INTERNET OF THINGS ACCESS CONTROL IDENTITY AUTHENTICATION METHOD BASED ON BLOCKCHAIN

Peng ZHAI[1], Liping ZHANG[2], Yu ZHANG[3]*

At present, Internet of Things (IoT) is widely used, which requires high security of network communication and data transmission. The terminal devices of IoT need mutual identification and identity authentication to ensure network security and data security. Identity authentication is the first line of defense of IoT security. The existing traditional public key infrastructure (PKI) is cumbersome and computationally intensive and cannot meet the needs of the open and distributed IoT environment with limited resources. In this paper, a two-way identity authentication mechanism of IoT terminal based on Blockchain is designed by using SM9 ID cipher algorithm. The scheme uses device ID as the public key, which simplifies the key distribution management process. In addition, based on the assumptions of computational Diffie-Hellman difficulty, q-Diffie-Hellman inverse problem and bilinear Diffie-Hellman difficulty problem, it can greatly meet the confidentiality and unforgeability, and thus is more in line with the practical application environment of IoT. Through simulation experiments to verify and analyze, and comparing several mainstream authentication methods at present, the identity authentication scheme based on SM9 identity cryptographic algorithm and blockchain technology in the IoTs environment can meet the requirements of time, performance and security.

Keywords: IoTs; access control; identity authentication; Blockchain.

1. Introduction

Internet of Things(IoT) technology is an important part of the fourth industrial technology revolution. With the continuous update of IoTs technology, IoTs devices have shown an explosive growth trend, and a variety of physical devices, intelligent instruments, wireless temperature sensors, intelligent vehicle accessories have been widely used [1]. The wide application of IoTs technology in various fields brings us convenience and innovation, but also faces many security risks. Studying identity authentication and access control technology is an important content to ensure the security of IoTs [2].

[1] Prof., School of Computer Science and Engineering, Jining University, Jining, China, e-mail: zbzx@jnxy.edu.cn

[2] Prof., Institute of Technical Education, Shandong Yankuang Technician Institute, Jining, China, e-mail: 1328457112@qq.com

[3] Prof., School of Computer Science and Engineering, Jining University, Jining, China, e-mail: B201602005@jnxy.edu.cn

Blockchain is essentially a trusted decentralized distributed ledger model, representing a comprehensive application of cryptographic algorithms [3]. The paper is based on the Fabric consortium blockchain, which removes the consensus layer and the cryptocurrency issuance functions from traditional blockchain systems. It utilizes blockchain technology to store metadata during the IoT identity authentication process, thereby achieving the characteristics of data immutability and traceability throughout the authentication process.

The terminal devices of IoT need mutual identification and identity authentication to ensure the device security and data security [4,5]. Identity authentication is generally realized by public key cryptography. At present, the mainstream public key cryptography authentication schemes include Public Key Infrastructure (PKI) and Identity-based Cryptography (IBC).

PKI is a traditional encrypted identity authentication method based on Certificate of Authority (CA), which mainly provides four security services: identity authentication, data integrity, data confidentiality and non-repudiation [6]. However, PKI/CA system is difficult to support the identity authentication of intelligent terminal users with limited resources in the IoT scene, because it is necessary to distribute a digital certificate for each IoT device and use public key and private key in digital certificate to complete signing and verification work. The process of key distribution, management and maintenance, and key exchange is very complicated and cumbersome, and the system overhead is large, which can not meet the lightweight identity authentication requirements of a large number of IoT terminal users [7,8].

IBC cryptographic algorithms was first proposed by A. Shamir in 1984 [9]. It is essentially an identity-based public key encryption algorithm and can use attribute value string as an effective public key. In the IBC system, users use their own unique identity such as ID number, phone number and email address as public keys, and the private key is generated by the master key and user identity through the Key Generation Center (KGC). Compared with the traditional CA PKI key system, it simplifies the process of generating, distributing, managing and maintaining public keys. SM9 is a kind of IBC cryptography, and it is a national commercial cipher standard algorithm officially promulgated by China National Cryptography Administration [10].

The above traditional authentication mechanism requires multiple interactions between users and the Certificate of Authority, which is an obstacle for massive IoT terminals. As a distributed system, Blockchain can maintain the same information without the need for central authorization, which can effectively reduce the communication overhead between IoT terminals, gateways, and IBC (CA). This paper designs a two-way identity authentication method for IoTs terminals based on Fabric using SM9 identification cryptography algorithm. In summary, the SM9 algorithm has evident advantages in IoT applications, simplifying key management by using identities as public keys, reducing communication overhead, enhancing security, and providing high adaptability and standardization support.

The remainder of this paper is organized as follows. The related research is introduced in the second chapter. The third chapter introduces the model architecture and authentication process of IoT terminal identity authentication based on Blockchain in detail. Chapter four analyzes the model described in this paper by formal and informal methods. The fifth chapter is a conclusive argument.

2. Related research

2.1 Research on identity authentication in IoT

Extensive research has been conducted by scholars on IoT identity authentication technologies. Xie et al. proposed an ID-based aggregated digital signature algorithm against alliance attack for WSN IoT network [11]. Yan et al. introduce a fine-grained access control scheme tailored for smart homes based on identity authentication, which can effectively prevent unauthorized functions from being maliciously executed [12]. Valmiki et al. proposed a lightweight authentication protocol that utilizes implicit certificates to enable mutual authentication and key negotiation between IoT devices [13].

The research involved in this paper is based on the consortium chain of Blockchain. The consortium chain only selects some nodes for consensus negotiation, which has the following characteristics: first, for the consistency determination of the consortium chain, a group of selected nodes is responsible for verifying the block; second, the read permission of the record can be public It can also be limited; finally, it is not a completely distributed architecture, not all Blockchain nodes can join the consensus process of the consortium chain, and the consensus efficiency is high [14]. With the popularity of Blockchain technology, many scholars have combined Blockchain, IBC identity-based cryptographic, access control and IoT technology to solve the access control problem for IoTs. Cui et al. proposed a multi-sensor network authentication scheme based on hybrid Blockchain, which solved the problem of authentication between IoT devices [15]. Guntur et al. designed a distributed AC mechanism based on attributes, which can realize dynamic and flexible access control by scoring the trust and reputation of nodes in the IoT and incorporating their quantified values into the access control strategy [16]. Gu et al. proposed a cross-domain identity authentication schema of identity password based on Fabric, solving the instant revocation problem of identity authentication based on IBC in consortium Blockchains, and introducing a secure arbitration node to realize user identity management [17].

2.2 SM9 Key derivation function KDF

The key derivation function is responsible for generating key data from a shared secret bit string. In the key negotiation process, it operates on the secret bit string produced by the key exchange to generate the necessary session keys or key data for subsequent encryption tasks. This process requires invoking a cryptographic Hash function.

Let the cryptographic hash function be $H_v(\,)$, and its output is a hash value of length exactly$v$ (in bits).

The key derivation function is$KDF(Z, klen)$.

Input: Bit string Z (data shared by both parties), Integer $klen$ (Indicates the bit length of the obtained key data, the value is required to be less than$(2^{32}-1)v$);

Output: Key data bit string K of length$klen$.

Step 1: Initialize a 32-bit counter $ct = 0x00000001$.

Step 2: Execute for i from 1 to $\left\lceil \frac{klen}{v} \right\rceil$:

Step2.1: Calculate $Ha_i = H_v(Z \parallel ct)$.

