Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9942950, 14 pages
https://doi.org/10.1155/2021/9942950

Research Article

A Cross-Domain Authentication Optimization Scheme between Heterogeneous IoT Applications

Shichang Xuan, Haibo Xiao, Dapeng Man, Wei Wang, and Wu Yang

Information Security Research Center, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Wei Wang; w_wei@hrbeu.edu.cn

Received 5 March 2021; Revised 9 August 2021; Accepted 15 September 2021; Published 29 September 2021

Academic Editor: Ding Wang

With the continuous enrichment of the Internet of Things (IoT) applications, the demand for value exchange and collaborative control between heterogeneous IoT applications is increasing. However, the user management space varies depending on the IoT application, where the security domain stands as an example. It is one of the key technologies of data sharing between heterogeneous IoT organizations to cross the boundary of the security domain and verify the identity and authority of users in other security domains. Aiming at the slow speed of authentication protocol authority authentication during cross-domain access and without considering the actual cross-domain situation, the same cryptographic system parameters are used for all communication nodes in a cross-domain environment. This article proposes a heterogeneous Internet of Things data access authority authentication scheme between applications. Based on certificate-less public key cryptography and smart contract technology, a certificate-less cross-domain authentication scheme that supports parameter differentiation is designed and implemented. The theoretical and empirical analyses, comparing the communication volume, identity signature, and verification calculation cost, validated that the method proposed improves the cross-domain identity authorization authentication ability and supports the use of differentiated cryptographic system parameters among different IoT applications.

1. Introduction

The increasing popularity of the Internet of Things (IoT) has resulted in the growth of connected devices, such as sensors and smart devices, at an alarming rate, which have become an integral part of daily human life [1–5]. The Internet data center predicts that by 2025, there will be more than 40 billion IoT devices connected to the Internet [6]. Although the Internet has been deployed on a large scale in many Internet of Things application scenarios such as smart cities, industrial IoT, and vehicle Internet, it is more vulnerable due to limited resources. The traditional client-server architecture mostly relies on a centralized cloud architecture, which means that massive amounts of IoT data will be transmitted to a centralized cloud server via the Internet. The centralized IoT communication model faces the era of the explosive growth of big data and brings many drawbacks, such as high-latency response, lack of security, and a large amount of workload. Moreover, these peer-to-peer networks and cloud environment-based traditional centralized IoT data sharing solutions cannot prevent the single points of failure and the attacks targeting the centralized storage. The traditional IoT architecture has many major limitations that cannot meet the security requirements of IoT, such as relying on trusted servers, powerless to time-sensitive applications, and high data maintenance costs [7, 8].

The booming blockchain technology is considered highly promising to tighten the security in the IoT. The blockchain can be used to establish a distributed trust mechanism improving the efficiency of IoT communication, thereby solving the security and accuracy problems in the abovementioned centralized IoT architecture. The blockchain is essentially a distributed ledger distributed throughout the distributed system [9]. It is a data structure shared by all nodes and cannot be tampered with. Anyone can upload and access the data, and everyone needs to be

responsible for their data. Therefore, the blockchain can conduct transactions in a mutually distrusted distributed system. Unlike the existing transaction management system where the central agency needs to verify transactions, the blockchain can realize decentralized verification of transactions, thereby greatly saving costs and alleviating the performance bottleneck of the central agency. Besides, every transaction stored in the blockchain is inherently immutable, because every node in the network stores all submitted transactions in the blockchain. At the same time, encryption mechanisms (such as asymmetric encryption algorithms, digital signatures, and hash functions) ensure the integrity of blockchain data blocks [10]. Therefore, the blockchain can ensure the nonrepudiation of transactions. First, the blockchain verifies the data in chronological order and then counts it into unchangeable blocks. Furthermore, it maintains the data consistency of each node through the node consensus, which does not require any trusted intermediary. These features are distributed IoT security. Sexual expansion provides new solutions [11–14].

With the continuous enrichment of IoT applications, there will inevitably be a demand for valuable exchange and collaborative control between different IoT applications, which cannot avoid the problem of cross-domain identity authorization authentication. Cross-domain authentication is one of the key technologies for secure IoT communications as user space and autonomy vary from application to application. However, traditional centralized solutions have problems such as single node failure, key escrow, and man-in-the-middle attacks, which are not suitable for IoT terminals [15]. Blockchain technology with the characteristics of a decentralized, fully distributed P2P network, transaction transparency, nontampering, and encryption algorithms to ensure security is considered an effective means to achieve decentralized authentication [16–18]. The researchers combined blockchain with cross-domain authentication technology to address the issues related to cross-domain user identity authentication and establishing trust between entities in a distributed IoT environment. However, most cross-domain authentication protocols adopt complex bilinear pairing operations, and the authentication efficiency is low, and the actual cross-domain situation is not considered. Using the same cryptographic system parameters between all communication nodes in the cross-domain environment, to guarantee the private key safety, a secure channel is required for transmission, which leads to problems such as reduced security. This work introduces a Certificate-less Cross-domain Authentication Scheme with Different System Parameters (DSP-CCAS) that supports parameter differentiation based on certificate-less passwords. By improving the certificate-less authentication algorithm, the authentication efficiency and security in the cross-domain authentication process of a variety of IoT applications are improved. The contributions of the proposed authentication scheme, DSP-CCAS, can be listed as follows:

(1) A cross-domain access control authentication method based on certificate-less passwords supporting parameter differentiation is proposed

(2) A specific implementation plan for cross-domain authentication was proposed, and its effectiveness was evaluated through a prototype system

The remainder of this paper is organized as follows. Section 2 highlights the state-of-the-art research in the area, Section 3 introduces the certificate-less signature algorithm for authentication, Section 4 describes the proposed DSP-CCAS in detail, Section 5 presents the performance evaluation results, and Section 6 finalizes the paper.

2. Related Works

Blockchain was first mentioned by Nakamoto in "Bitcoin: A peer-to-peer electronic cash system" [19] in 2008. Generally, a blockchain is defined as a specific data structure formed by combining data blocks in a chain in a chronological order and cryptographically ensuring that it cannot be tampered with and is unforgeable decentralized, trustless distributed sharing general ledger system. In view of the unique combination of attributes of blockchain, many fields have listed it as the primary development direction, such as financial technology [20], cross-border e-commerce [21], data sharing [22], and other fields. The blockchain also provides the PKI system with the transparency, revocability, and reliable transaction records of the certificate and eliminates the security attributes of the center failure node. At this stage, blockchain-based cross-domain authentication schemes can be divided into two categories, namely, the authentication model for deploying the PKI system on the blockchain and the cross-domain authentication scheme by building an interdomain consortium blockchain model.

2.1. PKI Authentication Model Based on Blockchain Technology. The use of blockchain technology to build a decentralized PKI eliminates the single point of failure caused using CA. If the CA certification node is destroyed, the entire certificate chain may be damaged [23]. Compared to the WoT-based PKI, blockchain-based PKI (PB-PKI) has more advantages. WoT-based PKI has a higher entry threshold and needs more workload to build a trusted network. In the blockchain-based PKI, the proof of web members is not needed between entities, so the workload of executing as network members is eliminated.

The core idea of the PKI system based on blockchain technology is to record user certificates through the public ledger. In 2014, MIT scholar Conner first proposed Certcoin [24]. The core idea of Certcoin is to maintain the public ledger of domain names and the related public keys. The process of account and certificate issuance is accessible by users, which can be queried. This process is to solve the issues related to the single point of failure and certificate management and maintenance in the traditional CA system. However, Certcoin's operations (registration, update, and verification) are publicly published in the form of transactions through the blockchain. All actions performed using the public key can be traced to the identity owned by the public key by any entity viewing the ledger, so it does not apply to the scenario where the user's identity privacy needs

to be protected. Based on this, Axon and Goldsmith [25] improved the Certcoin model and offered privacy-awareness to the PB-PKI models. This model provides an unlinkable short-term key update and user control mechanism, in which the identity of the user and the previously used public keys can be revealed by the user itself or through a consensus of the network and use offline keys and online keys for user privacy protection. Privacy protection reduces the risk of users' privacy information leakage.

