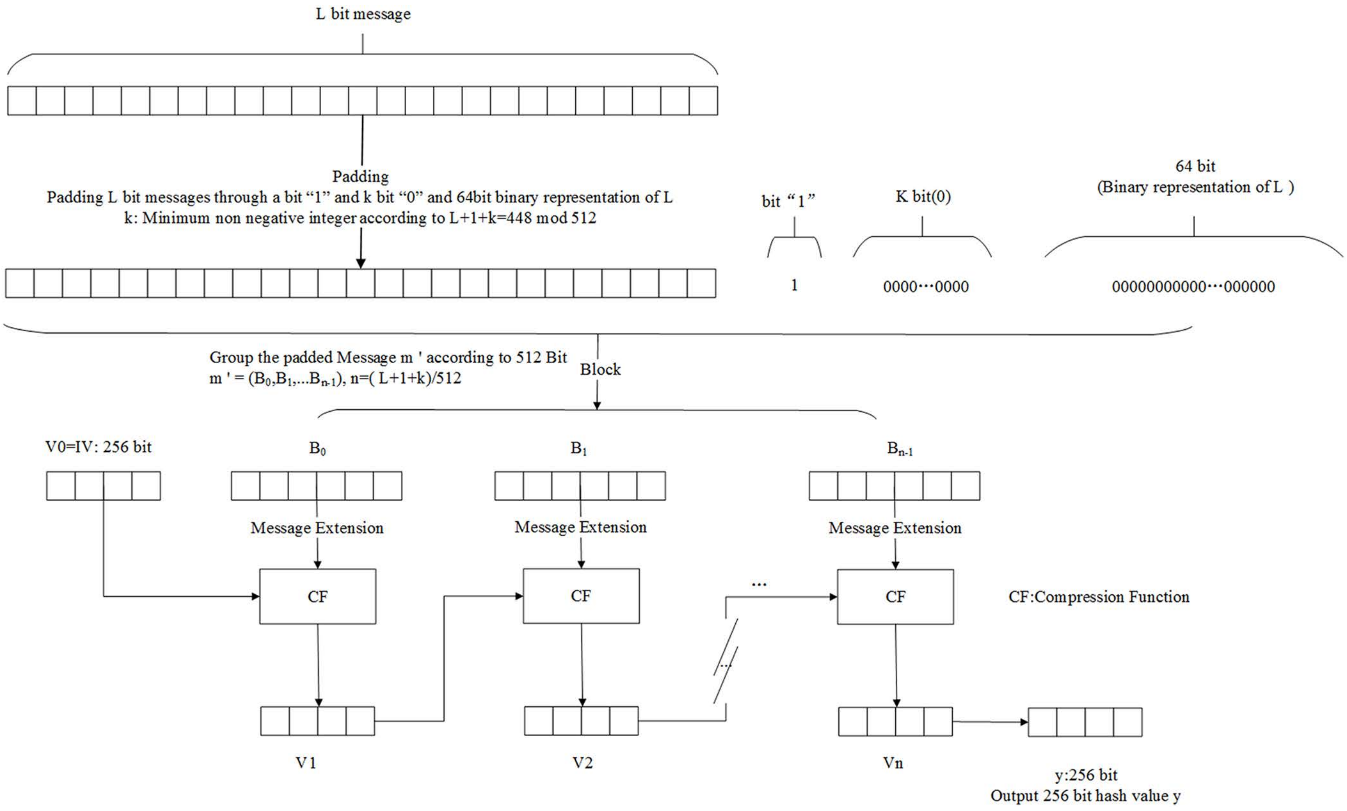
L bit message
Padding
Padding L bit messages through a bit “1” and k bit “0” and 64bit binary representation of L
k: Minimum non negative integer according to L+1+k=448 mod 512
bit “1”
K bit(0)
64 bit
(Binary representation of L)
1
0000···0000
00000000000···000000
Group the padded Message m ' according to 512 Bit
m ' = (B0,B1,...Bn-1), n=(L+1+k)/512
Block
V0=IV: 256 bit
B0
B1
Bn-1
Message Extension
Message Extension
Message Extension
CF
CF
...
CF
CF:Compression Function
...
V1
V2
Vn
y:256 bit
Output 256 bit hash value y

Fig 2. The SM3 cryptographic hash algorithm.

https://doi.org/10.1371/journal.pone.0332665.g002

$$n = (L + k + 65)/512 \tag{15}$$

$$V_{(i+1)} = CF(V_i,B_i)(i=0, 1, 2\cdots, n–1) \tag{16}$$

SM4 block cipher algorithm

SM4 is a block cipher characterized by a block size and key length of 128 bits. It utilizes an unbalanced Feistel structure and performs 32 iterations of the round functions during both the encryption and key expansion processes. The decryption process is designed to mirror that of encryption, with the round keys applied in reverse order. SM4 algorithm flow chart as shown in Fig 3.

Symbols and acronyms indicate meanings:

$\oplus$: Denotes the bitwise exclusive OR operation performed on 32-bit words;

<<< i: Represents a circular left rotation by i bits.

Let X_0, X_1, X_2, X_3 be the round inputs and rk the round key. Then, F is defined as in Eq. (17).

$$F(X_0,X_1,X_2,X_3,rk) = X_0 \oplus T(X_1 \oplus X_2 \oplus X_3 \oplus rk) \tag{17}$$

T is an invertible transformation consisting of a nonlinear part τ and a linear part L. τ uses four parallel S-boxes. For input A = (a_0, a_1, a_2, a_3), the output B = (b_0, b_1, b_2, b_3) is shown in Eq (18).

$$B = \tau(A) = (Sbox(a_0),Sbox(a_1),Sbox(a_2),Sbox(a_3)) \tag{18}$$

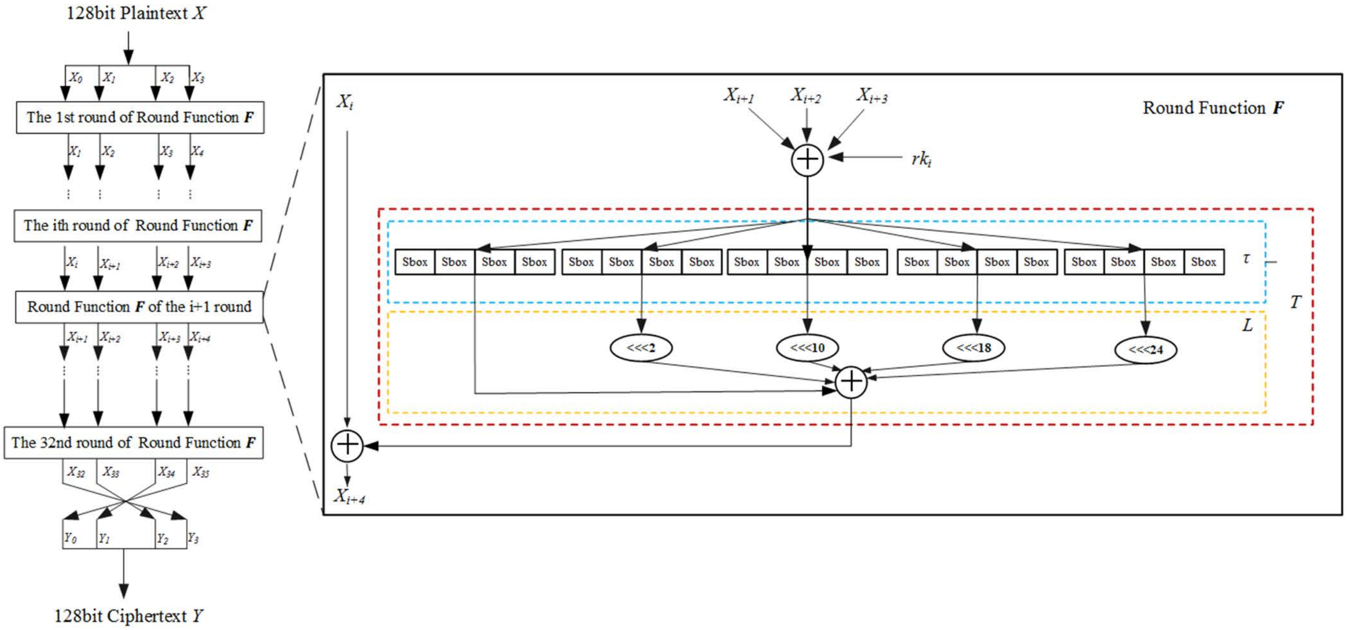

Fig 3. SM4 cryptographic algorithm flow chart.

https://doi.org/10.1371/journal.pone.0332665.g003

The output from the nonlinear transformation τ is the input to the linear transformation L. Suppose the input to L is B, and the corresponding output is C, as shown in Eq (19).