Step2.2: $ct++$.

Setp 3: If $\frac{klen}{v}$is an integer, let$Ha!_{\left\lceil \frac{klen}{v} \right\rceil} = Ha_{\left\lceil \frac{klen}{v} \right\rceil}$, Otherwise let $Ha!_{\left\lceil \frac{klen}{v} \right\rceil}$ be the leftmost$\left(klen - \left(v \times \left\lfloor \frac{klen}{v} \right\rfloor\right)\right)$ bits of $Ha_{\left\lceil \frac{klen}{v} \right\rceil}$.

Setp 4: Let $K = Ha_1 \parallel Ha_2 \cdots \parallel Ha_{\left\lceil \frac{klen}{v} \right\rceil_{-1}} \parallel Ha!_{\left\lceil \frac{klen}{v} \right\rceil}$.

2.3 SM9 Identity-based cryptography (IBC) algorithm

SM9 identity-based cryptographic algorithms [18] mainly utilizes the characteristics of ECC point group operation and bilinear pairings operation based on elliptic curves over finite field groups. Elliptic curve pairings with bilinear properties establish the association between cyclic subgroups and multiplicative cyclic subgroups of extended fields, thus forming problems such as bilinear DH (Diffie_Hellman), bilinear inverse DH, deterministic bilinear inverse DH and τ-Gap bilinear inverse to construct a safe and efficient identity-based cryptographic [19].

SM9 algorithm includes: digital signature scheme, key agreement protocol, key encapsulation method, and public key cryptographic algorithm. The paper mainly involves the key exchange algorithm protocol of SM9. In order to establish a trusted channel, the two communicating parties can first obtain a temporary session key shared by both parties through calculation (a symmetric cipher can be selected). This key exchange process of the sender and receiver is shown in Fig 1.

The system selects the master key as $s \in [1, N-1]$, and uses it as the global key. The master public key $P_{pub} = [s]P_1$ is obtained by calculation (belongs to the elements in G_1),So as to get the master key pair (s, P_{pub}), Key generation center of the identity-based cryptographic (KGC) exposes master key P_{pub}, and secretly saves the master private keys$s$. The public key involving sender user A is $Q_A = [H_1(ID_A \parallel hid)]P_2 + P_{pub}$,and private key of the receiver user B is $d_B =$

$\left[\frac{s}{H_1(ID_B \| hid)+s}\right] P_1$, private key of the user A corresponding to it is $d_A = [t_2]P_1 = \left[\frac{s}{H_1(ID_A \| hid)+s}\right] P_1$, the user B public key is $Q_B = [H_1(ID_B \parallel hid)]P_2 + P_{pub}$.

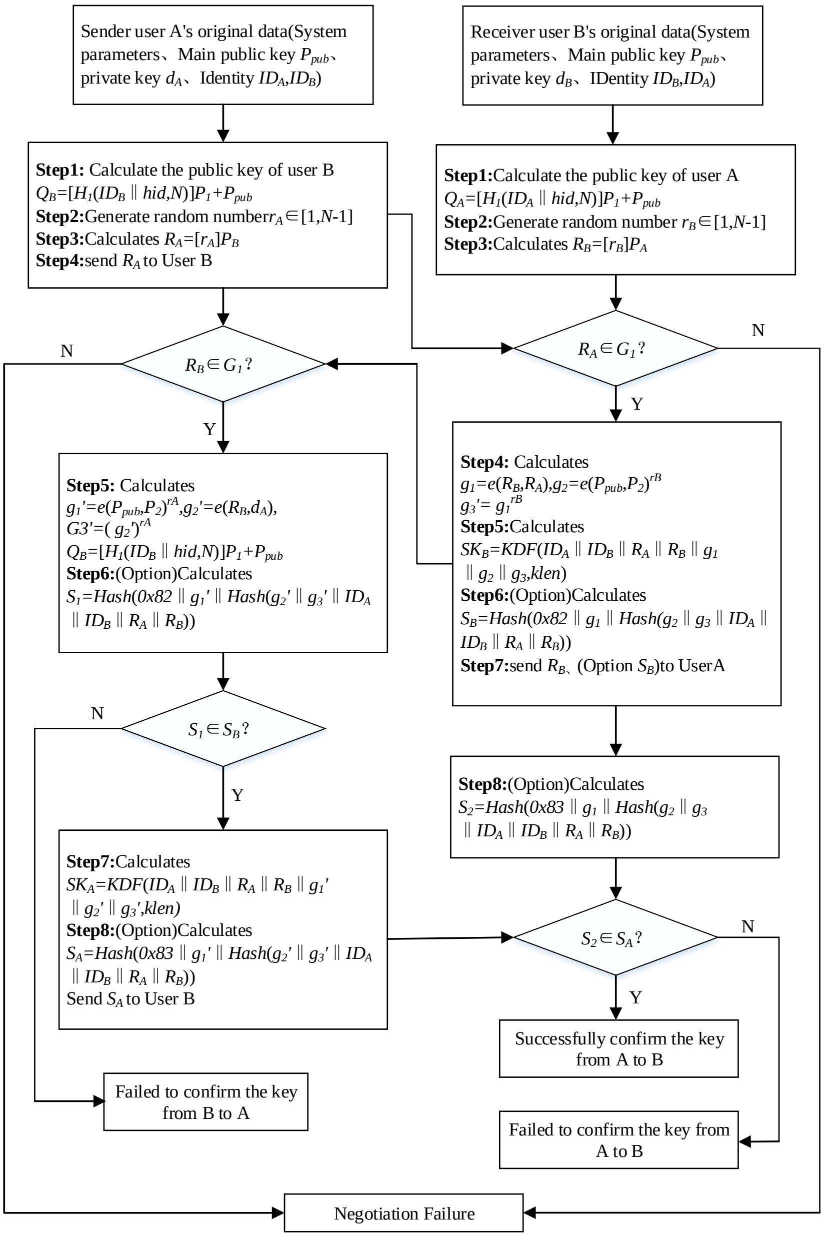

Sender user A's original data(System parameters、Main public key P_{pub}、private key d_A、Identity ID_A,ID_B)
Receiver user B's original data(System parameters、Main public key P_{pub}、private key d_B、IDentity ID_B,ID_A)
Step1: Calculate the public key of user B Q_B=[H_1(ID_B ‖ hid,N)]P_1+P_{pub}
Step2:Generate random numberr_A∈[1,N-1]
Step3:Calculates R_A=[r_A]P_B
Step4:send R_A to User B
Step1:Calculate the public key of user A Q_A=[H_1(ID_A ‖ hid,N)]P_1+P_{pub}
Step2:Generate random number r_B∈[1,N-1]
Step3:Calculates R_B=[r_B]P_A
R_B∈G_1?
R_A∈G_1?
N
Y
Step5: Calculates $g_1'=e(P_{pub},P_2)^{rA}$,$g_2'=e(R_B,d_A)$, $G3'=(g_2')^{rA}$
Q_B=[H_1(ID_B ‖ hid,N)]P_1+P_{pub}
Step6:(Option)Calculates S_1=$Hash$(0x82 ‖ g_1' ‖ $Hash$(g_2' ‖ g_3' ‖ ID_A ‖ ID_B ‖ R_A ‖ R_B))
Step4: Calculates $g_1=e(R_B,R_A)$,$g_2=e(P_{pub},P_2)^{rB}$ $g_3'=g_1^{rB}$
Step5:Calculates SK_B=KDF(ID_A ‖ ID_B ‖ R_A ‖ R_B ‖ g_1 ‖ g_2 ‖ g_3,$klen$)
Step6:(Option)Calculates S_B=$Hash$(0x82 ‖ g_1 ‖ $Hash$(g_2 ‖ g_3 ‖ ID_A ‖ ID_B ‖ R_A ‖ R_B))
Step7:send R_B、(Option S_B)to UserA
S_1∈S_B?
Step8:(Option)Calculates S_2=$Hash$(0x83 ‖ g_1 ‖ $Hash$(g_2 ‖ g_3 ‖ ID_A ‖ ID_B ‖ R_A ‖ R_B))
Step7:Calculates SK_A=KDF(ID_A ‖ ID_B ‖ R_A ‖ R_B ‖ g_1' ‖ g_2' ‖ g_3',$klen$)
Step8:(Option)Calculates S_A=$Hash$(0x83 ‖ g_1' ‖ $Hash$(g_2' ‖ g_3' ‖ ID_A ‖ ID_B ‖ R_A ‖ R_B))
Send S_A to User B
S_2∈S_A?
Successfully confirm the key from A to B
Failed to confirm the key from B to A
Failed to confirm the key from A to B
Negotiation Failure