Aiming at the reputability problem of the traditional credit system in the cloud network transaction architecture, Zhu and Fu [26] proposed a blockchain-based dynamic multicenter collaborative authentication model tailored to the B2B + B2C supply chain. For joint authentication of supply chain transaction behavior, this model uses multiple transaction entities as different authentication centers, which eliminates the problems of tampering with transaction records, fraudulent customers, and single points of failure that exist in the traditional single authentication center, and improves the provability and reliability of transaction behavior. Stability ensures high consistency, transparency, and authenticity of the transaction information.

Chen et al. [27] explore the ways of overcoming the unified trust service challenge of the national PKI via consensus. Furthermore, they applied some functions of the CA management to the blockchain and eventually proposed a blockchain-based revocation list (BCRL). The release cycle of the blockchain revocation list (BCRL) is much shorter than traditional and incremental CRL. The update cycle reaches ten seconds, effectively improving the security of certificate cross-domain verification and authentication. However, this solution consumes a lot of storage. For every thousand transactions (mainly the status of the certificate), a peer node consumes about 3.2 M, and a subscriber node consumes around 3.7 M storage cost, which is not suitable for a large number of nodes and access more frequent systems.

To easily detect malicious certificates when issued, Al-Bassam [28] proposed a decentralized and transparent PKI system by combining smart contracts and the Web-of-Trust model. The design of this model alleviates the verification of fine-grained attributes of another entity's identity (such as company name or domain name), realizing the trust transfer relationship between the identity and attributes of the entity.

2.2. Build a Cross-Domain Authentication Model for Consortium Blockchains. Zhou et al. [29] proposed a blockchain-based PKI interdomain authentication scheme and designed the trust model and the system architecture of the blockchain certificate authority CA (BCCA). The root CA that joins the consortium blockchain in the BCCA model was credible. As a VP, the root CA blockchain certificate was self-generated, and the hash value of the certificate was recorded in the blockchain, which was not easily tampered with as the trust certificate for each domain.

Wang et al. [30] propose a cross-domain authentication model based on consortium blockchain (BlockCAM) and an accompanying protocol. BlockCAM builds a decentralized network with the root certificate authority as the verification node. Each block stores the hash value of the authorization certificate, and the verification process only needs to compare whether the user-provided hash value calculated by the certificate is consistent with the hash value stored in the blockchain. The authentication process omits the key encryption and decryption overhead, thereby improving authentication efficiency.

Liu et al. [31] proposed a consortium blockchain-based V2G network cross-administrative domain authentication scheme using the SM9 digital signature algorithm. To reduce the number of signatures and verifications in the scheme and improve the efficiency and scalability of the program, a hash algorithm is used in the digital identity verification process.

Given the frequent exchange of trust domain information and the inability of secure and efficient authentication between domains, Ma et al. [32] develop a blockchain-based cross-heterogeneous domain authentication scheme, where the consortium blockchain model consists of the blockchain domain proxy server in the IBC domain and the PKI domain blockchain certificate server. The cross-domain model designs cross-domain authentication protocols and reauthentication protocols. It reduces the computing, communication, and storage burden of the combined terminal; simplifies the reauthentication process; and realizes the safe and efficient communication between IBC and PKI. However, this cross-domain authentication scheme cannot address user identity and certificate update and revocation issues. Blockchain data will only increase and not decrease, which will cause waste of the entire system due to data storage.

Facing user privacy scenarios, Ma et al. [33] proposed a blockchain-based distributed key management architecture (BDKMA) to reduce latency by using fog computing and to achieve cross-domain access through multiple cloud-based blockchains.

Jia et al. [34] proposed a cross-domain authentication scheme for the IoT based on identity (IRBA). Innovated traditional authentication schemes include the IBC-based cross-domain authentication methods and threshold passwords and smart contract-based multidomain joint authorization mechanisms. By joint utilization of these methods, a decentralized cross-domain authentication model is realized. This model has greatly improved the cost of calculation and communication. However, this scheme uses complex bilinear pairing operations in the signature and verification process, and there is still room for improvement.

Shu et al. [35] described a two-tier system model for medical data sharing, in which medical records are stored outside the blockchain and shared in the blockchain. In this model, a blockchain-based MCP certificate-less set signature scheme is proposed by using the proposed multinotch hash function. The purpose is to realize the certification of relevant medical personnel, medical equipment, and medical applications; ensure the integrity of medical records; and support the safe storage and sharing of medical information. This scheme includes a cross domain authentication protocol (MCPSP). The proposed cross domain authentication protocol is based on elliptic curve, which has higher

computational efficiency and lower computational cost. However, this scheme has the same problems as the scheme proposed by Jia et al. and does not consider the different cryptographic system parameters (system master key) in different domains and the private key of secure channel transmission.

The existing data sharing systems based on consortium blockchain, they mainly use the certificate system or the bilinear pairing operation to complete the cross-security domain user access control, resulting in substantial management and calculation costs. Hence, lightweight cross-domain authentication schemes, suitable for frequent access in the IoT, still have a wide research space.

3. Certificate-Less Cross-Domain Signature Algorithm for Authentication

The certificate-less signature algorithm is of utmost importance to the cross-domain authentication scheme. When a user joins the data sharing system, it must be registered; when performing cross-domain access, the key steps of the signature algorithm are executed. First, users who apply for cross-domain, signing a request message with the registered private key. Then, the target security domain needs performing signature verification to verify the validity of the identity. When the verification is passed, verify the authority to complete the cross-domain identity authority authentication. The following details the proposed Certificate-less Cross-domain Signature Algorithm with Different System Parameters (DSP-CCSA) that supports parameter differentiation and discusses its correctness and security.

3.1. Procedure of the DSP-CCSA. DSP-CCSA mainly includes seven stages: setup, secret-value-set, partial-private-key-set, private-key-set, public-key-set, sign, and verify. Algorithm 1 shows the detailed process of DSP-CCSA.

The first five stages are performed during registration, where the last two are performed meanwhile execution. The registration is interactively executed by the authentication server and the device.

3.1.1. The Setup. First, the security parameter λ is used as the input, and the public system cryption parameter (CSP) of the Key Generation Center (KGC) is returned as follows:

(1) Assume that there exists a root KGC, which calculates and creates a tuple $\{q, G\}$ according to λ, where G refers to an additive cyclic group and q denotes the order of group G. Then, choose 4 hash functions: $H_1 : \{0,1\}^* \times G^2 \longrightarrow Z_q^*, H_2 : G^3 \longrightarrow Z_q^*, H_3 : G^2 \times (Z_q^*)^2 \longrightarrow Z_q^*, H_4 : G^2 \longrightarrow \{0,1\}^*$

(2) The KGC of each security domain generates a tuple $\{s_k, P_k\}$, where P_k is the generator of group G and $s_k \in Z_q^*$ is the master private key of KGC. KGCs in different security domains can generate different tuples $\{s_k, P_k\}$

(3) The KGC of each security domain calculates its master public key $\mathrm{KC}_k = s_k P_k$; each KGC publishes the system parameters $\{q, G, P_k, KC_k, H_1, H_2, H_3, H_4\}$ and secretly saves its master private key s_k

3.1.2. The Secret-Value-Set. A user UE_k whose identity information is $\mathrm{ID}_{\mathrm{UE}_k}$ chooses a random secret value $x_{\mathrm{UE}_k} \in Z_q^*$, calculates $\mathrm{PK}_{\mathrm{UE}_k} = x_{\mathrm{UE}_k} \cdot P_k$, and sets x_{UE_k} as his secret value (where it is assumed that UE_k is a user in the security domain D_k and is connected to KGC in the domain k with system parameters P_k).