The output of τ serves as the input to the linear transformation L. Let B be the input to L, and C the corresponding output, as defined in Eq (19).

$$C = L(B) = B \oplus (B <<< 2) \oplus (B <<< 10) \oplus (B <<< 18) \oplus (B <<< 24) \quad (19)$$

The encryption process performs 32 iterations of the round function F, followed by a final reverse transformation R. The corresponding output ciphertext is (Y_0, Y_1, Y_2, Y_3) as shown in Eq (20).

$$(Y_0,Y_1,Y_2,Y_3) = R\,(X_{32},X_{33},X_{34},X_{35}) = (X_{35},X_{34},X_{33},X_{32}) \quad (20)$$

Overall security architecture

Based on the security requirements of IMS, this paper proposes a highly confidential and secure communication framework based on SM2, SM3, and SM4 algorithms, including system security authentication, data security communication, and data security storage. This paper presents a hybrid encrypted communication framework based on SM series algorithms, which facilitates the secure transfer, storage, and validation of ciphertext throughout the communication process. The framework is designed to defend against various security threats, including unauthorized access, replay attacks, eavesdropping, cryptographic cracking, and MITM attacks. The overall system framework is depicted in Fig 4.

A decentralized communication framework with hybrid encryption based on SM-series cryptographic algorithms

To guarantee the security of communication between the client and the server, this paper designs a decentralized socket-encrypted communication framework based on the SM hybrid algorithm. Encrypted Socket technology is adopted for network communication between the server and the client. The communication process is realized through sockets. The socket operating on the client side is referred to as the Client Socket, while the one on the server side is called the Server Socket. Fig 5 illustrates the communication principle of the Socket encryption framework.

After client A successfully establishes a connection with client B, it subsequently acquires client B's public key and prepares the data for transmission. Client A encrypts the data and transmits it to the server. The server then forwards the encrypted information to client B, who decrypts it layer by layer until obtaining the plaintext information. The same process applies when client B sends a message to client A. The sequence diagram for forwarding and transmitting information data between client A, client B, and the server is shown in Fig 6.

A hybrid encryption scheme for data communication based on SM2, SM3, and SM4 algorithms

After successful dual authentication between the client and server, the client is granted authorization to access the server. To ensure secure data communication among users, this paper proposes a decentralized end-to-end encrypted communication framework based on a hybrid SM algorithm. In this framework, data encryption and decryption are performed exclusively on the client side, while the server-side solely handles information forwarding, storage, and management tasks. Additionally, it facilitates ciphertext transmission and integrity verification throughout the entire process. The framework employs a symmetric SM4 algorithm along with SK for encrypting communication data; asymmetric algorithm SM2 is utilized for encrypting SK as well as random Salt authentication; furthermore, the SM3 algorithm is employed for verifying data integrity and random Salt authentication during data communication. The secure data communication framework based on a hybrid SM algorithm is shown in Fig 7.

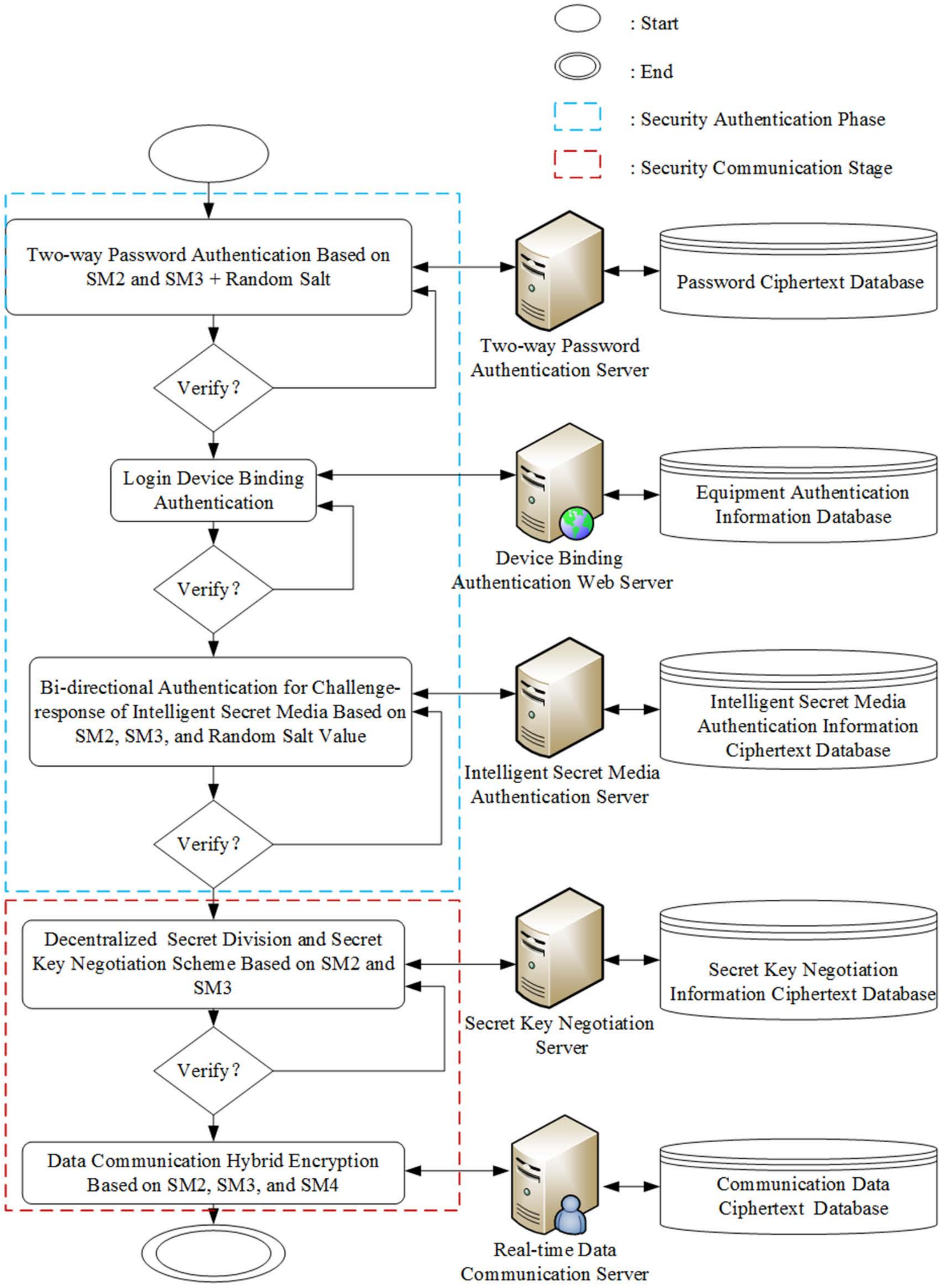
: Start
: End
: Security Authentication Phase
: Security Communication Stage
Two-way Password Authentication Based on SM2 and SM3 + Random Salt
Verify ?
Login Device Binding Authentication
Verify ?
Bi-directional Authentication for Challenge-response of Intelligent Secret Media Based on SM2, SM3, and Random Salt Value
Verify ?
Decentralized Secret Division and Secret Key Negotiation Scheme Based on SM2 and SM3
Verify ?
Data Communication Hybrid Encryption Based on SM2, SM3, and SM4
Two-way Password Authentication Server
Password Ciphertext Database
Device Binding Authentication Web Server
Equipment Authentication Information Database
Intelligent Secret Media Authentication Server
Intelligent Secret Media Authentication Information Ciphertext Database
Secret Key Negotiation Server
Secret Key Negotiation Information Ciphertext Database
Real-time Data Communication Server
Communication Data Ciphertext Database

Fig 4. The overall security architecture.