Fig. 1. SM9 key exchange flowchart

3. IoT terminal identity authentication model based on Fabric

IoT terminal devices often operate in open and diverse network environments. To enhance the security of data during collection, transmission, and storage within the IoT, it is an essential measure to implement efficient and secure identity authentication, which is a powerful guarantee for the implementation of access control mechanism [8]. The networking mode of IoT involved in this paper is WSN (Wireless Sensor Network). We mainly study the identity authentication and key negotiation among the terminal devices in the perception layer of IoT based on WSN.

3.1 Identity authentication process based on IoT

The identity authentication model is composed of three phases: the initialization phase, the registration phase, and the mutual authentication phase.

First, defining the symbols used in the model, as shown in Table 1.

Table 1

Symbol definitions used in this scheme

Symbol	Meaning represented	Symbol	Meaning represented
N	Orders of G_1, G_2 and G_T , which are all prime numbers.	ID_A	Identification of terminal A
P_1	Generator of G_1	ID_B	Identification of terminal A
P_2	Generator of G_2	Q_A	Public key of terminal A
KGC	Key generation center of the identity-based cryptographic authentication authority	d_A	Private key of terminal A
KDF	Key derivation function	Q_B	Public key of terminal B
K	String generated by key derivation function	d_B	Private key of terminal B
CA	identity-based cryptographic authentication authority	r_A, r_B, r_1, r_2	Random number
hid	Private key generation function identity	TS	Time stamp
ID_S	Identity of the identity-based cryptographic authentication authority server	LT	Life time
P_{pub}	Master public key of the identity-based cryptographic authentication authority server	$Enc(k_1, m)$	Symmetric encryption algorithm of m with key k_1
s	Master private key of the identity-based cryptographic authentication authority server	$Dec(k_1, m)$	Symmetric decryption algorithm of m with key k_1

(1) Initialization stage

The server of KGC selects three cyclic groups $(G_1,+)$, $(G_2,+)$ and $(G_T,)$, where the orders of G_1, G_2 and G_T are all prime number N. P_1 is the generator of G_1, and P_2 is the generator of G_2. There is a homomorphic mapping ψ from G_2 to G_1, so that $\psi(P_2) = P_1$; Bilinear pair e is the mapping of $G_1 \times G_2 \to G_T$.

The KGC of the ID authentication authority generates a random number $s \in [1, N-1]$ as the master private key, and then calculates the element $P_{pub} = [s]P_1$ in G_2 as the master public key. The master key pair is (s, P_{pub}). KGC secretly saves s , and made P_{pub} public. The server of the identity-based cryptographic authentication authority selects one of its own identity ID_S and makes it public.

$k_{1_{len}}$ $k_{3_{len}}$ is the bit string length of key k_1 k_3 in symmetric cryptographic algorithm used in encryption and decryption, and $k_{2_{len}}$ $k_{4_{len}}$is the bit string length of key k_2 in function $MAC()$.

(2) Registration stage

The steps of terminal registration are shown in Fig 2. Take terminal A as an example.

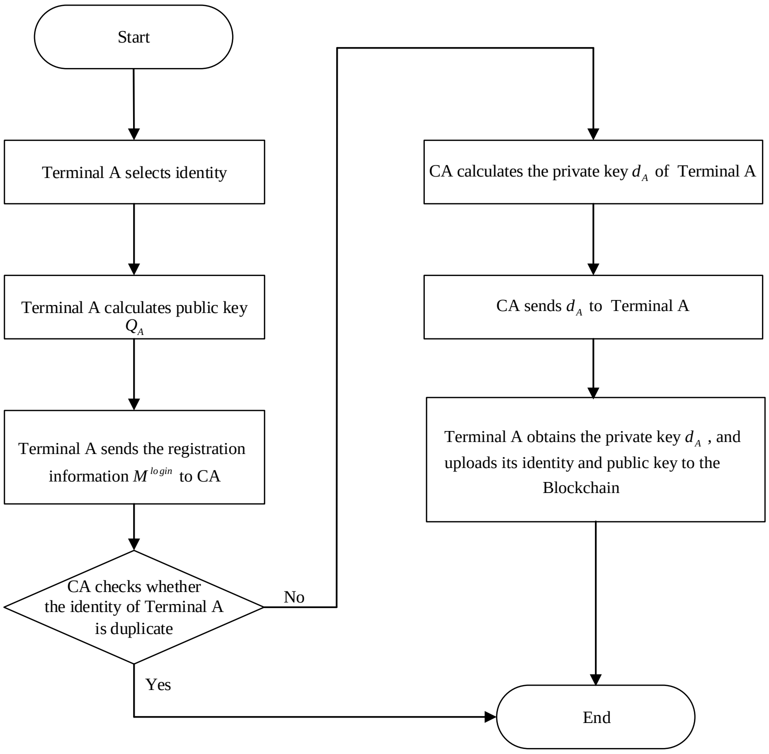

Start
Terminal A selects identity
Terminal A calculates public key Q_A
Terminal A sends the registration information M^{login} to CA
CA checks whether the identity of Terminal A is duplicate
No
Yes
CA calculates the private key d_A of Terminal A
CA sends d_A to Terminal A
Terminal A obtains the private key d_A , and uploads its identity and public key to the Blockchain
End

Fig. 2. Sensor authentication registration process

Step 1: Terminal A independently selects its own identity ID_A. Calculate to obtain the public key of terminal A.

$$Q_A = [H_1(ID_A \parallel hid)]P_1 + P_{pub} \quad (1)$$

Step 2: Terminal A composes registration information ${M_1}^{login} = \{ID_A, Q_A, TS, LT\}$ and sends it to the KGC through encrypted network;

Step 3: After receiving the registration request information ${M_1}^{login}$ of terminal A, KGC checks whether ID_A of the user has been registered. If it has been registered, it will refuse to register again. If it has never been registered, then the key generation center KGC will calculate private key of terminal A according to the identity ID_A and public key Q_A of terminal A.

$$d_A = [H_1(ID_A \parallel hid) + s] \cdot P_2 \tag{2}$$

Then, the message ${M_2}^{login} = \{d_A, TS, LT\}$ is sent to the terminal A through the secure channel;

Step 4: After receiving ${M_2}^{login}$, terminal A uploads the identity ID_A and public key Q_A to the Blockchain.