3.1.3. The Partial-Private-Key-Set. The algorithm uses the system cryption parameters, master private key, user identity, and public key of the KGC as the input and returns part of the private key for users in the domain.

(1) The user equipment UE_k in the security domain D_k submits its identity information $\mathrm{ID}_{\mathrm{UE}_k}$ and part of the public key $\mathrm{PK}_{\mathrm{UE}_k}$

(2) After receiving the registration information sent by the user, KGC in the security domain D_k randomly selects $r_i \in Z_q^*$ and calculates $R_i = r_i \mathrm{KC}_k$, $h_i = H_1(\mathrm{ID}_i, R_i, \mathrm{PK}_{\mathrm{UE}_k})$.

(3) KGC in the security domain D_k further calculates $sk'_{\mathrm{UE}_k} = r_i \cdot s_k \cdot h_i + H_1(\mathrm{ID}_{\mathrm{UE}_k}, R_i, s_k, \mathrm{PK}_{\mathrm{UE}_k})$ and sends $\{sk'_{\mathrm{UE}_k}, R_i\}$ to the user via the public channel

3.1.4. The Private-Key-Set. When the user receives the message $\{sk'_{\mathrm{UE}_k}, R_i\}$ returned by KGC in the security domain D_k, the user can calculate $sk_{\mathrm{UE}_k} = sk'_{\mathrm{UE}_k} - H_1(\mathrm{ID}_{\mathrm{UE}_k}, R_i, x_{\mathrm{UE}_k}, \mathrm{KC}_k)$ to verify whether the message $\{sk'_{\mathrm{UE}_k}, R_i\}$ is valid and check whether the equation $sk_{\mathrm{UE}_k} P_k = h_i R_i$ is true; if it is true, the user sets $\{sk_{\mathrm{UE}_k}, x_{\mathrm{UE}_k}\}$ as its complete private key. Suppose that the full private key of KGC is $sk_{\mathrm{KGC}_k} = (x_k + r_k s_k h_k) \bmod q$.

3.1.5. The Public-Key-Set. Since the user UE_k in the security domain D_k sets $\{\mathrm{PK}_{\mathrm{UE}_k}, R_i\}$ as its complete public key, we consider $\mathrm{PK}_{\mathrm{KGC}_k} = sk_{\mathrm{KGC}_k} P_k$ as KGC's complete public key.

3.1.6. Signs. If the user UE_1 in the security domain D_1 is aimed at using the service of the security domain D_2, the UE_1 first sends an authentication request $\{\mathrm{request}, \mathrm{PK}_{\mathrm{UE}_1}\}$ to the KGC_{D_2} in security domain D_2.

After KGC_{D_2} receives the authentication request message, it will generate a random number $N \in Z_q^*$ and send a response message $\{N, \mathrm{PK}_{\mathrm{UE}_k}\}$ to the user device UE_1.

In response to the KGC_{D_2}, the user equipment UE_1 performs the following calculations $U_1 = (N \cdot x_{\mathrm{UE}_1}) P_2$, $T_1 = (N \cdot x_{\mathrm{UE}_1}) \mathrm{PK}_{\mathrm{KGC}_2}$, $y_1 = H_2(U_1 || T_1 || \mathrm{PK}_{\mathrm{KGC}_2})$, $Q_1 = (N + y_1) P_1$, $V_1 = (y_1 \cdot sk_{\mathrm{UE}_1})^{-1} (N + x_{\mathrm{UE}_1} + y_1)$, $\mathrm{MID}_1 = H_4(U_1 || T_1) \oplus (\mathrm{ID}_1 \| V_1 \| C_1 || T_m)$. T_m is the current timestamp, sending a message $\{U_1, \mathrm{MID}_1\}$ to KGC_{D_2} in the security domain D_2.

1/∗G refers to the additive cyclic group, and q denotes the order of group G
2$\{0,1\}^* \times G^2 \longrightarrow Z_q^*, H2 : G^3 \longrightarrow Z_q^*$,
3$G^2 \times (Z_q^*)^2 \longrightarrow Z_q^*, H4 : G^2 \longrightarrow \{0,1\}^*$
4 ∗ /
5//Registration phase
6[Setup**]:**
7$GenMP(k) \longrightarrow \{q, G, H1, H2, H3, H4\}$ // Generate main system parameters
8 $Gen(G) \longrightarrow \{s_k, P_k\}$ //Each domain randomly generates its system parameters
9 $\{q, G, P_k, KC_k, H1, H2, H3, H4\} \longrightarrow params$ // Release parameters
10[Secret-value-set**]:**
11 $Gen(params) \longrightarrow x_{UE_i}$
12[Partial-private-key-set**]:**
13 $GenSk(params, ID_{UE_i}, PK_{UE_i}) \longrightarrow \{sk_{UE_i}^{\cdot}, R_i\}$
14[Private-key-set**]:**
15 $GenS_k(params, sk'_{UE_{ii}}, R_i, ID_{UE_i}, x_{UE_i}) \longrightarrow \{sk_{UE_i}, x_{UE_i}\}$
16[Public-key-set**]:**
17 $GenP_k(x_i, R_i) \longrightarrow \{PK_{UE_i}, R_i\}$
18//Execution phase
19[Sign**]:**
20 $SigID(m, params, pk_{KGC}, P_2) \longrightarrow \{U_i, MID_i\}$
21[Verify**]:**
22 $Ver(U_i, MID_i, params, sk_{\mathrm{KGC}}) \longrightarrow valid/invalid$

Algorithm 1: DSP-CCSA.

3.1.7. Verify. When KGC_{D_2} in security domain D_2 receives the response message $\{U_1, \mathrm{MID}_1\}$ from user UE_1 in security domain D_1 at time T_c, it performs the following operations:

(1) Calculate $T_1' = sk_{\mathrm{KGC}_2} U_1, (\mathrm{ID}_1 || V_1 || C_1 || T_m) = \mathrm{MID}_1 \oplus H_4(U_1 || T_1')$, if $T_c - T_m < \Delta t$, within the reauthentication time, pass the authentication directly, otherwise proceed to step (2).

(2) Calculate $y_{i_1}' = H_2(U_1 || T_1' || \mathrm{PK}_{\mathrm{KGC}_2})$, $Q' = y_{i_1}' h_1 V_1 R_1 - \mathrm{PK}_{\mathrm{UE}_1}$, $C_1' = H_3(Q_1' || y_1' || V_1 || \mathrm{PK}_{\mathrm{UE}_1})$, if $C_1 \neq C_1'$, the authentication fails, otherwise, passes

3.2. Correctness Analysis. When the user in the security domain D_1 sends a signed message $\{U_1, \mathrm{MID}_1\}$ to the KG C_{D_2} in the security domain D_2, the KGC_{D_2} has to validate that the data is valid.

Proof. Because $T_1' = sk_{\mathrm{KGC}_2} U_1 = sk_{\mathrm{KGC}_2} N \cdot x_{\mathrm{UE}_1} P_2 = (N \cdot x_{\mathrm{UE}_1})\mathrm{PK}_{\mathrm{KGC}_2}$

Thus, $y_{i_1}' = H_2(U_1 || T_1' || \mathrm{PK}_{\mathrm{KGC}_2}) = y_1$

Thus

$$\begin{aligned} Q^{\cdot} &= y_{\cdot_1}^{\cdot} h_1 V_1 R_1 - \mathrm{PK}_{\mathrm{UE}_1} = y_{i_1}' h_1 (y_1 \cdot sk_{\mathrm{UE}_1})^{-1} (N + x_{\mathrm{UE}_1} + y_1) R_1 - x_{\mathrm{UE}_1} P_1 \\ &= y_{i_1}' h_1 (y_1 r_1 s_1 h_1)^{-1} (N + x_{\mathrm{UE}_1} + y_1) r_1 s_1 P_1 - x_{\mathrm{UE}_1} P_1 \\ &= (N + x_{\mathrm{UE}_1} + y_1) P_1 - x_{\mathrm{UE}_1} P_1 = (N + y_1) P_1 = Q. \end{aligned} \tag{1}$$

Thus, $C_1' = H_3(Q_1' || y_1' || V_1 || \mathrm{PK}_{\mathrm{UE}_1}) = C_1$. □

3.3. Safety Analysis. In this part, it is proved that the proposed DSP-CCSA is safe in the random prediction model. In DSP-CCSA, the communication entities originated from different security domains can employ different system parameters, CSP. Using different CSPs is safer than using the same CSP. The security of the proposed Certificate-less Cross-domain Signature Algorithm is affected by the difficulty of certain mathematical problems. To better understand the following security proof, a brief introduction to the mathematical assumptions is made firstly.