https://doi.org/10.1371/journal.pone.0332665.g004

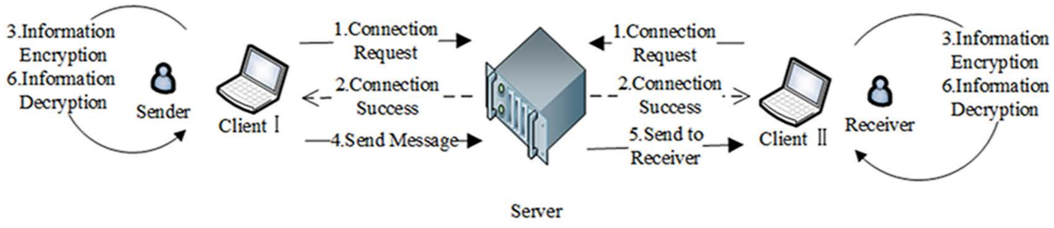

Fig 5. The schematic diagram of Encrypted Socket Communication.

https://doi.org/10.1371/journal.pone.0332665.g005

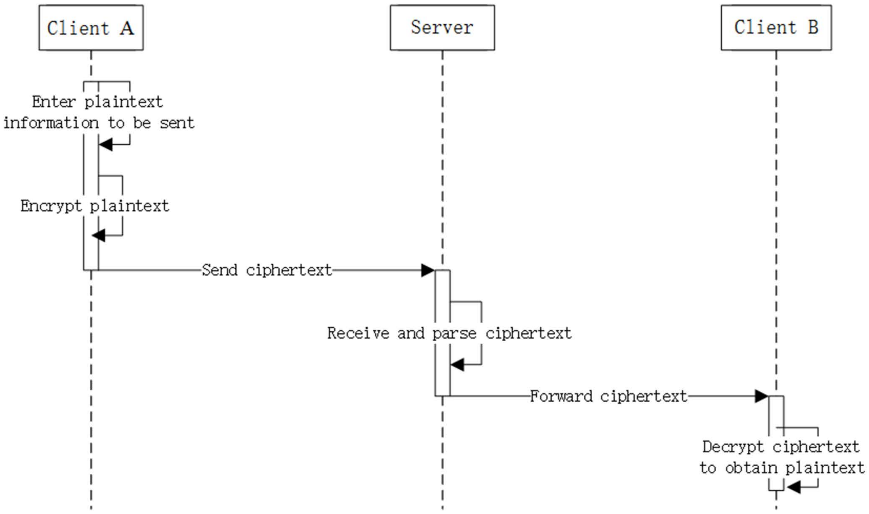

Fig 6. End-to-end communication timing diagram.

https://doi.org/10.1371/journal.pone.0332665.g006

In the context of data communication, this study integrates the SM2, SM3, and SM4 algorithms to design a hybrid encryption-based secure communication framework utilizing the SM cryptographic algorithm suite. The SM2 asymmetric cryptographic algorithm is employed to ensure the security of the key, specifically encrypting the SK of SM4. The SM4 block cipher algorithm is primarily used for encrypting communication data, while the SM3 hash algorithm is responsible for ensuring data integrity verification. The notations and descriptions used in this framework are summarized in Table 1. Prior to communication between clients, the sender must first obtain the recipient's public key. Using the SM2 algorithm, the sender encrypts both the SK and a randomly generated salt value with the recipient's public key. These encrypted components, along with verification information and the ciphertext block, are transmitted to the recipient, thereby achieving key exchange, encrypted data transmission, and data integrity verification. Upon receiving the encrypted key block and ciphertext block, the recipient decrypts the key block using their SM2 private key to retrieve the SK. This SK is then used in conjunction with the SM4 algorithm to decrypt the ciphertext block, restoring the original plaintext message. Finally, the recipient computes SM3(Salt) and sends it back to the sender for session verification, ensuring the integrity and authenticity of the communication. The SM2 public key of the receiver is denoted as RPUK, and the receiver's private key is RPRK. The plaintext information is denoted as M. Initially, the sender obtains the receiver's public key. The flowchart of the end-to-end data transmission protocol is presented in Fig 8.

Sender main steps:

Step S1: Sender generates the SK.

Step S2: The Sender generates the ciphertext block $SC_{1,}$ as shown in Eq (21).

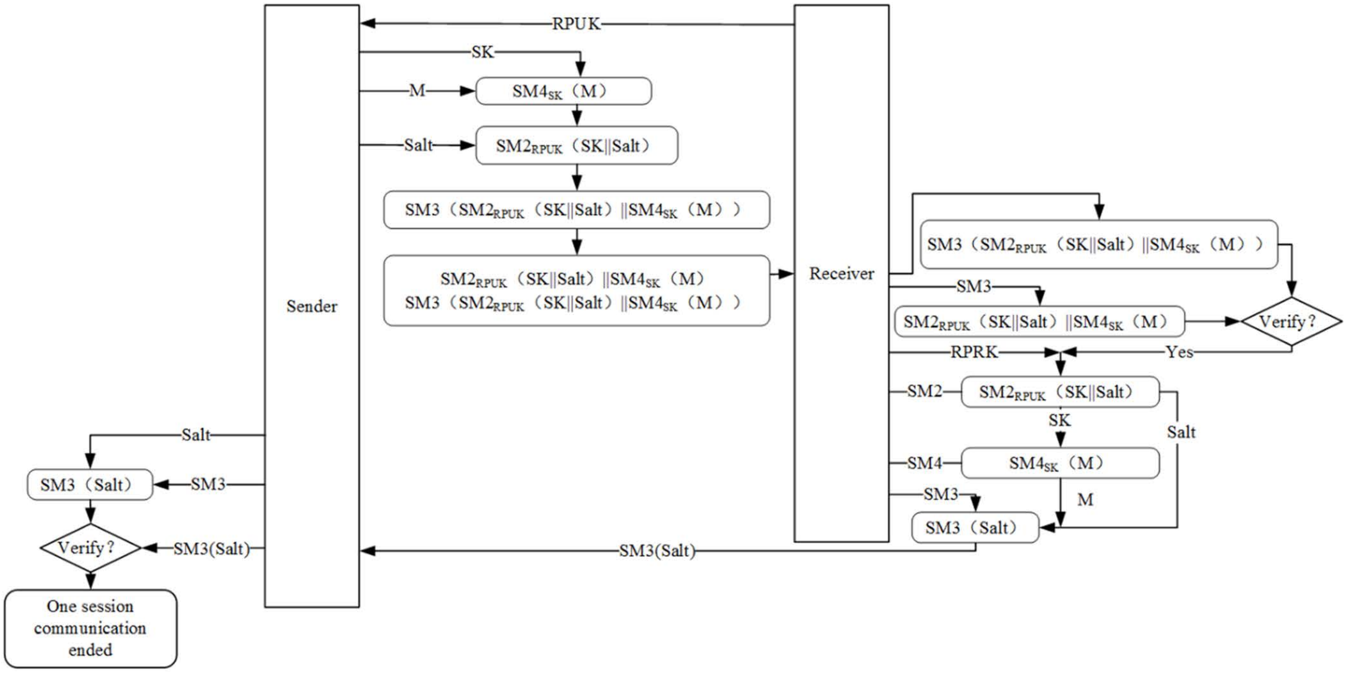
RPUK
SK
M
SM4SK（M）
Salt
SM2RPUK（SK||Salt）
SM3（SM2RPUK（SK||Salt）||SM4SK（M））
SM2RPUK（SK||Salt）||SM4SK（M）
SM3（SM2RPUK（SK||Salt）||SM4SK（M））
Sender
Receiver
SM3（SM2RPUK（SK||Salt）||SM4SK（M））
SM3
SM2RPUK（SK||Salt）||SM4SK（M）
Verify？
RPRK
Yes
SM2
SM2RPUK（SK||Salt）
SK
Salt
SM4
SM4SK（M）
SM3
M
SM3（Salt）
SM3(Salt)
Salt
SM3（Salt）
SM3
Verify？
SM3(Salt)
One session communication ended

Fig 7. The proposed hybrid encryption scheme for data communication is based on SM2, SM3, and SM4.

https://doi.org/10.1371/journal.pone.0332665.g007

Table 1. The identifiers and descriptions used in the framework.