The registration process of terminal B is the same as above.

(3) Authentication stage

Terminals A and B need to complete the following steps for mutual authentication, as shown in Fig 3.

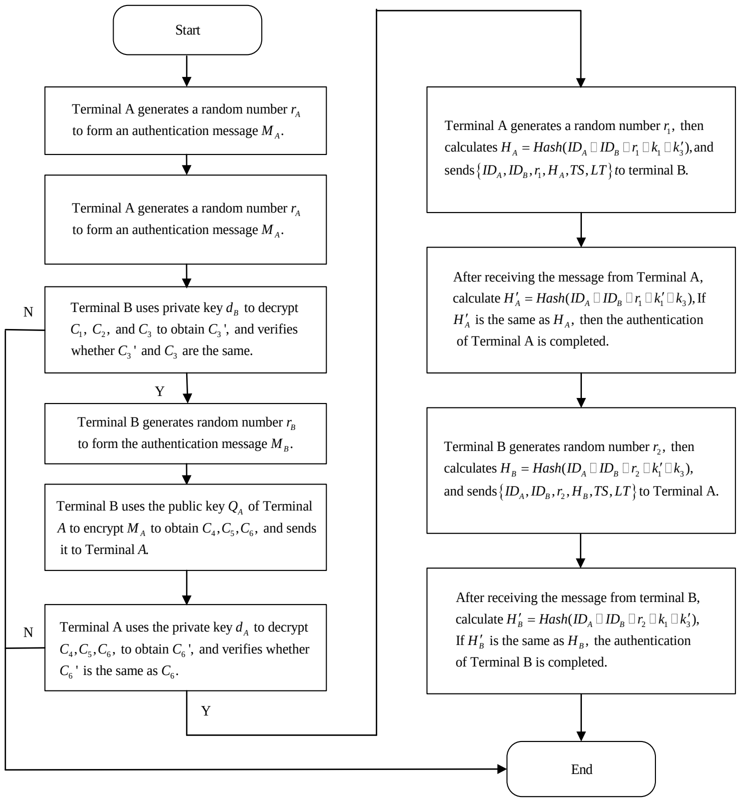

Start
Terminal A generates a random number r_A to form an authentication message M_A.
Terminal A generates a random number r_A to form an authentication message M_A.
N
Terminal B uses private key d_B to decrypt C_1, C_2, and C_3 to obtain C_3', and verifies whether C_3' and C_3 are the same.
Y
Terminal B generates random number r_B to form the authentication message M_B.
Terminal B uses the public key Q_A of Terminal A to encrypt M_A to obtain C_4, C_5, C_6, and sends it to Terminal A.
N
Terminal A uses the private key d_A to decrypt C_4, C_5, C_6, to obtain C_6', and verifies whether C_6' is the same as C_6.
Y
Terminal A generates a random number r_1, then calculates $H_A = Hash(ID_A □ ID_B □ r_1 □ k_1 □ k_3')$, and sends $\{ID_A, ID_B, r_1, H_A, TS, LT\}$ to terminal B.
After receiving the message from Terminal A, calculate $H_A' = Hash(ID_A □ ID_B □ r_1 □ k_1' □ k_3)$, If H_A' is the same as H_A, then the authentication of Terminal A is completed.
Terminal B generates random number r_2, then calculates $H_B = Hash(ID_A □ ID_B □ r_2 □ k_1' □ k_3)$, and sends $\{ID_A, ID_B, r_2, H_B, TS, LT\}$ to Terminal A.
After receiving the message from terminal B, calculate $H_B' = Hash(ID_A □ ID_B □ r_2 □ k_1 □ k_3')$, If H_B' is the same as H_B, the authentication of Terminal B is completed.
End

Fig. 3. Authentication phase flow

Step 1: Terminal A generates a random number r_A $r_A \in [1, N-1]$ to form an authentication application message $M_A = \{ID_A, Q_A, ID_B\}$. Search for the identity ID_B and public key Q_B of terminal B in the Fabric, and use the PK of terminal B for encryption operation.

$$C_1 = [r_A]Q_B \tag{3}$$

$$g = e(P_{pub}, P_2) \tag{4}$$

$$w_1 = g^{r_A} \tag{5}$$

$$K_A = KDF(C_1 \parallel w_1 \parallel ID_B, klen) \tag{6}$$

$$klen = k1_{len} + k2_{len} \tag{7}$$

Wherein, let k_1 be the leftmost $k1_{len}$ bit string of K_A and k_2 be the remaining $k2_{len}$ bit string (note whether k_1 is all 0 or not), and then perform encryption calculation (choose symmetric cryptographic algorithm).

$$C_2 = Enc(k_1, M_A) \tag{8}$$

Calculate the message authentication code (select Hash function)

$$C_3 = MAC(k_2, C_2) \tag{9}$$

Get ciphertext $C_A = \{C_1 \parallel C_2 \parallel C_3\}$, and send message $\{ID_A, C_A, TS, LT\}$ to terminal B through public network;

Step 2: after receiving the message $\{ID_A, C_A, TS, LT\}$, terminal B uses its private key d_B to decrypt the message. First, take C_1 out of C_A and calculate it.

$$w_1^{'} = e(C_1, d_B) \tag{10}$$

$$K_A^{'} = KDF(C_1 \parallel w_1^{'} \parallel ID_B, klen) \tag{11}$$

$$klen = k1_{len} + k2_{len} \tag{12}$$

Wherein, let k_1' be the leftmost $k1_{len}$ bit string of K_A' and k_2' be the remaining $k2_{len}$ bit string (note whether $k_1^{'}$ is all 0 or not), and then perform decryption calculation (symmetric cryptographic algorithm).

$$M_A^{'} = Dec(k_1^{'}, C_2) \tag{13}$$

Then, take C_3 out of C_A, and calculate the message authentication code (select Hash function)

$$C_3^{'} = MAC(k_2^{'}, C_2) \tag{14}$$

Step 3: verify whether C_3' and C_3 are the same. If they are different, report an error and exit. If they are the same, generate a random number ^{r_B} $r_B \in [1, N-1]$ to form an authentication request message $M_B = \{ID_B, Q_B, ID_A\}$. Find the public key Q_A of terminal A in the Blockchain, and use the public key of terminal A to perform encryption operation.

$$C_4 = [r_B]Q_A \tag{15}$$

$$g = e(P_{pub}, P_2) \tag{16}$$

$$w_2 = g^{r_B} \tag{17}$$

$$K_B = KDF(C_4 \parallel w_2 \parallel ID_A, klen) \tag{18}$$

$$klen = k_{3_{len}} + k_{4_{len}} \tag{19}$$

Wherein, let k_3 be the leftmost $k_{3_{len}}$ bit string of K_B and k_4 be the remaining $k_{4_{len}}$ bit string (note whether k_3 is all 0 or not), and then perform encryption calculation (choose symmetric cryptographic algorithm).

$$C_5 = Enc(k_3, M_B) \tag{20}$$

Calculate the message authentication code (select Hash function)

$$C_6 = MAC(k_4, C_5) \tag{21}$$

Get the ciphertext $C_B = \{C_4 \parallel C_5 \parallel C_6\}$;