(1) The elliptic curve discrete logarithm (ECDL) problem

The problem of ECDL calculates the integer value of $x \in Z_q^*$ for a prime q order additive cyclic group G by the setting $Q = xP(P, Q \in G)$. However, given P and Q, there are no known algorithms that can effectively determine x, and the use of brute force methods is computationally expensive; that is, assuming that the base point of the elliptic curve is known, it is impossible to find the discrete logarithm corresponding to a random element [36].

(2) The Diffie-Hellman decision calculation problem (DCDH)

The problem of DCDH is to determine whether the equation $\xi = abP$ holds for a random instance (P, aP, bP, ξ), where $a, b \in Z_q^*$ and $P \in G$. [37].

3.3.1. Security Analysis of Cross-Domain Authentication Protocol

(1) The proposed cross-domain authentication scheme realizes the security of the cross-domain authentication protocol (CAP) against the adversary under the assumption of the DCDH problem

Type I adversary. This type of adversary $A^{\wedge}{}_I$ is a dishonest user. We supposed that $A^{\wedge}{}_I$can acquire the public keys and secret values of KGC (AS) and group users. The public keys of KGC (AS) and user can be replaced by $A^{\wedge}{}_I$. $A^{\wedge}{}_I$ can not acquire the master private key of KGC (AS).

Type II adversary. This type of adversary $A^{\wedge}{}_{\coprod}$ is modeled as a malicious KGC (AS). We supposed that Â2 can acquire the master private key of KGC (AS). The public keys of KGC (AS) and user cannot be replaced by $A^{\wedge}{}_{\coprod}$.

Proof. Assume that $C^{\wedge}$ is the challenger of the Diffie-Hellman decision calculation problem, and he has a living example of the DCDH problem (P, aP, bP, ξ). If $C^{\wedge}$ can distinguish $\xi = abP$, it can help the opponent $A^{\wedge}(A^{\wedge}{}_I \text{or} A^{\wedge}{}_{\coprod})$ by winning the next game to destroy the CAP security of the proposed DSP-CCSA [38]. □

Initialization: $C^{\wedge}$ chooses a random identity C_{ID} as the identity of the KGC, it wants to challenge. Afterward, $C^{\wedge}$ creates the CSP and the pair of master private and public keys $(s \in Z_q^*, \mathrm{PK} = sP)$ of the KGC. Then, the cryptographic system parameters and public key of the security domain are returned to the opponent $A^{\wedge}$, and the private key s is transferred to $A^{\wedge}{}_{\coprod}$.

Probe: The below query is executed:

(i) *Hash Query*. $C^{\wedge}$ retains four lists $L_i (i = 1, 2, 3, 4)$, which represent the corresponding Hash query H_i $(i = 1, 2, 3, 4)$. All lists are empty when initialized. When $A^{\wedge}$ submits the corresponding message m_j for H_i query, if there is a tuple $\{m_j, h_j\}$ in the corresponding hash list L_i, the corresponding hash value h_j will be returned. Otherwise, $C^{\wedge}$ randomly selects a value $h_j \in H_i$ and stores it in the list L_i, and finally, $C^{\wedge}$ returns h_j to $A^{\wedge}$

(ii) *Secret Value Query*. $C^{\wedge}$ retains a list L_s, empty at first, for secret value query. When $A^{\wedge}$ enters the user identity information UID_i for query, if there is a record $\{\mathrm{UID}_i, x_i, pk_i\}$ in the list L_s, then $C^{\wedge}$ returns x_i. Otherwise, $C^{\wedge}$ randomly selects a value $x_i \in Z_q^*$ and calculates $pk_i = x_i P$, and finally, $C^{\wedge}$ stores $\{\mathrm{UID}_i, x_i, pk_i\}$ in L_s and returns x_i to $A^{\wedge}$

(iii) *Partial Private Key Query*. $C^{\wedge}$ retains a list L_p, empty at first, for this query. When $A^{\wedge}$ enters user identity information UID_i for this query (assuming that the secret value query is executed beforehand), if there are records $\{\mathrm{UID}_i, sk_i, R_i\}$ in the list L_p, then $C^{\wedge}$ returns sk_i. Otherwise, when $A^{\wedge}$ is $A^{\wedge}{}_I$, $C^{\wedge}$ randomly selects a value $r_i \in Z_q^*$ and then calculates $R_i = r_i \cdot pk_i$, $h_i = H_1(\mathrm{UID}_i, Ri, pk_i)$, $sk_i = r_i \cdot h_i \cdot x_i \bmod q$. When $A^{\wedge}$ is $A^{\wedge}{}_{\coprod}$, $C^{\wedge}$ randomly selects a value $r_i \in Z_q^*$ and then calculates $R_i = r_i \cdot \mathrm{PK}$, $h_i = H_1(\mathrm{UID}_i, Ri, pk_i)$, $sk_i = r_i \cdot h_i \cdot s \bmod q$. Finally, $C^{\wedge}$ stores $\{\mathrm{UID}_i, sk_i, R_i\}$ in L_p and returns sk_i

(iv) *Private Key Query*. It is assumed that the secret value query and partial private key query have been queried before executing this query. When $A^{\wedge}$ enters the user identity information UID_i for this query, $C^{\wedge}$ returns the data $\{x_i, sk_i\}$ in the lists L_p and L_s. When $A^{\wedge}$ enters KGC identity information $C_{\mathrm{ID}j}$ for this query, $C^{\wedge}$ updates the list L_r with the initial tuple $\{C_{\mathrm{ID}}, \perp, bp\}$. If $C_{\mathrm{ID}j} = C_{\mathrm{ID}}$, then $C^{\wedge}$ returns "fail"; otherwise, $C^{\wedge}$ randomly selects a value $sk_j \in Z_q^*$, then calculates $pk_j = sk_j P$, and then, $C^{\wedge}$ returns sk_j. Finally, $C^{\wedge}$ stores $\{C_{\mathrm{ID}j}, sk_j, pk_j\}$ into the list L_r

(v) *Public Key Query*. When $A^{\wedge}{}_I$ enters the tuple information $\{\mathrm{UID}_i, x_i', r_i'\}$ for this query, $C^{\wedge}$ executes the secret value query and partial private key query by using $\{UID_i, x_i', r_i'\}$, generate new values $\{sk_i', pk_i', R_i'\}$, and then use $\{\mathrm{UID}_i, x_i', pk_i'\}$ to query the corresponding result $\{\mathrm{UID}_i, x_i, pk_i\}$ in L_s, use $\{UID_i, sk_i', R_i'\}$ to query the corresponding tuple $\{\mathrm{UID}_i, sk_i, R_i\}$ in L_p, and return the query result. When $A^{\wedge}{}_I$ enters the tuple information $\{C_{\mathrm{ID}j}, sk_j'\}$ to perform this query, if $C_{\mathrm{ID}j} = C_{\mathrm{ID}}$, $C^{\wedge}$ returns "fail"; otherwise, $C^{\wedge}$ is performed by using $\{C_{\mathrm{ID}j}, sk_j'\}$ query the private key and generate a new pk_j' value. Finally, $C^{\wedge}$ returns $\{C_{\mathrm{ID}j}, sk_j', pk_j'\}$ the corresponding value $\{C_{\mathrm{ID}j}, sk_j, pk_j\}$ in the list L_r