Identifier	Interpretative Statement
Sender	The client that sends messages
Receiver	The client that receives messages
SK	Session key of SM4
M	Messages
RPUK	Receiver public key
RPRK	Receiver private key
Salt	Random salt value
T	Timestamp

https://doi.org/10.1371/journal.pone.0332665.t001

$$SC_1 = SM4_{sk}(M) \tag{21}$$

Step S3: Sender generates a random Salt value.

Step S4: The sender embeds the SK into the random Salt value and encrypts it using the SM2 algorithm and the receiver's public key (RPUK) to generate the ciphertext block SC_2, as shown in Eq (22).

$$SC_2 = SM2_{RPUK}(SK \mid\mid Salt) \tag{22}$$

Step S5: The sender uses the SM3 algorithm to generate the ciphertext block SC_3 as shown in Eq (23).

$$SC_3 = SM3(SM4_{sk}\,(M)\ \mid\mid SM2_{RPUK}(SK \mid\mid Salt)) \tag{23}$$

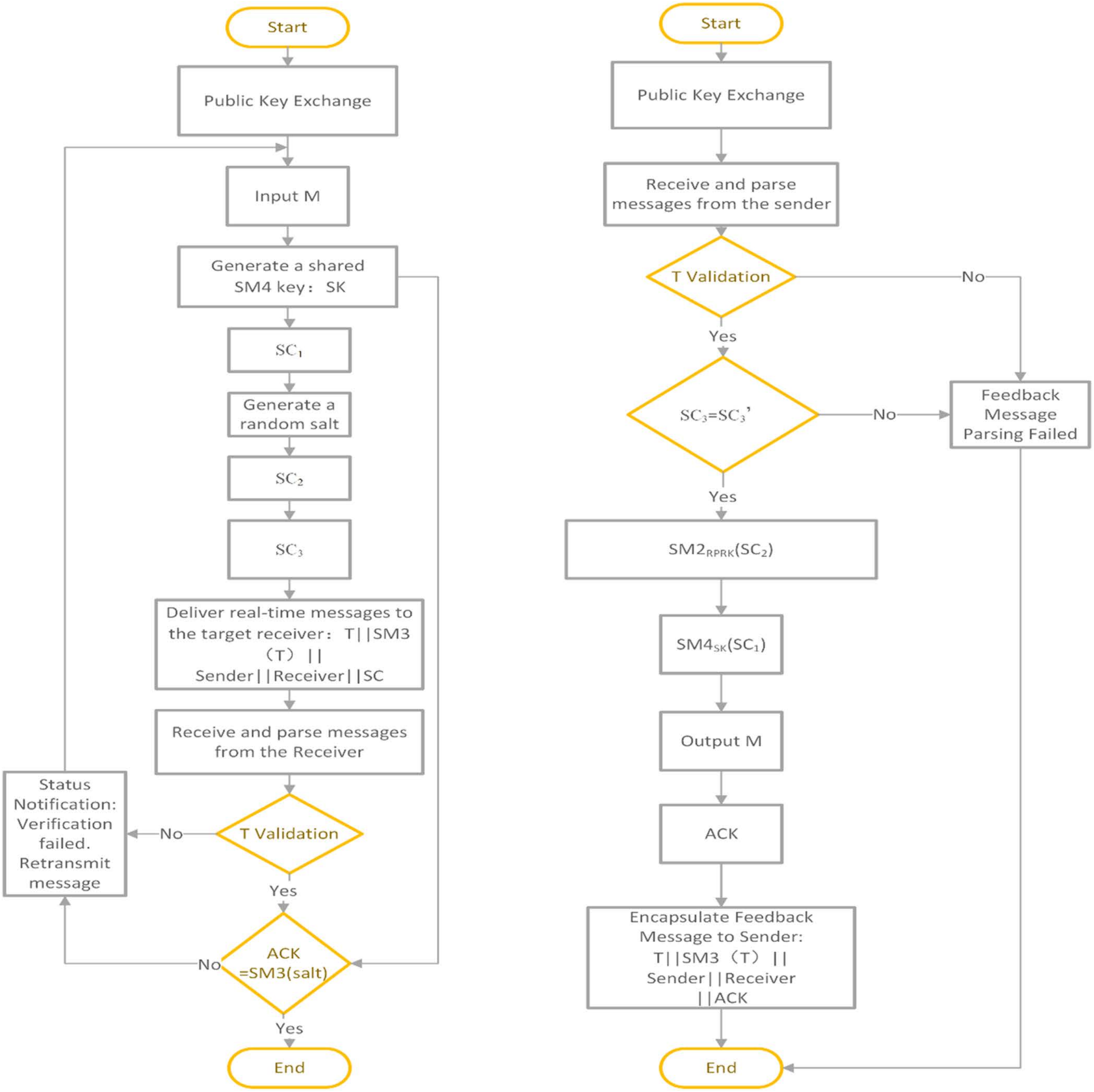
Start
Public Key Exchange
Input M
Generate a shared SM4 key： SK
SC₁
Generate a random salt
SC₂
SC₃
Deliver real-time messages to the target receiver： T||SM3（T）|| Sender||Receiver||SC
Receive and parse messages from the Receiver
T Validation
No
Status Notification: Verification failed. Retransmit message
Yes
ACK =SM3(salt)
No
Yes
End
Start
Public Key Exchange
Receive and parse messages from the sender
T Validation
No
Yes
SC₃=SC₃'
No
Feedback Message Parsing Failed
Yes
SM2RPRK(SC₂)
SM4SK(SC₁)
Output M
ACK
Encapsulate Feedback Message to Sender: T||SM3（T）|| Sender||Receiver ||ACK
End

Fig 8. End-to-end data transmission protocol flowchart.

https://doi.org/10.1371/journal.pone.0332665.g008

Step S6: The sender sends the ciphertext block SC, as shown in Eq (24), to the receiver.

$$SC = SC_1 \parallel SC_2 \parallel SC_3 \tag{24}$$

Receiver main steps:

Step R1: The receiver obtains the ciphertext block SC and extracts $SC_{1,}$ SC_2, and SC_3 separately. The receiver verifies it using the SM3 algorithm, calculating SC_3' as shown in the Eq (25). If $SC_3 = SC_3$', the verification is successful.

$$SC_3' = SM3(SC_1 \parallel SC_2) \tag{25}$$

Step R2: The receiver uses the RPRK of the SM2 algorithm to decrypt the SC_2 block, obtaining the value SK || Salt, as shown in Eq (26), and then extracts the Salt and SK values using the Salt separation algorithm.

$$SK \parallel Salt = SM2_{RPRK}(SC_2) \tag{26}$$

Step R3: The receiver uses the SM4 algorithm and SK to decrypt the ciphertext block SC_1 and obtain the plaintext M, as shown in Eq (27).

$$M = SM4_{sk}(SC_1) \tag{27}$$

Step R4: The receiver sends an acknowledgment (ACK) confirmation message to the sender, as shown in Eq (28).

$$ACK = SM3(Salt) \tag{28}$$

Finally, the sender uses the SM3 algorithm and Salt to compute SM3(Salt), which is then compared with the received ACK for verification. If the verification is successful, the session process concludes; otherwise, the session fails.

Experimental environment

All experiments were conducted on a Windows 11 Professional workstation with an Intel Core i9-9880H CPU and 32 GB DDR4 RAM. The encryption algorithms were implemented in Java (JDK 17) with IntelliJ IDEA 2024.1.7 (Ultimate Edition) and the Bouncy Castle cryptography library. Hardware acceleration instructions (AES-NI and SHA-NI) were disabled to ensure consistent measurement results.

Results

The results of confidentiality, integrity, and availability testing

In the experiment, the memory values of the receiver and sender are obtained, and the packets of the sender and receiver are captured and analyzed by Wireshark, a packet capture and analysis tool, to obtain the communication data. The experimental results of the hybrid encryption scheme for data communication based on SM2, SM3, and SM4 in the proposed framework are presented in Table 2.