Step 4: terminal B sends $\{ID_B, C_B, TS, LT\}$ message to terminal A through public network;

Step 5: after receiving the message $\{ID_B, C_B, TS, LT\}$, terminal A uses its own private key d_A to perform decryption operation. First, take C_4 out of C_B, and calculate.

$$w_2^{'} = e(C_4, d_A) \tag{22}$$

$$K_B^{'} = KDF\left(C_4 \parallel w_2^{'} \parallel ID_A, klen\right) \tag{23}$$

$$klen = k_{3_{len}} + k_{4_{len}} \tag{24}$$

Wherein, let k_3' be the left $k_{3_{len}}$ bit string of K_B' and k_4' be the remaining $k_{4_{len}}$ bit string (note whether $k_3^{'}$ is all 0);

Step 6: perform decryption calculation (symmetric cryptographic algorithm)

$$M_B^{'} = Dec\left(k_3^{'}, C_5\right) \tag{25}$$

Step 7: take C_6 out of C_B and calculate the message authentication code (select Hash function)

$$C_6^{'} = MAC\left(k_4^{'}, C_5\right) \tag{26}$$

Step 8: verify whether C_6' and C_6 are the same, if they are different, report an error and exit. If they are the same, generate a new random number $^{r_1}r_1 \in [1, N-1]$, and calculate $H_A = Hash(ID_A \parallel ID_B \parallel r_1 \parallel k_1 \parallel k_3')$. Send $\{ID_A, ID_B, r_1, H_A, TS, LT\}$ to terminal B through public network;

Step 9: after receiving the message $\{ID_A, ID_B, r_1, H_A, TS, LT\}$, terminal B calculates $H_A' = Hash(ID_A \parallel ID_B \parallel r_1 \parallel k_1' \parallel k_3)$. If $H_A^{'} = H_A$, the authentication of terminal A by terminal B is completed;

Step 10: terminal B generates a new random number $r_2 \in [1, N-1]$. Calculate $H_B = Hash(ID_A \parallel ID_B \parallel r_2 \parallel k_1' \parallel k_3)$, and send $\{ID_A, ID_B, r_2, H_B, TS, LT\}$ to Terminal A through the public network;

Step 11: after receiving the message $\{ID_A, ID_B, r_2, H_B, TS, LT\}$, terminal A calculates $H_B' = Hash(ID_A \parallel ID_B \parallel r_2 \parallel k_1 \parallel k_3')$. If $H_B^{'} = H_B$, the authentication of terminal B by terminal A is completed. So far, the mutual authentication between terminal A and terminal B has been successfully completed.

In this model, the Blockchain is used in both the registration stage and the authentication stage. The tamper-proof feature of the Blockchain ensures that the related information of the terminal cannot be tampered, which is equivalent to the second confirmation of the authentication information in the key link of the authentication process. In this way, the security of identity authentication is improved. In this model, after mutual authentication is completed in the authentication stage, $K_{AB} = Hash(ID_A, \ ID_B, k_1, k_3) k_1 = k_1', k_3 = k_3'$ can be further calculated to obtain the shared temporary session key between terminal A and B, which can be used as the encryption and decryption key for the next step of mutual information between the two parties. Because k_1, k_3 are randomly generated, the shared key K_{AB} is also random and one-time. In conclusion, the shared key has higher security.

3.2 Cross domain identity authentication

This model is aimed at the mutual authentication of two terminals under the jurisdiction of the same identity-based cryptographic authority. If the terminals to be authenticated are under the jurisdiction of different identity-based cryptographic authorities (for example, terminal A is under the jurisdiction of and terminal B is under the jurisdiction of), both parties may negotiate an identity-based cryptographic authority recognized by both parties. Identity authentication can be performed after registration, as shown in Fig 4.

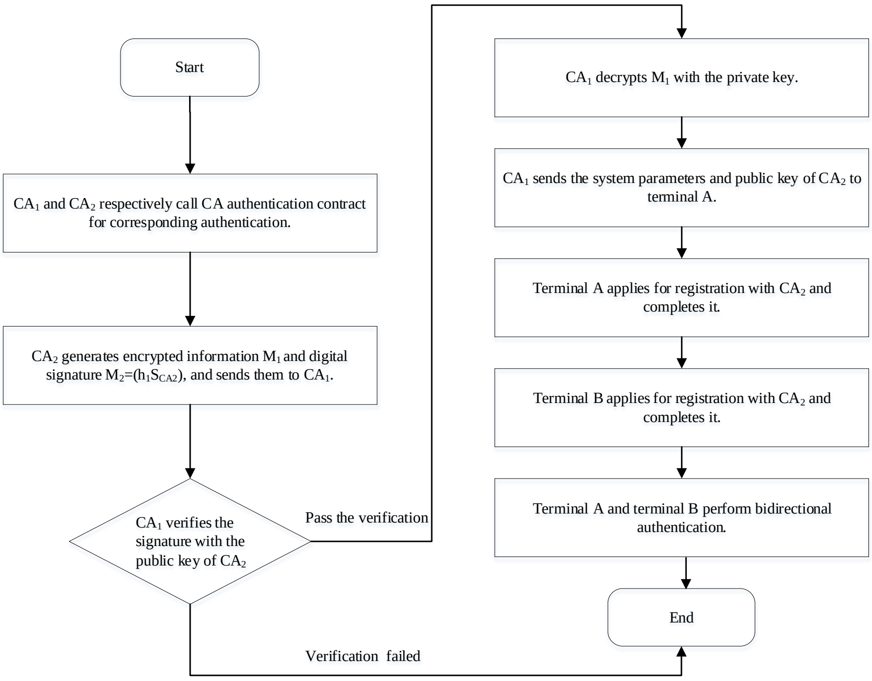

Start
CA₁ and CA₂ respectively call CA authentication contract for corresponding authentication.
CA₂ generates encrypted information M₁ and digital signature M₂=(h₁S_CA2), and sends them to CA₁.
CA₁ verifies the signature with the public key of CA₂
Pass the verification
Verification failed
CA₁ decrypts M₁ with the private key.
CA₁ sends the system parameters and public key of CA₂ to terminal A.
Terminal A applies for registration with CA₂ and completes it.
Terminal B applies for registration with CA₂ and completes it.
Terminal A and terminal B perform bidirectional authentication.
End

Fig. 4. Cross-domain authentication process

The detailed steps are as outlined below:

Step 1: CA_1 and CA_2 can mutually authenticate by calling CA authentication contract of Blockchain. After passing the authentication, by default, the receiver (e.g.CA_2) of the first communication uses the public key of the sender (e.g. CA_1) to encrypt its own system parameters (cyclic group, group order, generator, etc.) and public key. After obtaining the encrypted information M_1, get the digital signature $M_2 = (h, S_{CA2})$ of M_1 through calculation, and send M_1 and M_2 to CA_1;

Step 2: CA_1 verifies the digital signature with public key of CA_2 after receiving M_1 , M_2 . If the verification fails, report an error and exit. If the verification passes, CA_1 decrypts M_1 with the private key, and then sends the decrypted data such as system parameters and public key of CA_2 to terminal A;

Step 3: Terminal A applies to CA_2 for registration and completes it;

Step 4: Terminal B applies to CA_2 for registration and completes it;

Step 5: Two-way authentication between terminal A and B;

Step 6: the authentication is completed and the process is over.