(vi) *Send Query*. When $A^{\wedge}$ logs a query with a request message $\{m_s, C_{IDj}\}$, $C^{\wedge}$ first selects a random integer $\alpha \in \{1, 2, \cdots, q_s\}$ (assuming that $A^{\wedge}$ executes this query at most q_s times). If $C_{\mathrm{ID}j} = C_{\mathrm{ID}}$ and $k = \alpha$, $C^{\wedge}$ sets $U_i = U_\alpha = aP$. At the same time randomly selects a $\mathrm{MID}_i = \mathrm{MID}_\alpha \in \{0, 1\}^*$. Otherwise, $C^{\wedge}$ executes the signature operation and generates the response message $\{U_i, \mathrm{MID}_i\}$. Finally, $C^{\wedge}$ returns $\{U_i, \mathrm{MID}_i\}$

(vii) *Test Query*. When $A^{\wedge}$ logs a query with request message $\{C_{\mathrm{ID}j}, U_i, \mathrm{MID}_i\}$, if $C_{\mathrm{ID}j} \neq C_{\mathrm{ID}}$, $U_i \neq U_\alpha$, $\mathrm{MID}_i \neq \mathrm{MID}_\alpha$, $C^{\wedge}$ declares "fail" or randomly picks a value $b \in \{0, 1\}$ to perform some operations: If $b = 1$, $C^{\wedge}$ sets $T_i' = T_\alpha = \xi$, calculates $ID_i || V_i || C_i || T_{mi} = \mathrm{MID}_i \oplus H_4(U_i || T_i')$, $y_i' = H_2(U_i || T_i' || pk_{\mathrm{KGC}})$, $Q_i' = y_i' h_i V_i R_i - pk_i$, $C_i' = H_3(Q_i' || y_i' || V_i || pk_i)$. Verify that whether $C_i' = C_i$ is established

Finally, $A^{\wedge}$ returns a guess bit $b^{'} \in \{0, 1\}$. If $b' = b$, then $A^{\wedge}$ can destroy the CAP security of DSP-CCSA, because $C^{\wedge}$ can crack the DCDH problem by discriminating $T_{\alpha} = \xi = abP$. However, it is accepted that there is not a known method to solve the DCDH problem in polynomial time. Therefore, the proposed certificate-less signature algorithm realizes CAP security against the adversary $A^{\wedge}$ under the assumption of DCDH.

3.3.2. Security Analysis of Partial Private Key Transmission. In the proposed certificate-less cross-domain authentication algorithm, the KGC in the security domain D_k calculates the partial private key $sk'_{\mathrm{UE}_k} = r_i \cdot s_k \cdot h_i + H_1(\mathrm{ID}_{\mathrm{UE}_k}, R_i, s_k, \mathrm{PK}_{\mathrm{UE}_k})$ of user UE_k. An attacker $C^{\wedge}$ can obtain part of the private key sk'_{UE_k} transmitted on the public channel. However, under the assumption that the ECDL problem is difficult to solve, he does not have the master key s_k of the security domain D_k or the secret value of the user UE_k. It is impossible to calculate the user's real partial private key $sk_{\mathrm{UE}_1} = r_i \cdot s_k \cdot h_i$, so DSP-CCSA is safe for partial private key public channel transmission.

3.3.3. Antireplay Attack. Whenever UE_i signs a specific message m, it selects a new timestamp T_m. If the attacker intercepts the interactive information $< U_i, \mathrm{MID}_i = H_4(U_i || T_i) \oplus (\mathrm{ID}_{\mathrm{UE}_i} || V_i || C_i || T_m) >$ and replies it to the KGC in the security domain D_k, by verifying the freshness of the timestamp T_m, the KGC can determine that is a reply message.

4. Certificate-Less Cross-Domain Authentication Scheme Supporting Parameter Differentiation

4.1. Scheme Model. The identity authentication process consists of two main stages: (i) authorization and (ii) authentication [39]. In the former, the security domain D_A validates the authenticity of the device identity in the domain. After passing the authenticity verification, the authorization is granted, and the smart contract is used to automatically obtain the authorization joint signature of the domain to be accessed and recorded in the blockchain. In the latter, mainly, the validity of the device identity and network access rights are checked. Considering these, this article introduces a Certificate-less Cross-domain Authentication Scheme with Different System Parameters (DSP-CCAS). The scheme model shown in Figure 1 has two aspects: (i) it uses the smart contract and consensus mechanism of blockchain to replace the trusted third-party authorization process, and (ii) it completes the cross-domain authentication process using the proposed DSP-CCAS algorithm.

In Figure 1, each dashed box represents a security domain, which represents a data sharing unit. The local domain authentication server (AS) plays the role of KGC and completes the authentication calculation of user identity and permission to access the domain. The APP server stores the actual shared data. And the terminal represents the user terminals in the IoT who want to obtain cross-domain authorization data. Ultimately, the consortium blockchain is a consortium blockchain composed of authentication servers in each security domain that stores user permissions. Security domains A and B build a consortium relationship. When user X in security domain A requests data in security domain B, traditional RSA authentication is not used, but a self-authentication method is used. In this way, the authentication server of the security domain B can complete the authentication of the X identity without the authority of a trusted third party. Next, check whether the user authority record stored on the blockchain contains the authorization result for the user X; if it exists, the authentication is successful; otherwise, the authentication fails. Compared with the centralized authentication model, decentralized authentication can guarantee the autonomy and initiative of the security domain. There is no need to rely on a third party to dynamically adjust the mutual trust relationship.

4.2. Authorization Mechanism

4.2.1. Smart Contract in Authorization Mechanism. Smart contracts mainly refer to general-purpose calculations performed on blockchains or distributed ledgers. They are composed of computer code and constitute a set of rules or conditions agreed by the parties. When these predefined rules or conditions are met, the smart contract will execute itself and provide output [40]. To request and publish cross-domain permissions, the following three types of smart contracts are used to implement a complete traceable, irreversible, and secure authorization process in a fully distributed environment without centralized trusted institutions.

(i) *The Main Contract*. It accepts authorization requests and maintains a list of applications. The blockchain consists of only one master contract, and the blockchain address is known by all entities. The authentication server needs to use the master contract to establish a new authorization contract, obtaining cross-domain authorization

(ii) *Authorization Contract*. A product of the main contract, which receives the authorization signature. It can be automatically executed in the blockchain to collect the permissions granted by each security domain to the user

(iii) *Storage Contract*. It acts as the receiver of transactions containing authorized data and signatures

4.2.2. Cross-Domain Access Permission Acquisition. Figure 2 shows the authorization process when the device UE_1 belonging to the domain D_1 tries accessing the service of another domain.

The parameter descriptions are provided in Table 1.

Step 1. User UE_1 from domain D_1 applies for cross-domain access authorization. Then, he sends the application of cross-domain access authorization $\mathrm{Sig}sk_{\mathrm{UE}_1}(\mathrm{Request})$ signed with his private key sk_{UE_1} to the local authentication server AS_1.