Benchmarking results of SM2, SM3, and SM4 cryptographic algorithms

Under identical experimental conditions, implemented exclusively using the Bouncy Castle cryptographic library and without utilizing CPU (Intel Core i9 9880H) hardware acceleration instructions, a comparative analysis of performance was conducted between the SM2, SM3, and SM4 cryptographic algorithms and their respective international counterparts: RSA-3072, SHA-256, and AES-256. To enhance the measurement accuracy of the experimental results, this study adopts byte (Byte) and nanosecond (ns) as the basic units of measurement. These high-precision time units and standardized data measurement units can more accurately reflect the system performance indicators and provide a reliable quantitative basis for subsequent data analysis.

Table 2. The test results of the hybrid encryption scheme for data communication.

Data block name	Length (byte)	Data (hex)
Sender's plaintext data: M	60	48656c6c6f2c20746869732069732064792062616e6b206163636f756e742e20506c65617365206b65657020697420636f6e666964656e7469616c21
SK	16	D65F878CE8E6DC3F2428382540845DDE
SC_1	64	4361E9EE86FA4828E80EB2C39A418351417DA3E558A447C98E7C9A9E31D7A2D8CEEA034ED01C6EB35D8E9F9158B466808A7C902E6F7DB95DC2D6E332CB5F2BC4
Salt	16	e9e679e5b90a03fd918248f51f27d265
RPUK	64	d490b5099d99b2ae4b4d7941ea341b59bd960726b21a9296176947063dcc4edccf1001590e64ff30d038544258688aa36d70c3ccd546534f9659b75a20d96eff
SC_2	152	d2f3ce73717fc67c0ebd0938ddca2d76611b7726fcae1 cd6fcf8d3d97c5d7ed206eacbf5adbbe188dc319f9711d511d418fd38f4f8679d0358c854578c26ebf49daee3fb6129564dbbfe2af934e7cd6abe198dbfab097a168fbf36131bec99aba13c04f30c5ffae0a26bc22de38c5de03b9b13ce5369d5e1917d98d55f76c9fb7e02cfde79b5a34f7ff4888f7c37b4629028ad3632faccef
SC_3	32	79202867054278d7651ea824b16e4440156846c1b6c17e34248f4907f4f077a5
Sender's ciphertext data: SC	248	4361E9EE86FA4828E80EB2C39A418351417DA3E558A447C98E7C9A9E31D7A2D8CEEA034ED01C6EB35D8E9F9158B466808A7C902E6F7DB95DC2D6E332CB5F2BC4d2f3ce73717fc67c0ebd0938ddca2d76611b7726fcae1 cd6fcf8d3d97c5d7ed206eacbf5adbbe188dc319f9711d511d418fd38f4f8679d0358c854578c26ebf49daee3fb6129564dbbfe2af934e7cd6abe198dbfab097a168fbf36131bec99aba13c04f30c5ffae0a26bc22de38c5de03b9b13ce5369d5e1917d98d55f76c9fb7e02cfde79b5a34f7ff4888f7c37b4629028ad3632faccef79202867054278d7651ea824b16e4440156846c1b6c17e34248f4907f4f077a5
Ciphertext data in transmission	248	4361E9EE86FA4828E80EB2C39A418351417DA3E558A447C98E7C9A9E31D7A2D8CEEA034ED01C6EB35D8E9F9158B466808A7C902E6F7DB95DC2D6E332CB5F2BC4d2f3ce73717fc67c0ebd0938ddca2d76611b7726fcae1 cd6fcf8d3d97c5d7ed206eacbf5adbbe188dc319f9711d511d418fd38f4f8679d0358c854578c26ebf49daee3fb6129564dbbfe2af934e7cd6abe198dbfab097a168fbf36131bec99aba13c04f30c5ffae0a26bc22de38c5de03b9b13ce5369d5e1917d98d55f76c9fb7e02cfde79b5a34f7ff4888f7c37b4629028ad3632faccef79202867054278d7651ea824b16e4440156846c1b6c17e34248f4907f4f077a5
Receiver's data: SC'	248	4361E9EE86FA4828E80EB2C39A418351417DA3E558A447C98E7C9A9E31D7A2D8CEEA034ED01C6EB35D8E9F9158B466808A7C902E6F7DB95DC2D6E332CB5F2BC4d2f3ce73717fc67c0ebd0938ddca2d76611b7726fcae1 cd6fcf8d3d97c5d7ed206eacbf5adbbe188dc319f9711d511d418fd38f4f8679d0358c854578c26ebf49daee3fb6129564dbbfe2af934e7cd6abe198dbfab097a168fbf36131bec99aba13c04f30c5ffae0a26bc22de38c5de03b9b13ce5369d5e1917d98d55f76c9fb7e02cfde79b5a34f7ff4888f7c37b4629028ad3632faccef79202867054278d7651ea824b16e4440156846c1b6c17e34248f4907f4f077a5
RPRK	32	620880abd1a3aac4981b18d72ebd129f735ff4e795e98b0702ed9a3e2098e7d7
SC_3'	32	79202867054278d7651ea824b16e4440156846c1b6c17e34248f4907f4f077a5
Receiver's decrypted data: SK'	16	D65F878CE8E6DC3F2428382540845DDE
Receiver's decrypted data: Salt'	16	e9e679e5b90a03fd918248f51f27d265
Receiver's decrypted data: M'	60	48656c6c6f2c20746869732069732064792062616e6b206163636f756e742e20506c65617365206b65657020697420636f6e666964656e7469616c21

https://doi.org/10.1371/journal.pone.0332665.t002

Performance testing results of SM2 and RSA-3072

Under the same conditions, in this study, for small data blocks not exceeding 32 bytes, a performance comparison test was conducted between the SM2 algorithm and the RSA-3072 algorithm. It consists of three parts: encryption and decryption, and key generation. The performance test results are shown in Table 3.

Table 3. The performance testing results of the SM2 and RSA-3072.

Data size (bytes)	SM2			RSA-3072		
	Encryption (ns)	Decryption (ns)	Key generation (ns)	Encryption (ns)	Decryption (ns)	Key generation (ns)
32	2154627	1591348	609266	97343	3500014	391766550
16	2142340	1580580	609266	96765	3492717	391766550
8	2163487	1587019	609266	96551	3493019	391766550

https://doi.org/10.1371/journal.pone.0332665.t003

Performance testing results of SM3 and SHA-256

This study compared the computational performance of the SM3 and SHA-256 algorithms in a standard experimental environment. The detailed results of the performance tests are presented in Table 4.

Performance testing results of SM4 and AES-256

Under the same experimental conditions, performance tests for encryption and decryption of the SM4 algorithm were conducted in comparison with the AES-256 algorithm. The experimental results are presented in Table 5.

Evaluation results of security performance

The hybrid security framework proposed in this study, which integrates the SM2, SM3, and SM4 cryptographic algorithms with dynamic salt values, demonstrates comprehensive protection against mainstream security threats in instant messaging systems. Security performance evaluation results are shown in Table 6.

Discussion

Through the data analysis of the experimental results in Table 2, the experimental verification results, as shown in Table 7, were obtained.

According to the experimental verification results in Table 7, the IMS secure communication scheme proposed in this paper achieves the design goals in all three core security dimensions. Firstly, by comparing and analyzing the consistency

Table 4. The performance testing results of the SM3 and SHA-256.

Data size (bytes)	SM3	SHA-256
	Time (ns)	Time (ns)
8	394	445
16	403	456
32	414	469
64	773	850
128	1152	1261
256	1924	2097
512	3431	3729
1024	6451	6954
2048	12315	13462
4096	24289	26326
8192	48064	52279
16384	95225	103789
32768	189877	205891
65536	386191	411346
131072	759878	819850

https://doi.org/10.1371/journal.pone.0332665.t004

Table 5. The performance testing results of SM4 and AES-256.

Data size (bytes)	SM4			AES-256		
	Encryption time (ns)	Decryption time (ns)	Total time (ns)	Encryption time (ns)	Decryption time (ns)	Total time (ns)
8	1721	1742	3464	2150	2287	4437
16	1941	1934	3874	2307	2424	4731
32	2160	2141	4300	2457	2537	4994
64	2597	2554	5151	2741	2770	5510
128	3348	3376	6724	3283	3262	6545
256	4950	4935	9885	4361	4192	8553
512	8170	8185	16354	6496	6058	12554
1024	14576	14490	29066	10731	9838	20569
2048	27509	27180	54689	19190	17193	36383
4096	53121	52718	105839	36226	32348	68574
8192	104157	103634	207790	69694	62042	131736
16384	206636	204806	411442	137587	121717	259303
32768	409587	407611	817198	272212	240724	512936
65536	817139	814137	1631276	539627	478569	1018196
131072	1630337	1624348	3254685	1072700	950883	2023583

https://doi.org/10.1371/journal.pone.0332665.t005

Table 6. Evaluation results of security performance.