4. Analysis of identity authentication model

4.1 BAN logic Formal analysis

We employ BAN logic to conduct a formal analysis of the method proposed in the paper. BAN logic is a belief-based modal logic that primarily deals with the following three elements: subject, key and formula. According to convention, P, Q, R represent subject variable; K represents key variable; X and Y represent formula variable; A, B represent two ordinary subjects; S is authentication server, K_{ab}, K_{ac}, K_{bc} and so on represent specific shared key; K_a, K_b, K_c and so on represent specific public key; $K_a^{-1}, K_b^{-1}, K_c^{-1}$ and so on represent specific secret key; N_a, N_b, N_c and so on represent temporary value, and $h(X)$ represents one-way hash function of X.

The logic proof process of BAN is as follows:

(1) Message idealization

Between terminal A and KGC:

$$M_1 : CA \lhd ID_A; CA \lhd Q_A \tag{27}$$

$$M_2 : ID_A \lhd d_A \tag{28}$$

Between terminal B and KGC:

$$M_3 : CA \lhd ID_B; CA \lhd Q_B \tag{29}$$

$$M_4 : ID_B \lhd d_B \tag{30}$$

Between terminal A and terminal B:

$$M_5 : ID_B \lhd ID_A; ID_B \lhd Q_A; ID_B \lhd C_A; (C_A = \{C_1 \parallel C_2 \parallel C_3\}) \tag{31}$$

$$M_6 : ID_A \lhd ID_B; ID_A \lhd Q_B; ID_A \lhd C_B; (C_B = \{C_4 \parallel C_5 \parallel C_6\}) \tag{32}$$

(2) Establishment of initial assumptions

Between terminal A and KGC, for KGC:

$$A1: CA|\equiv ID_A \overset{d_A}{\rightleftharpoons} CA; \tag{33}$$

$$A2: CA|\equiv ID_A|\sim {M_1}^{login}; \tag{34}$$

$$A3: CA|\equiv \#(TS); \tag{35}$$

$$A4: CA|\equiv \#(LT); \tag{36}$$

$$A5: \xrightarrow{{d_{CA}}^{-1}} CA; \tag{37}$$

$$A6: CA|\equiv (ID_A|\Rightarrow ID_A); \tag{38}$$

Between terminal A and KGC, for terminal A:

$$A7: ID_A|\equiv CA|\sim {M_2}^{login}; \tag{39}$$

$$A8: ID_A|\equiv \#(TS); \tag{40}$$

$$A9: ID_A|\equiv \#(LT); \tag{41}$$

$$A10: \xrightarrow{{d_A}^{-1}} ID_A; \tag{42}$$

$$A11: ID_A|\equiv CA \overset{d_A}{\rightleftharpoons} ID_A; \tag{43}$$

Between terminal A and terminal B, for terminal A:

$$A12: ID_A|\equiv CA \overset{k_1,k_2,k_3,k_4}{\rightleftharpoons} ID_A; \tag{44}$$

$$A13: ID_A|\equiv \#(r_1); \tag{45}$$

Between terminal A and terminal B, for terminal A:

$$A14: CA|\equiv ID_A \overset{k_1,k_2,k_3,k_4}{\rightleftharpoons} CA; \tag{46}$$

$$A15: CA|\equiv \#(r_2); \tag{47}$$

(3) Establishment of security objectives

$$G1: ID_B|\equiv ID_A|\equiv (k_1,k_2); \tag{48}$$

$$G2: ID_A|\equiv ID_B|\equiv (k_3,k_4); \tag{49}$$

$$G3: ID_B|\equiv (k_1,k_2); \tag{50}$$

$$G4: ID_A|\equiv (k_3,k_4); \tag{51}$$

$$G5: ID_B|\equiv ID_A|\equiv (k_1,k_3); \tag{52}$$

$$G6: ID_A|\equiv ID_B|\equiv (k_1,k_3); \tag{53}$$

(4) Formalization of authentication model

$$\text{F1. } ID_A|\equiv Q_A, ID_A|\equiv d_A, ID_A|\Rightarrow Q_A, ID_A|\Rightarrow d_A \tag{54}$$

$$\text{F2. } ID_B|\equiv Q_B, ID_B|\equiv d_B, ID_B|\Rightarrow Q_B, ID_B|\Rightarrow d_B \tag{55}$$

$$\text{F3. } ID_A|\equiv (k_1,k_2), ID_A|\Rightarrow (k_1,k_2) \tag{56}$$

$$\text{F4. } ID_B|\equiv (k_3,k_4), ID_B|\Rightarrow (k_3,k_4) \tag{57}$$

$$\text{F5. } ID_A|\sim \{C_A\}_{Q_B}, ID_A|\sim \langle C_A\rangle_{(k_1,k_2)}, ID_B|\equiv ID_A|\Rightarrow \{C_A\} \tag{58}$$

$$\text{F6. } ID_B|\sim \{C_B\}_{Q_A}, ID_B|\sim \langle C_B\rangle_{(k_3,k_4)}, ID_A|\equiv ID_B|\Rightarrow \{C_B\} \tag{59}$$

$$\text{F7. } ID_B \lhd \{C_A\}_{Q_B}, ID_A \lhd \{C_B\}_{Q_A} \tag{60}$$

$$\text{F8. } ID_A|\sim \{H_A\}, ID_B|\sim \{H_B\} \tag{61}$$

$$\text{F9. } ID_B|\equiv (k_1,k_2) \tag{62}$$

$$\text{F10. } ID_A|\equiv (k_3,k_4) \tag{63}$$

$$\text{F11. } ID_A|\equiv (r_1), ID_A|\Rightarrow r_1 \tag{64}$$

F12. $ID_B|\equiv(r_2), ID_B|\Rightarrow r_2$ (65)

F13. $ID_A|\equiv \Big|\xrightarrow{Q_B} ID_B$ (66)

F14. $ID_B|\equiv \Big|\xrightarrow{Q_A} ID_A$ (67)

The specific authentication process is as follows:

V1. Terminal B decrypts $\{C_A\}_{(k_1k_2)}$ with its own private key d_B, which can verify the information $\{C_A\}_{(k_1k_2)}$, namely $ID_B|\equiv ID_A\big|\sim \{C_A\}_{(k_1k_2)}$, sent by terminal A; according to F9 and rule R11, $ID_B|\equiv \#(k_1,k_2), \vdash ID_B\big|\equiv \#\{C_A\}_{(k_1,k_2)}$; according to rule R4, $ID_B|\equiv\#\{C_A\}_{(k_1,k_2)}, ID_B|\equiv ID_A\big|\sim \{C_A\}_{(k_1,k_2)} \vdash ID_B|\equiv ID_A|\equiv\{C_A\}_{(k_1,k_2)}$, namely $ID_B|\equiv ID_A|\equiv(k_1,k_2)$. So G1 goal is established;

V2. According to the above G1 proving course, similarly, it can be concluded that the G2 goal is established;

V3. According to F5 and rule R5, $ID_B|\equiv ID_A|\Rightarrow(k_1,k_2), ID_B|\equiv ID_A|\equiv(k_1,k_2) \vdash ID_B|\equiv(k_1,k_2)$ can be obtained. So G3 goal is established;

V4. According to the above G3 proving course, similarly, G4 goal can be established;

V5. According to G3, F4 and rule R5, $ID_B|\equiv(k_1,k_2), ID_B|\equiv(k_3,k_4) \vdash ID_B|\equiv(k_1,k_3)$ can be obtained. So G5 goal is established;

V6. According to the above G5 proving course, similarly, G6 goal can be established;

Through the above-mentioned BAN logic analysis, the identity authentication model proposed in this chapter can attain the intended security objective, demonstrating that the model is secure.