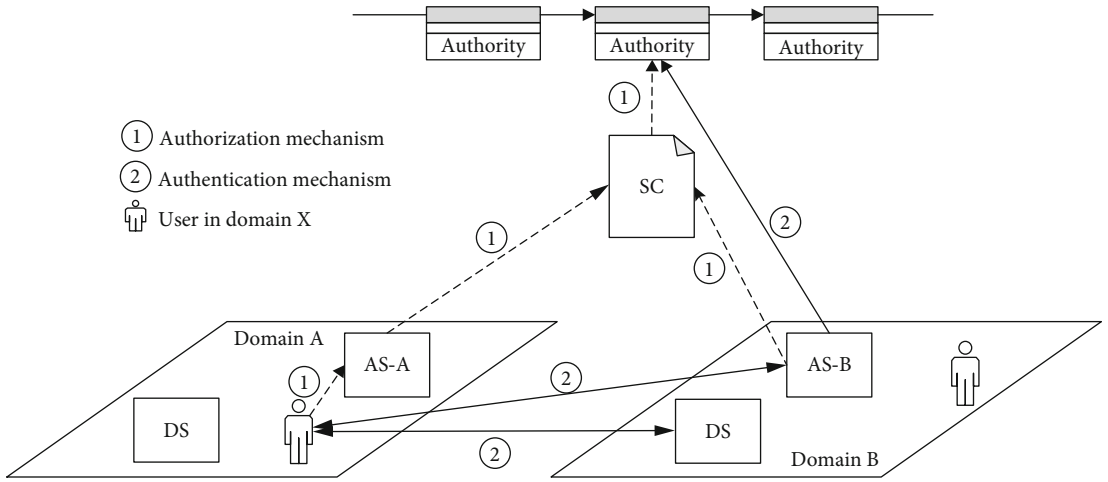

Figure 1: Scheme model.

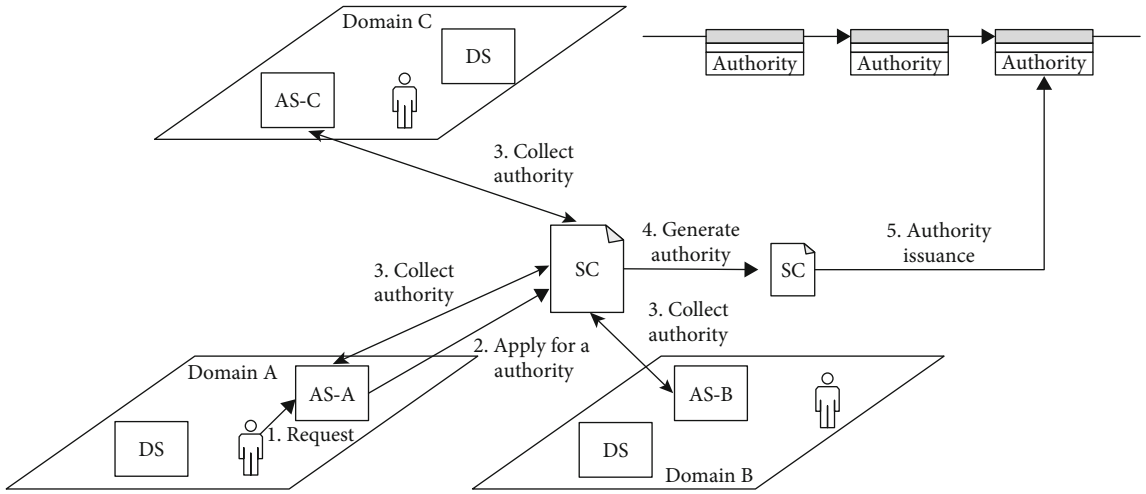

Figure 2: Authorization process for cross-domain access.

Table 1: Parameter descriptions.

Parameter	Description
UE_i	User in domain i
AS_i	Authentication server in domain i
$Sig_{sk}()$	Sign with the private key sk
PK_{UEi}	The public key of UE_i
P_i	Generator in domain i
BC	Blockchain
SK_X	The private key of X
$Authority_i$	Authority of UE_i

Step 2. After AS_1 receives the cross-domain application from UE_1 in the same domain, it first uses the public key of UE_1 to verify the request message. After the verification is passed, AS_1 uses the master contract to create an authorization contract and specifies the AS_1 address of the security domain to be accessed to collect the signature.

Step 3. The authorization contract encrypts the authorization request with the public key of the identity authentication server in the designated security domain and then sends it to the identity authentication server of the corresponding security domain.

Step 4. The authorization contract collects user authority records and submits the collected user authority records to the storage contract.

Step 5. The storage contract stores the authorized transaction in the blockchain after packing it into a block.

4.3. The Cross-Domain Authentication Process. Now, let us assume that the user UE_1 in the security domain D_1 issues a cross-domain authentication request to AS_2 in the security domain D_2 as illustrated in Figure 3, and the protocol procedure is detailed afterward.

(1) $UE_1 \longrightarrow AS_2 : \{\text{Access Request}, PK_{UE_1}\}$

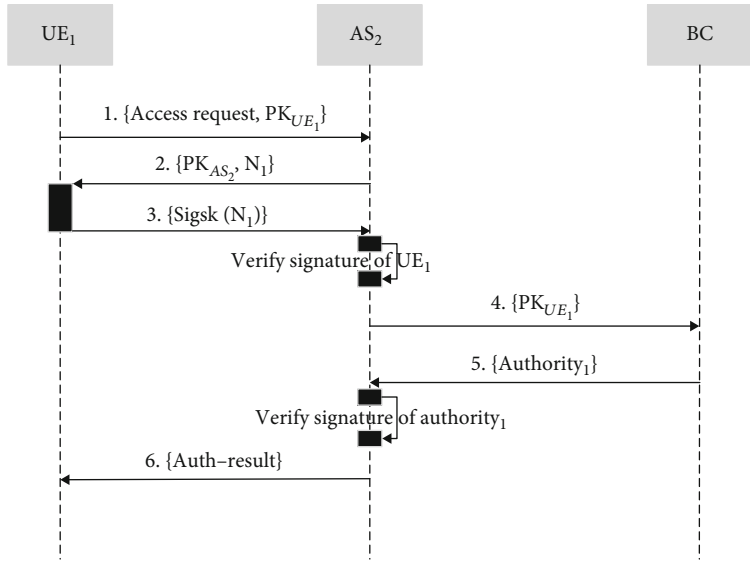

Figure 3: Cross-domain authentication process.

The user UE_1 in the security domain D_1 sends a request for cross-domain authentication to the user authentication server AS_2 in the security domain D_2 and sends the public key PK_{UE_1} required in the subsequent process together.

(2) $AS_2 \longrightarrow UE_1 : \{PK_{AS_2}, N_1\}$

After receiving the cross-domain authentication request from user UE_1 in the security domain D_1, the authentication server AS_2 in the security domain D_2 sends the random number N_1 and the public key PK_{AS_2} of the security domain D_2 to the user UE_1 in the security domain D_1. To complete the cross-domain authentication needs to support different system parameters.

(3) $UE_1 \longrightarrow AS_2 : \{Sigsk_{UE_1}(N_1)\}$

(i) After the user UE_1 in the security domain D_1 receives the response from the AS_2 in the security domain D_2, it uses the private key SK_{UE_1} to generate the identity signature $Sigsk_{UE_1}(N_1)$

(ii) The user UE_1 in the security domain D_1 responds to the message of the authentication server AS_2.Then, it sends the signature $Sigsk_{UE_1}(N_1)$ of the random number N_1 to the authentication server AS_2 in the security domain D_2

(4) $AS_2 \longrightarrow BC : \{PK_{UE_1}\}$

(i) The authentication server AS_2 in the security domain D_2 uses the public key PK_{UE_1} of user UE_1 in the security domain D_1 to verify the signed message $Sigsk_{UE_1}(N_1)$

(ii) After the verification is passed, the authentication server AS_2 queries the authorization result of UE_1 from the blockchain

(5) $BC \longrightarrow AS_2 : \{Authority_1\}$

If there is an authorization record of the security domain D_2 for the user UE_1 in the query record, the authentication is passed; otherwise, the authentication fails.

(6) $AS_2 \longrightarrow UE_1 : \{Auth - result\}$

The authentication server AS_2 in the security domain D_2 returns the authentication result of user UE_1 in the security domain D_1.