Security mechanism	Information leakage	Credential theft	MITM attacks	Server attacks	Database attacks	Network sniffing
Bidirectional authentication [35]	●	●	●	●		●
Decentralized architecture				●	●	
Hybrid message encryption	●		●	●	●	●
Ciphertext transmission and storage	●	●	●	●	●	●

(● Indicates effective protection, blank indicates non-applicability).

https://doi.org/10.1371/journal.pone.0332665.t006

Table 7. Confidentiality, integrity, and availability analysis results.

Sender's data block name	Receiver's data block name	Experimental verification results	Confidentiality, integrity, and availability results
M	M'	M = M'	Confidentiality and availability are ok
SK	SK'	SK = SK'	Confidentiality and availability are ok
SC3	SC3'	SC3 = SC3'	Integrity is ok
SC	SC'	SC = SC'	Confidentiality is ok

https://doi.org/10.1371/journal.pone.0332665.t007

of the original plaintext data block (M, SK) at the sender's end and the decrypted and restored data block (M', SK') at the receiver's end (M ≡ M' and SK ≡ SK'), it is confirmed that the system simultaneously meets the requirements of confidentiality and availability at the message transmission layer. No plaintext data leakage occurred during the communication process, and the data packets were complete and could be correctly parsed. Secondly, the verification value SC3 generated by the SM3 hash algorithm is completely consistent with the calculated value SC3' at the receiver's end (SC3 ≡ SC3'), which verifies that the integrity protection mechanism during data transmission can effectively resist MITM tampering attacks. Finally, through network packet capture analysis, it is confirmed that the encrypted data SC remains completely consistent with the received data SC' at the receiver's end during transmission (SC ≡ SC'). This result not only verifies the correctness of the encryption algorithm but also indicates that the system can ensure that sensitive information does not leak during transmission over public channels, meeting the confidentiality requirements of high-security-level instant messaging.

Experimental analysis of the performance comparison between SM2 and RSA-3072 algorithms (Table 3) reveals significant operational disparities when processing data blocks ≤32 bytes. During the encryption and decryption test, each set of data undergoes 1,000 cycles of calculation. After the same 30 rounds of calculation, the obtained experimental data were sorted in descending order and processed using the median truncation method: the first 10 maximum values and the last 10 minimum values are removed, and the arithmetic mean of the 10 valid data points in the middle is calculated. Meanwhile, 1000 key pairs were calculated respectively for the SM2 and RSA-3072 algorithms. The median truncation method was also adopted for processing, and the arithmetic mean of the generation time of the middle 600 pairs of keys was taken. The cryptographic evaluation demonstrates that RSA-3072 exhibits superior encryption efficiency, with execution times ranging 96,551–97,343 ns, representing merely 4.55% (1/22) of SM2's encryption duration (2,142,340–2,163,487 ns). Conversely, SM2 demonstrates a remarkable decryption performance advantage, operating at 1,580,580–1,591,348 ns compared to RSA-3072's 3,492,717-3,500,014 ns, achieving approximately 2.2 × faster processing speed. The most pronounced performance divergence occurs during the key generation phase, where SM2 completes the operation in 609,266 ns versus RSA-3072's 391,766,550 ns, exhibiting a 642:1 performance ratio. As illustrated in Fig 9, a comprehensive evaluation confirms SM2's dominant performance characteristics in small-data cryptographic scenarios. For 32-byte data processing, SM2's overall performance surpasses RSA-3072 by a factor of 90.78, primarily attributable to RSA's computationally intensive key generation mechanism. In our proposed scheme, SM2 demonstrates particular efficacy in handling 16-byte session keys (SK) and 16-byte salt values, where rapid key deployment is essential. This performance advantage, coupled with equivalent security guarantees, establishes SM2 as an optimal cryptographic solution for lightweight IMS communication architectures.

Benchmark testing of hash algorithms in Table 4 demonstrates that SM3 exhibits significant efficiency advantages across varying data scales, as illustrated in Fig 10. Each algorithm was executed for 20,000 iterations, and 30 independent tests were conducted. To ensure robust data processing, a median-trimmed method was employed: after sorting the durations from the 30 test rounds, the top and bottom 10 extreme values were excluded, and the average of the remaining 10 rounds was calculated. When processing 8-byte data packets, SM3 achieves a processing time of 394 ns,

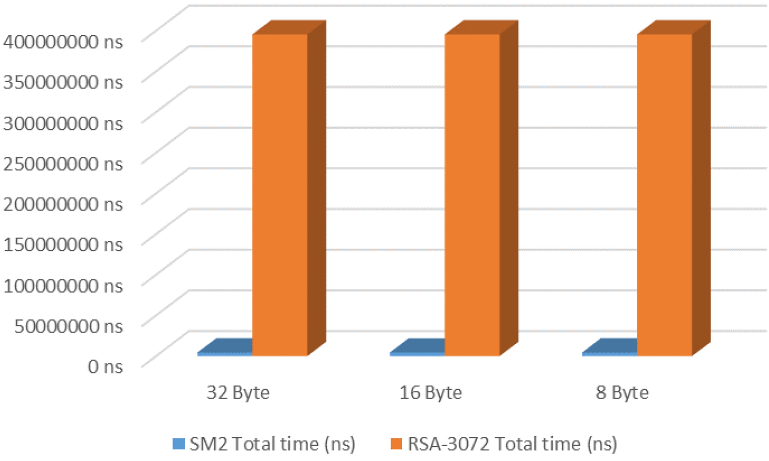

Fig 9. Overall performance comparison of SM2 and RSA-3072 algorithms.

https://doi.org/10.1371/journal.pone.0332665.g009

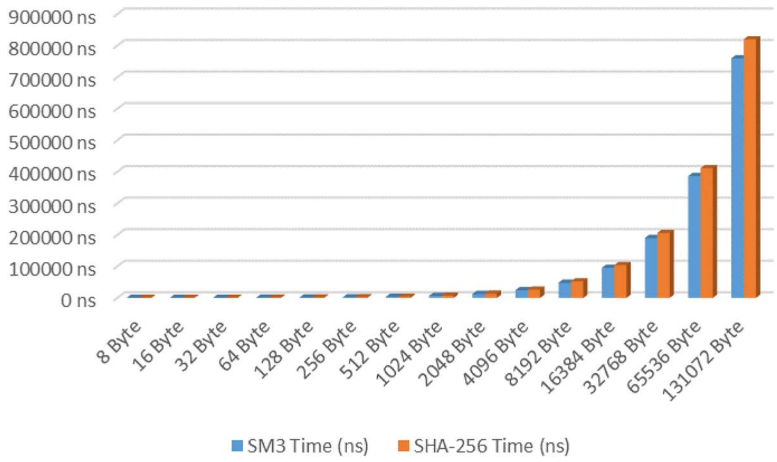

Fig 10. The performance comparison between SM3 and SHA-256.