4.2 Proverif simulation experiment verification

In this paper, the ProVerif formal tool is used to analyze the security of the algorithm of this protocol. ProVerif is a cryptographic protocol analysis tool, which is a widely used automatic verification tool for the formal verification of cryptographic protocols, based on the Dolev-Yao model. Developed using the Prolog language, ProVerif can describe most cryptographic primitives, covering shared keys, public keys, hash functions, and the Diffie-Hellman key exchange protocol. Its input language is based on the app's pi-calculus or Horn clause.

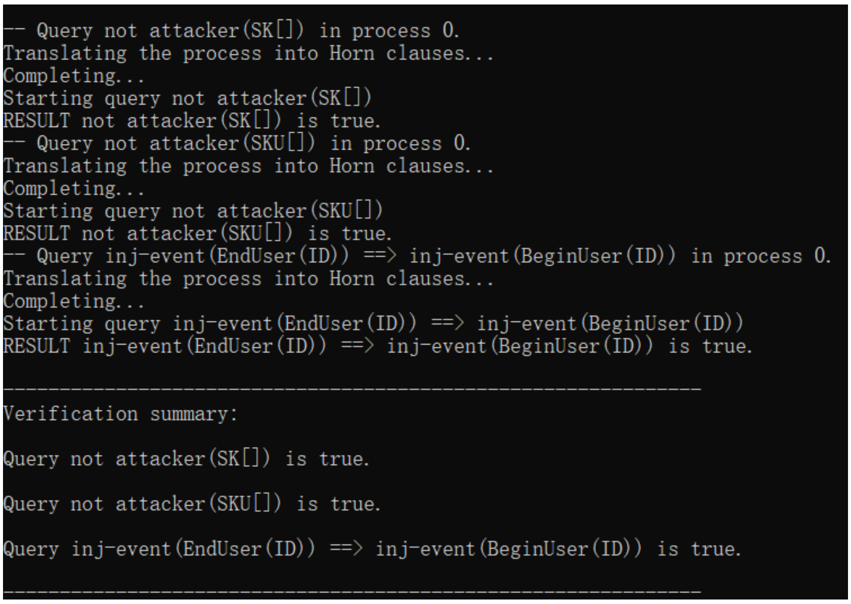
(1) Proverif verification code

```
(*----------------Verification channel---------------------*)
free sch: channel [private].
free ch: channel.
(*----------------variable and constants---------------------*)
const ID: bitstring. (*---User ID----*)
```

```
free R: bitstring[private].
free SR: bitstring[private].
(*----------------constructor---------------------*)
fun fi(bitstring): bitstring.
fun Fi(bitstring): bitstring.
fun senc(bitstring, bitstring): bitstring.
fun REP(bitstring, bitstring, bitstring, bitstring): bitstring.
(*----------------destructors&equations---------------------*)
reduc forall m:bitstring, key:bitstring; sdec(senc(m,key),key)=m.
(*----------------Verification events ---------------------*)
event BeginUser(bitstring).
event EndUser(bitstring).
(*----------------Verification query ---------------------*)
query attacker(R).
query attacker(SR).
query id:bitstring; inj-event(EndUser(ID))==>inj-event(BeginUser(ID)).
(*----------------Verification process---------------------*)
(*----------------user process---------------------*)
let User=
in(ch,(ID:bitstring));
!
(
event BeginUser(ID);
in(sch,(CC:bitstring, C1:bitstring, tbx:bitstring, C2: bitstring, C3: bitstring, C4: bitstring));
let SR=REP(C1, C2, C3, C4) in
if SR=R then
let MSG=sdec(CC,SR) in
event EndUser(ID);
0
).
(*----------------Attack process---------------------*)
let Att=
!
(
in(ch, (AID: bitstring));
0
).
(*----------------CA process---------------------*)
let CA=
new a: bitstring;
new ID: bitstring;
new AID: bitstring;
out(sch, (ID));
out(sch, (AID));
new msg: bitstring;
new C1: bitstring;
new C2: bitstring;
new C3: bitstring;
new C4: bitstring;
let CC=senc(msg, R) in
out(sch, (CC, C1, C2, C3, C4 ));
0.
process ( User| Att |CA)
```

(2) Proverif verification results

The Proverif simulation software verifies the code execution results as shown in Fig. 5 below. The results show that the protocol described in this chapter is safe and reliable, and can meet the user identity authentication requirements in the Internet of Things scenario.



```
-- Query not attacker(SK[]) in process 0.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.
-- Query not attacker(SKU[]) in process 0.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(SKU[])
RESULT not attacker(SKU[]) is true.
-- Query inj-event(EndUser(ID)) ==> inj-event(BeginUser(ID)) in process 0.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(EndUser(ID)) ==> inj-event(BeginUser(ID))
RESULT inj-event(EndUser(ID)) ==> inj-event(BeginUser(ID)) is true.

--------------------------------------------------------------
Verification summary:

Query not attacker(SK[]) is true.

Query not attacker(SKU[]) is true.

Query inj-event(EndUser(ID)) ==> inj-event(BeginUser(ID)) is true.

--------------------------------------------------------------
```

Fig. 5. The ProVerif tool performs the verification results

Subsequently, we constructed the identity authentication model described in this paper during the data access control process in a smart manufacturing factory, implementing identity authentication functions between IoT devices such as IoT gateways, smart cameras, mobile pad terminals, and smart smoke detectors.

4.3 Comparative evaluation of security and performance

According to the above security analysis and the BAN logic proof of this scheme, the comparative assessment of the scheme introduced in the paper and other models is presented in Table 2. Table 2 highlights the security features of this identity authentication scheme proposed in the paper compared to other models.

Table 2

Comparison of model security

Model	Resisting replay attack	Resisting tampering attack	Resisting collusion attack	Resisting Counterfeiting attack	Resisting man-in-the-middle attack	Resisting denial of service attack	Forward security
Literature [4]	No	No	Yes	Yes	No	Yes	Yes
Literature [7]	Yes	Yes	Yes	No	Yes	Yes	No
Literature [20]	Yes	No	No	Yes	Yes	No	Yes
Literature [21]	No	No	No	Yes	No	No	Yes
Literature [22]	Yes	No	No	No	No	No	Yes
Literature [23]	Yes	No	Yes	Yes	Yes	No	Yes
Literature [24]	Yes	Yes	Yes	No	Yes	Yes	No
This article	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Key findings from the analysis include:

(1) Resisting Replay Attack: The proposed model effectively resists replay attacks, unlike models from Literature [4] and [21], which are vulnerable to such attacks.

(2) Resisting Tampering Attack: The proposed model can resist tampering attacks. However, models from Literature [20], [21], [22], and [23] are not able to counter these attacks.

(3) Resisting Collusion Attack: The proposed model is resilient to collusion attacks, a weakness identified in models from Literature [20], [21], and [22].

(4) Resisting Counterfeiting Attack: The proposed model successfully resists counterfeiting attacks, while models from Literature [7], [22], and [24] are vulnerable.

(5) Resisting Man-in-the-Middle Attack: The proposed model can resist man-in-the-middle attacks, contrary to models from Literature [4], [21], and [22].