5. Experiment and Result Analysis

5.1. Experimental Design

5.1.1. Experiment Environment. A simulation experiment platform was constructed to evaluate the performance of each program. The platform has ten authentication servers installed on a super ledger structure, which formed a consortium blockchain for a security domain. Each server performed cross-domain access authorization and device cross-domain access authentication. The authentication servers establish a structure-based permissioned blockchain. The servers can be set up as various node types, such as confirmation, endorsement, and authentication center nodes. They locally store the identity information of domain members, besides maintaining distributed ledgers and smart contracts. The ordering service deployment employs the Kafka model of fabric to ensure the reliability of the blockchain system, avoiding the single point of failure of the ordering node [41].

The simulation experiment was completed on a Linux server. The authentication server had an Intel-Core i5 6300 HQ CPU (2.30 GHz) with 16 GB memory and CentOS Linux 7.4 operating system. The blockchain platform is Hyperledger Fabric version v1.0, the experimental code writing language is Go, and the version is 1.15.1.

5.1.2. Experimental Design. The experimental design mainly compares the cross-domain authentication performance and authority processing performance of several blockchain-based certificate-less cross-domain authentication schemes.

(1) Experiment 1: Authority Granting Processing Performance. This experiment measures the processing performance of the DSP-CCAS authority granted and verifies whether the solution works well in the resource-constrained IoT data sharing environment. Considering the impact of the size of the security domain in the shared system on the authorization processing performance, the cross-domain access relationship is, respectively, established from 2 to 10 security domains. And the performance of authorized signature calculation is observed to change with the increase of the number of security domains. In other terms, first, the users in 2 security domains, then the users in 4 security domains, and finally the users in 10 security domains can access each other. To facilitate data statistics, a user with cross-domain access requirements is set in each security domain, and the abovementioned access process is performed to calculate the performance of the authorized signature calculation. Table 2 shows the relevant settings of the blockchain during the experiment.

(2) Experiment 2: Authority Verification Processing Performance. In this round of experiments, the processing performance of authorization verification between different numbers of security domains is evaluated, and the approximate time consumption of one authorization verification process is counted. It is validated that the proposed DSP-CCAS scheme is in the IoT massive data system, and the number of security domains continues to increase. Which can verify whether the proposed DSP-CCAS scheme can maintain efficient processing performance and high scalability, with the increase number of security domains in the massive data system of the Internet of Things. Taking the number of security domains T as a variable, where $T = \{2, 4, 6, 8, 10\}$, the test model is repeated a hundred times, and a hundred samples are averaged.

(3) Experiment 3: Cross-Domain Authentication Processing Performance. To validate the performance of the proposed DSP-CCAS cross-domain authentication processing, it includes identity signature time, signature verification time, authority verification time, and communication volume. Compare the proposed scheme (the certificate-less cross-domain authentication scheme DSP-CCAS that supports different system parameters) with the existing blockchain-based certificate-less cross-domain authentication schemes, including IRBA proposed by Jia et al. [34] and MCPSP proposed by Shu et al. [35]. From the comparison of the two indicators of computation time consuming and communication cost, three authentication schemes were loaded on the simulation experiment platform, and two security domains were selected for 100 cross-domain requests, and the average value was taken compare.

IRBA, a pairing-based cross-domain authentication scheme, can be simulated on the bilinear pair $e : G_1 \times G_2 \longrightarrow G_2$. G_1 is the additive group of order q_1 generated on the A-type elliptic curve $E_1 : y^2 = x^3 + x \bmod p_1$, G_2 is the factorial group of q_1 generated by E_1, and p_1 and q_1 are, respectively, 512-bit and 160-bit prime numbers. For the elliptic curve-based cross-domain authentication scheme (MCPSP and DSP-CCAS), the simulation can be performed on the nonsingular elliptic curve $E : y^2 = x^3 + ax + b \bmod p_2$. G is the additive group of order q_2 generated by E, where p_2 and q_2 are two 160-bit prime number. The aforementioned bilinear pair and elliptic curve constructed in the experiment are at the same 80-bit security level. As shown in Table 3, some basic parameter settings in the experiment are given.

Table 2: Blockchain-related parameter settings.

Parameter	Value
Sorting algorithm	Kafka
Size of blockchain	50 transactions per block
Timeout of blockchain	2 seconds

Table 3: Parameter settings.

Parameter	Description	Size/bit
$\|G_i\|$	Size of group G_1 and G_2	128
$\|G\|$	Size of group G	40
$\|\mathrm{ID}\|$	Size of user's identity ID	64
$\|N\|$	Size of random number N	32
$\|T\|$	Size of timestamp	32
q	Size of the element in Z_q^*	20
m	Size of request or result	8

5.2. Result Analysis

5.2.1. Processing Performance Granted by Access Rights. According to the design of experiment 1, the number of security domains T in the data sharing consortium blockchain is continuously adjusted, gradually increasing from $T = 2$ to $T = 10$, and the value of T is increased by 2 each time. That is, starting from 2 security domains, add 2 security domains each time until the number of security domains reaches 10. Through the log records, the average processing time of cross-domain access authorization in the proposed DSP-CCAS scheme and the influence of the number of security domains T on the performance (processing time/sec) of cross-domain access authorization are calculated. According to the statistical data, the results are illustrated in Figure 4.

Figure 4 shows that the time needed to issue authorization changes marginally with the number of data sharing security domains T for the proposed cross-domain authentication scheme (DSP-CCAS) that supports different system parameters. However, the authorization processing time is no more than five seconds, and the authorization duration is no more than three seconds in most cases, which accounts for 65%-72% of the test samples. According to the above data analysis, the proposed DSP-CCAS scheme has a higher performance in the processing of authorization.

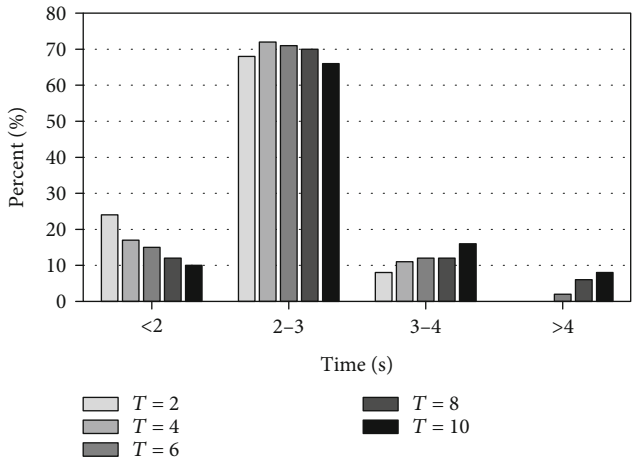

Figure 4: Authorization time allocation for different domain values (T).

5.2.2. Processing Performance of the Authorization Verification. Again, the number of security domains, i.e., the number of authentication servers T, is used as a variable to repeat the test model a hundred times, and a hundred samples are averaged. Figure 5 illustrates the results.

The experimental results showed that the verification time of DSP-CCAS does not dramatically vary with the change of threshold T and is stable at around 4.2 ms, which means that the proposed DSP-CCAS scheme has good scalability.

5.2.3. Processing Performance of the Cross-Domain Authentication. Calculation cost and communication cost are two important factors for evaluating certificate-less cross-domain authentication schemes. According to the design of experiment three, the computational and communication costs of the proposed scheme are compared with two recent blockchain-based certificate-less collective signature schemes, namely, IRBA and MCPSP. Since the registration phase, performance evaluation, and the running time of some lightweight operations minimally affect the overall system performance, they are ignored. Table 4 provides a comparison between the proposed and the related blockchain-based cross-domain authentication schemes in terms of computational cost.