https://doi.org/10.1371/journal.pone.0332665.g010

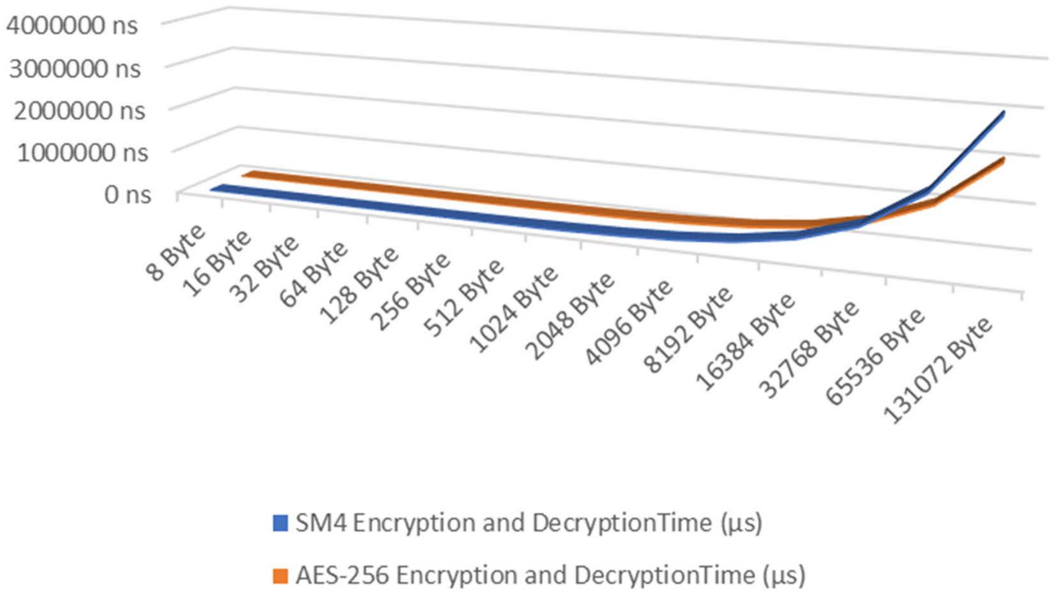

Fig 11. The performance comparison between SM4 and AES-256.

https://doi.org/10.1371/journal.pone.0332665.g011

representing an 11.5% throughput efficiency improvement over SHA-256 (445 ns). At the extended data volume of 128KB (131,072 bytes), SM3 maintains a 7.3% performance advantage with a processing duration of 759.9 μs. Experimental results confirm that compared to internationally prevalent algorithms, SM3 demonstrates superior operational suitability for IMS instant messaging scenarios, providing empirical support for its implementation in IMS systems.

The experimental data presented in Table 5 illustrate the performance comparison between the SM4 and AES-256 encryption algorithms, as shown in Fig 11. Each dataset was processed in a loop for 10000 iterations. Following 30 rounds of computation, the results from these rounds were sorted, and the top 10 as well as the bottom 10 results were excluded. Only the average value of the middle 10 rounds of test outcomes was retained. The SM4 algorithm demonstrates significantly superior encryption and decryption performance compared to AES-256 within the data range of 8–64 bytes. For 8-byte data, the average time taken by SM4 is 1731.5 ns, which represents a reduction of 21.95% compared to AES-256’s time of 2218.5 ns; for 16-byte data, SM4 takes an average of 1937.5 ns, reducing the time by 18.10% relative to AES-256’s 2365.5 ns; at the 32-byte mark, SM4 requires an average of 2150.5 ns, showing a decrease of 13.88% when compared with AES-256’s time of 2497 ns; finally, for data sizes up to 64 bytes, SM4 averages at 2575.5 ns—6.53% less than AES-256’s duration of 2755.5 ns. Considering the overall performance from sizes ranging between 8 and 64 bytes, there exists a notable advantage in encryption and decryption times for the SM4 algorithm that spans from approximately 6% to 22% over its counterpart. While it is acknowledged that AES-256 exhibits higher throughput with larger datasets (greater than 512 bytes), SM4 proves more suitable for scenarios involving smaller amounts of data—particularly in fulfilling short message encryption requirements within IMS —making it ideally suited for real-time interactive applications.

According to Table 6 of the experimental evaluation results, the comprehensive protection effectiveness of the security framework based on the SM series hybrid algorithm in the instant messaging system has been verified. The proposed security scheme was evaluated and analyzed by using the Kali Linux network penetration testing method [36]. The multi-layer defense architecture effectively mitigates six categories of security threats through four interlocking security mechanisms. This framework achieves multi-dimensional security protection through the following technical features: Firstly, it adopts a two-way authentication mechanism based on national encryption standards. It has been measured that it can effectively resist MITM attacks and credential theft behaviors. Secondly, the innovative decentralized architecture design, through the deployment of distributed nodes, significantly reduces the risk of a single point of failure and enhances system availability. Thirdly, implement a full-link encryption strategy, covering the data transmission and storage links, to ensure the confidentiality and integrity of end-to-end communication; Finally, through the organic synergy of the SM2, SM3, and SM4 algorithms, a multi-level defense system was constructed. Each algorithm complemented the other’s strengths, resulting in a significant improvement in security while maintaining system performance. This hierarchical and progressive security architecture design provides a solution that takes into account both security and availability for instant messaging systems.

The obtained results have several practical implications for the deployment of secure instant messaging systems (IMS). First, the demonstrated efficiency improvements of SM2, SM3, and SM4 over traditional algorithms indicate that the proposed framework can be integrated into real-time communication platforms without introducing additional latency, thereby ensuring a seamless user experience. Second, the decentralized and certificateless design reduces reliance on centralized authentication servers, which not only mitigates single-point failure risks but also enhances system robustness against targeted attacks. Third, the client-side encryption in combination with the encrypt-before-store principle provides an additional layer of protection for sensitive data, which is particularly valuable in high-security domains such as finance, enterprise communication, and public services. Overall, these practical benefits underscore the potential of the proposed solution to strengthen the confidentiality, integrity, and availability of IMS in real-world applications.

Despite the demonstrated security and efficiency advantages of the proposed hybrid encryption framework in lightweight, real-time IMS applications, several limitations should be noted. The framework is primarily tailored for short-message scenarios, and its relatively high design complexity may lead to reduced performance when processing large data blocks or operating in resource-constrained environments. Future work will aim to optimize processing throughput for large data blocks and improve performance on resource-limited devices, while preserving the high security guarantees established in the current experiments.

Conclusions

This study innovatively proposes a hybrid encryption security framework based on the SM2, SM3, and SM4 algorithms, significantly enhancing the security protection capabilities of IMS. Experimental results demonstrate that this framework successfully achieves triple guarantees of data confidentiality, integrity, and availability while maintaining communication efficiency. The proposed solution leverages the complementary advantages of the SM2, SM3, and SM4 algorithms. In scenarios involving small data sizes, it exhibits remarkable performance benefits. Specifically, the performance of the SM2 algorithm in key generation and decryption stages improves by 642 times and 2.2 times compared to RSA-3072; the processing speed of the SM3 hashing algorithm surpasses SHA-256 by 7.3% to 11.5%; furthermore, for encryption efficiency concerning small data blocks ranging from 8 to 64 bytes, the SM4 algorithm outperforms AES-256 by achieving an improvement of up to 22%. Through innovative decentralized architecture design and end-to-end full-link encryption strategies, this framework effectively mitigates security threats such as MITM attacks and data tampering while establishing an efficient end-to-end encrypted transmission system for IMS. Future research will refine the security scheme by optimizing processing throughput for large data blocks and enhancing performance in resource-constrained environments, ultimately strengthening cross-platform compatibility.

Author contributions

Conceptualization: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Data curation: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Formal analysis: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Funding acquisition: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Investigation: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Methodology: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Project administration: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Resources: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Software: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Supervision: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Validation: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Visualization: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Writing – original draft: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

Writing – review & editing: He-Jun Lu, Roben A. Juanatas, Mideth B. Abisado.