(6) Resisting Denial of Service Attack: The proposed model effectively resists denial of service attacks. In contrast, models from Literature [20], [21], [22], and [23] have deficiencies in this area.

(7) Forward Security: The proposed model ensures forward security, whereas models from Literature [7] and [24] do not guarantee this feature.

Overall, the proposed identity authentication model demonstrates high security and reliability across all evaluated dimensions, making it superior to the compared models.

5. Conclusion

This study addresses the pressing challenges in identity authentication for IoT systems by proposing a novel authentication model that integrates blockchain technology with the SM9 identity-based cryptographic algorithm. A formal description of the model was provided, and its security capabilities were analyzed comprehensively across seven critical dimensions: resistance to replay attacks, tampering attacks, collusion attacks, counterfeiting attacks, man-in-the-middle attacks, denial-of-service attacks, and forward security. The results of the security analysis demonstrate that the proposed model not only ensures one-time key usage but also offers strong resistance against various attack vectors, making it highly robust in securing IoT environments. Moreover, the model's low system overhead and adaptability to resource-constrained IoT devices highlight its suitability for deployment in open IoT networks. In comparison to existing identity authentication models, the proposed approach shows improved security performance and operational efficiency. Future research will focus on conducting a quantitative performance analysis of the proposed authentication method to further validate its practical applicability. Additionally, we aim to explore cross-domain identity authentication techniques to enhance interoperability across different organizational boundaries. The main contributions of this paper are:

(1) To address high system overhead, low efficiency, and insufficient security in identity authentication for resource-constrained IoT terminal nodes, a distributed method using Blockchain and the SM9 identification encryption algorithm was proposed. This method enhances security, immutability, and traceability of authentication data, and supports two-way identity authentication. It uses user IDs as public keys, simplifying key management and improving efficiency while reducing overhead.

(2) We designed processes for digital signature generation, verification, key exchange, key encapsulation, and decapsulation based on the SM9 identity-based cryptographic algorithm to achieve secure identity authentication for IoT terminals.

(3) The proposed identity authentication model was verified using BAN logic analysis, confirming its ability to meet expected security goals and proving the model’s security.

REFERENCES

[1]. *Qi S, Lu Y, Zheng Y, et al,* Cpds: Enabling Compressed and Private Data Sharing for Industrial Internet of Things Over Blockchain, IEEE Transactions on Industrial Informatics, 17(4), 2021, pp.2376-2387.

[2]. *Sammy F, Vigila S M C,* Anomaly Detection in Cloud Using Hexabullus Optimisation-Enabled Fuzzy Classifier with Smart Contract-Enabled Secure Communication, Journal of Information & Knowledge Management, 23(1),2024, pp.105−118.

[3]. *Khalid M, Hameed S, Qadir A ,et al,* Towards SDN-based smart contract solution for IoT access control, Computer communications, 2023,pp.206−220.

[4]. *Porambage P, Kumar P, Schmitt C, et al,* Certificate-Based Pairwise Key Establishment Protocol for Wireless Sensor Networks, 2013 IEEE 16th International Conference on Computational Science and Engineering,2013, pp. 667-681.

[5]. *Fugkeaw, S., L. Wirz, and L. Hak,* Secure and Lightweight Blockchain-Enabled Access Control for Fog-Assisted IoT Cloud Based Electronic Medical Records Sharing, IEEE Access 11,2023, pp.62998-63012.

[6]. *Ge C, Xia J, Fang L,* Key-Private Identity-Based Proxy Re-Encryption, Cmc-Computers Materials & Continua, 63(2),2020,pp. 633-647.

[7]. *Kumar P, Gurtov A, Iinatti J, et al* ,Lightweight and Secure Session-Key Establishment Scheme in Smart Home Environments, IEEE Sensors Journal, 16(1),2016,pp.254-264.

[8]. *Cheng X, Zhang Z, Chen F, et al* , Secure Identity Authentication of Community Medical IoT, IEEE Access, 7,2019,pp.115966-77.

[9]. *Shamir A,* Identity-based cryptosystems and signature schemes, in Workshop on the Theory and Application of Cryptographic Techniques, 1984,pp. 47–53.

[10]. *Yuan F,Cheng ZH,* Overview on SM9 Identity-Based Cryptographic Algorithm(in Chinese), Journal of Information Security Research. 2(11), 2016, pp.1008-1027.

[11]. *Xie Y, Xu F, Li X, et al,* EIAS: An Efficient Identity-Based Aggregate Signature Scheme for WSNs Against Coalition Attack, Cmc-Computers Materials & Continua, 59(3),2019,pp.903-924.

[12]. *Yan H Y, Wang Y, Jia C F, et al,* IoT-FBAC: Function-based access control scheme using identity-based encryption in IoT, Future Generation Computer Systems-the International Journal of Escience, 95,2019, pp.344-353.

[13]. *Siddhartha V G G S, Kansal L,* A Lightweight Authentication Protocol using Implicit Certificates for Securing IoT Systems, Procedia Computer Science, (167) 9, 2020.

[14]. *Shih D-H, Wu T-W, Shih M-H, et al,* Hyperledger Fabric Access Control for Industrial Internet of Things, Applied Sciences, 12(6), 2022.

[15]. *Cui Z, Xue F, Zhang S, et al,* A Hybrid Blockchain-Based Identity Authentication Scheme for Multi-WSN, IEEE Transactions on Services Computing, 2020,pp.1-16.

[16]. *Putra, G. D., Dedeoglu, V., Kanhere, S. S., Jurdak, R., & Ignjatovic, A,* Trust-Based Blockchain Authorization for IoT, IEEE Transactions on Network and Service Management, 18(2), 2022,pp.1646–1658.

[17]. *Gu, P., & Chen, L,* An Efficient Blockchain-based Cross-domain Authentication and Secure Certificate Revocation Scheme,2020 IEEE 6th International Conference on Computer and Communications (ICCC) ,2020.

[18]. *Lai J C, Huang X Y, He D B, et al,* Security analysis of SM9 digital signature and key encapsulation (in Chinese), SciSin Inform, 51,2021,pp.1900–1913.

[19]. *Peng C, He D B, Luo M, Huang X Y, Li D W,* An identity-based ring signature scheme for SM9 algorithm, Journal of Cryptologic Research, 8(4),2021,pp.724–734.

[20]. *Sciancalepore S, Piro G, Boggia G, et al*, Public Key Authentication and key negotiation in IoT Devices With Minimal Airtime Consumption, IEEE Embedded Systems Letters, 9(1),2017, pp.1-4.

[21]. *Zhang X, Liu C, Chai K K, et al*, A privacy-preserving consensus mechanism for an electric vehicle charging scheme,Journal of Network and Computer Applications, 2021, pp.174-186.

[22]. *Hossain M, Noor S, Hasan R*, HSC-IoT: A Hardware and Software Co-Verification Based Authentication Scheme for IoT, 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) ,2017.

[23]. *Kumar P, Braeken A, Gurtov A, et al*, Anonymous Secure Framework in Connected Smart Home Environments, IEEE Transactions on Information Forensics and Security, 12(4),2017, pp. 968-979.

[24]. *Han K, Kim J, Shon T, et al,*A novel secure key paring protocol for RF4CE ubiquitous smart home systems, Personal and Ubiquitous Computing, 17(5),2012, pp. 945-949.