References

1. China Internet Network Information Center. The 55th statistical report on China's internet development [Internet]. Beijing: China Internet Network Information Center; 2025 Jan 17 [Accessed 2025 April 6]. https://cnnic.cn/NMediaFile/2025/0313/MAIN17418452848150SDUMQZGSU.pdf
2. Ali G, Ally Dida M, Elikana Sam A. Two-factor authentication scheme for mobile money: a review of threat models and countermeasures. Future Internet. 2020;12(10):160. https://doi.org/10.3390/fi12100160
3. Lu Y-H, Hsiao SH-Y, Li C-Y, Hsieh Y-C, Chou P-Y, Li Y-Y, et al. Insecurity of operational IMS call systems: vulnerabilities, attacks, and countermeasures. IEEE/ACM Trans Netw. 2023;31(2):800–15. https://doi.org/10.1109/tnet.2022.3205183
4. Domenech MC, Abed Gregio AR, Erpen de Bona LC. On metadata privacy in instant messaging. In: 2022 IEEE Symposium on Computers and Communications (ISCC). Greece: IEEE; 2022: 1–7. https://doi.org/10.1109/iscc55528.2022.9912901
5. Shi J, Wang S, Chen M-Y, Tu G-H, Xie T, Chen M-H, et al. IMS is not that secure on your 5G/4G phones. In: Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. New York, NY: ACM; 2024: 513–27. https://doi.org/10.1145/3636534.3649377
6. Cheng Z, Ordean M, Garcia F, Cui B, Rys D. Watching your call: breaking VoLTE privacy in LTE/5G networks. Proc Priv Enhancing Technol. 2023;2023(2):282–97. https://doi.org/10.56553/popets-2023-0053
7. Wang S, Huang K, Ma K, Xu X, Hu X. A lightweight encryption and message authentication framework for wireless communication. IET Commun. 2022;17(3):265–78. https://doi.org/10.1049/cmu2.12535

8. Yu C, Chen S, Wei Z, Wang F. Toward a truly secure telecom network: analyzing and exploiting vulnerable security configurations/ implementations in commercial LTE/IMS networks. IEEE Trans Dependable Secure Comput. 2024;21(4):3048–64. https://doi.org/10.1109/tdsc.2023.3322267

9. Kara M, Merzeh HRJ, Aydın MA, Balık HH. VoIPChain: a decentralized identity authentication in Voice over IP using Blockchain. Comput Commun. 2023;198:247–61. https://doi.org/10.1016/j.comcom.2022.11.019

10. ISO/IEC 14888-3:2018 information technology — security techniques — digital signatures with appendix — part 3: discrete logarithm based mechanisms. Geneva: ISO/IEC; 2018. https://www.iso.org/standard/76382.html

11. International Organization for Standardization. ISO/IEC 10118-3:2018 information technology — security techniques — hash-functions — part 3: dedicated hash-functions. Geneva: ISO/IEC; 2018. https://www.iso.org/standard/67116.html

12. International Organization for Standardization. ISO/IEC 18033-3:2010/Amd 1:2021 information technology — security techniques — encryption algorithms — part 3: block ciphers — amendment 1: SM4. Geneva: ISO/IEC; 2021. https://www.iso.org/standard/81565.html

13. Li M, Li N, Liu H, Cheng S, Hu X, Li J. Implementation of a safe and efficient point multiplication for SM2 algorithm. In: 2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC). 2023: 137–41. https://doi.org/10.1109/itnec56291.2023.10082311

14. Zhai F, Qin Y, Lin C, Ni W, Xu M, Liang X. An efficient and secure distributed scheme for SM2 decryption in Internet of Things. In: Seventh International Conference on Advanced Electronic Materials, Computers, and Software Engineering (AEMCSE 2024). 2024: 132. https://doi.org/10.1117/12.3038791

15. Cao Z, Gao B, Xiong X, Liu Z. An improved SM2 digital signature algorithm with high-precision timestamps for trusted metrological data. Sensors (Basel). 2025;25(16):4920. https://doi.org/10.3390/s25164920 PMID: 40871784

16. Zhang Y, Lei H, Wang B, Wang Q, Lu N, Shi W, et al. Traceable ring signature schemes based on SM2 digital signature algorithm and its applications in the data sharing scheme. Front Comput Sci. 2024;18(2). https://doi.org/10.1007/s11704-023-3318-z

17. Wu K, Cheng R, Cui W, Li W. A lightweight SM2-based security authentication scheme for smart grids. Alexandria Eng J. 2021;60(1):435–46. https://doi.org/10.1016/j.aej.2020.09.008

18. Xu J, Feng D. Comments on the SM2 key exchange protocol. In: Lin D, Tsudik G, Wang X, eds. Lecture notes in computer science. Springer Berlin Heidelberg; 2011: 160–71. https://doi.org/10.1007/978-3-642-25513-7_12

19. Zhao C, Zhang J, Huang X. Unbalancing computations for SM2 key exchange protocols. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). 2020: 335–41. https://doi.org/10.1109/itnec48623.2020.9085115

20. Jayakumari D, Rupa Ch, Nikhila V. Pentatope based elliptic curve encryption for privacy and protection of multimedia data. In: Komanapalli VLN, Sivakumaran N, Hampannavar S, eds. Lecture notes in electrical engineering. Springer Nature Singapore; 2021: 551–60. https://doi.org/10.1007/978-981-15-8221-9_48

21. Prabhu Kavin B, Ganapathy S, Kanimozhi U, Kannan A. An enhanced security framework for secured data storage and communications in cloud using ECC, access control and LDSA. Wireless Pers Commun. 2020;115(2):1107–35. https://doi.org/10.1007/s11277-020-07613-7

22. Xiao Y, Zhang L, Yang Y, Wu W, Ning J, Huang X. Provably secure multi-signature scheme based on the standard SM2 signature scheme. Comput Stand Interfaces. 2024;89:103819. https://doi.org/10.1016/j.csi.2023.103819

23. National Institute of Standards and Technology. Recommendation for key management: part 1 – general (SP 800-57 Part 1 Rev. 5). Gaithersburg, MD: NIST; 2020.

24. Zheng X, Hu X, Zhang J, Yang J, Cai S, Xiong X. An efficient and low-power design of the SM3 hash algorithm for IoT. Electronics. 2019;8(9):1033. https://doi.org/10.3390/electronics8091033

25. Han J, Peng T, Zhang X. A CUDA-based parallel optimization method for SM3 hash algorithm. J Supercomput. 2024;80(14):21431–46. https://doi.org/10.1007/s11227-024-06141-6

26. Stevens M, Lenstra A, de Weger B. Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities. In: Naor M ed. Lecture notes in computer science. Springer Berlin Heidelberg; 2007: 1–22. https://doi.org/10.1007/978-3-540-72540-4_1

27. Leurent G, Peyrin T. SHA-1 is a shambles: first chosen-prefix collision on SHA-1. In: 29th USENIX Security Symposium (USENIX Security 2020). 2020. 1839–56. https://www.usenix.org/conference/usenixsecurity20/presentation/leurent

28. Guo Z, Wang G, Dunkelman O, Pan Y, Liu S. Tweakable SM4: How to tweak SM4 into tweakable block ciphers? J Inf Secur Appl. 2023;72:103406. https://doi.org/10.1016/j.jisa.2022.103406

29. Zhang L, Feng Z, Yu Y, Zhou H. Bit-sliced implementation of SM4 and new performance records on x86. IET Comput Digit Tech. 2023;17(6):589–99. https://doi.org/10.1049/2023/1821499

30. Hu J, Yang Y, Xiong L. SM algorithm analysis and software performance research. Netinfo Sec. 2021;21(10):8–16. https://doi.org/10.3969/j.issn.1671-1122.2021.10.002

31. Liu F, Wang DD, Yu B, Yu F. Instant messaging system based on hybrid 3DES and RC4 algorithm. Comput Syst Appl. 2020;29(8):80–9. https://doi.org/10.15888/j.cnki.csa.007564

32. Tajudeen KO, Ameen AO, Adeniyi AE. A systematic review on advanced encryption standard cryptography to enhance message security. Multimed Tools Appl. 2025. https://doi.org/10.1007/s11042-025-21041-4

33. Kasar M, Suryawanshi T, Kavimandan P, Chaudhari S, Deshmukh S, Jadhav P. An efficient approach for decentralized web messaging using WebRTC. In: Kumar A, Swaroop A, Shukla P, eds. Lecture notes in networks and systems. Springer Nature Singapore; 2025: 17–32. https://doi.org/10.1007/978-981-96-3942-7_2

34. Zhou X, Lan Q, Zheng J, Wan F. Design and implementation of instant messaging service system for internet enterprises based on hierarchical protection. In: 2023 International Seminar on Computer Science and Engineering Technology (SCSET). New York, NY, USA: IEEE. 2023: 426–9. https://doi.org/10.1109/scset58950.2023.00100

35. Lu H, Juanatas RA. Design and research of a dual and bidirectional certificateless identity authentication scheme based on hybrid SM series algorithms. In: Proceedings of the 2023 11th International Conference on Computer and Communications Management. New York, NY: ACM. 2023: 41–7. https://doi.org/10.1145/3617733.3617740

36. Lu H-J, Yu Y. Research on WiFi penetration testing with Kali Linux. Complexity. 2021;2021(1). https://doi.org/10.1155/2021/5570001