Table 4 reveals that DSP-CCAS has a significant improvement in the calculation time of individual signature, individual verification, and authorization verification compared with the scheme IRBA. This is because the IRBA scheme uses a complex bilinear mapping in the calculation process. The proposed scheme DSP-CCAS and scheme MCPSP are all completed under the elliptic curve cryptosystem. Under the same security level, the elliptic curve cryptosystem is more effective than bilinear mapping. Therefore, the elliptic curve-based certificate-less cross-domain authentication scheme has the characteristics of low calculation, low storage, high reliability, privacy protection, and timeliness. And it is suitable for the sharing of massive data based on blockchain technology in the resource-constrained IoT environment. Compared with the scheme MCPSP based

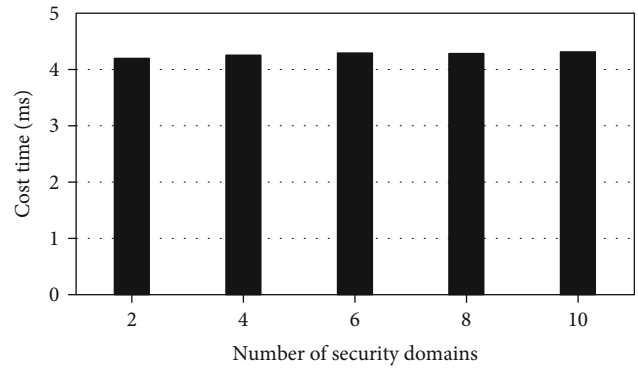

Figure 5: Permission verification time.

Table 4: Comparing calculation costs.

Program	Sign time/ms	Verify signature time/ms	Verify authority time/ms	Total time/ms
IRBA	24.124	8.972	6.627	39.768
MCPSP	2.165	6.534	5.424	14.123
DSP-CCAS	6.495	2.178	4.268	12.941

on the elliptic curve cryptosystem, the proposed scheme DSP-CCAS has a higher computational cost for personal signatures. Because in the scheme, MCPSP personal signatures only use 1 scalar multiplication, but to support different security domains with different system parameters, DSP-CCAS requires 3 scalar multiplications when signing. However, in individual signature verification and authorization verification, DSP-CCAS is better than the scheme MCPSP. This is because, in the verification phase, the scheme MCPSP requires 3 scalar multiplications and 3 scalar additions on the elliptic curve, while DSP-CCAS only needs 1 scalar multiplication and 1 scalar addition to reduce 2 scalar multiplications and 2 scalar addition operations. In terms of overall authentication calculation time (verify signature time + verify authority time), the proposed scheme is significantly better than the related cross-domain authentication schemes IRBA and MCPSP based on blockchain technology, which only takes 6.446 ms. Figure 6 presents the comparison of the proposed DSP-CCAS with IRBA and MCPSP in terms of cross-domain identity authentication performance.

Compared with IRBA that uses complex bilinear pairing operations, the proposed DSP-CCAS scheme has greatly improved performance at all stages. It not only decreases the signature time by 73.07% but reduces the signature verification time by 75.72%. The authorization verification time also reduced by 35.6%. Even the overall authentication time is decreased by 67.46%. As for comparing with the MCPSP scheme that is the same as based on the nonsingular elliptic curve, although the signature time of DSP-CCAS is 66.67% longer than it, we greatly shorter the signature verification time and authority verification time, which are decreased 66.67% and 21.31%. Respectively, the overall cost of verification time has reduced by 8.37%. It is a significant improvement. From the above analysis, it can be seen that,

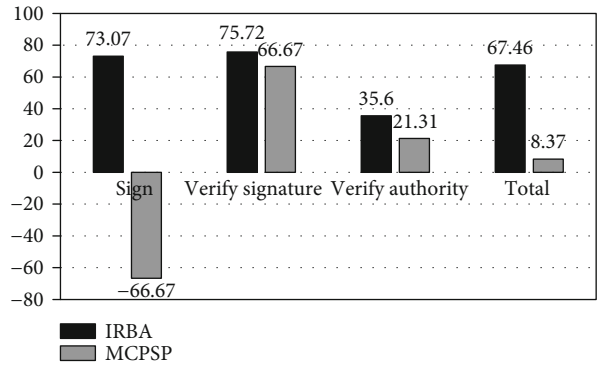

Figure 6: Comparison of the time of each stage of certification.

Table 5: Communication cost of blockchain-based solutions.

Program	Communication volume/bit
IRBA	Request+2|ID| $+N$ $+2(2\|G_1\| +2\|G_1\|)$ + result $= 8 + 2 * 64 + 32 + 2(2 * 128 + 2 * 128) + 8$ $=1200$
MCPSP	Request+2|ID| $+2\|G\| +N$ $+2(\|G\| +2q)$ + result $= 8 + 2 * 64 + 2 * 40 + 20 + 2(40 + 2 * 20) + 8$ $=404$
DSP-CCAS	Request+4|G| $+N$+2(|G| $+2q$) +2|ID| +result $= 8 + 4 * 40 + 20 + 2(40 + 2 * 20) + 2 * 64 + 8$ $=484$

compared with these schemes, the identity authentication process of the proposed scheme requires less calculation time.

DSP-CCAS uses the same password parameter value as MCPSP. The details are shown in Table 3. As shown in the process described in Figure 3, in the cross-domain request application phase, IRBA directly uses the identity as the public key. However, the public keys of the DSP-CCAS and MCPSP schemes consist of the user-calculated part of the public key and the part of the public key generated by KGC. To prevent replay attacks, the authentication server specifies a random number N during authentication. The user signs N and returns the signature and N to the authentication server. Once the identity authentication is passed, the authentication server uses the user ID for requesting the cross-domain authority of the user from the blockchain. This is a signature of the same user with the same aggregate signature size.

Table 5 presents the total communication cost of the above-analyzed schemes. As seen, IRBA needs 1200 bit, MCPSP needs 404 bit, and DSP-CCAS proposed in this paper needs 484 bit. Obviously, the proposed DSP-CCAS is significantly better than the IRBA communication cost in terms of communication cost, which is mainly affected by the security requirements of the algorithm itself. Under the same security level, the algorithm based on bilinear pairing requires a larger group size. And DSP-CCAS is slightly higher than the communication cost of MCPSP. This is because MCPSP is the same as IRBA, can only use the same parameters between security domains, and part of the private key is transmitted must through a dedicated security channel. However, unlike them, the scheme DSP-CAAS proposed in this paper (1) supports parameter differentiation of different systems. Although a certain communication cost has been added for this, the autonomy, privacy, and security of each security domain have been greatly improved. Each security domain can independently control the settings of its own authentication system security parameters without negotiating with other systems and share system security parameters. Moreover, the increased overhead is at an acceptable level. (2) Support the partial private keys to be transmitted through public channels. There is no need to build a dedicated transmission channel for partial private key transmission, and an open network can be used, such as a mobile data network, which reduces construction and operation and maintenance costs.

6. Conclusions

This article introduces a certificate-less cross-domain authentication scheme that supports parameter differentiation by improving the certificate-less signature algorithm. Through theoretical analysis and security classification, the correctness of the scheme is proved, and it can support different security domains using different master private key/-master public key pairs and supports a, which enhances the security of cross-domain authentication. Through comparative experiments, it is found that the overall verification time reaches 6.446 milliseconds, compared to the IRBA and MCPSP, in the case of cross-domain request access. That addressed the authority authentication issue in the current certificate-free cross-domain authentication scheme based on blockchain technology. The cost of one-time authentication communication is only 484 bits, which can meet the resource-constrained IoT data sharing environment.

Regarding the proposed scheme enabling blockchain-based massive data sharing, how to further reduce communication costs and design cross-domain authentication between heterogeneous domains that support different cryptosystems is worthy of further study in the future.

Data Availability

The data used to support the findings of this study are included within this article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers 61802086 and U20B2048), the Defense Industrial Technology Development Program (grant number 2020604B004), the Heilongjiang Provincial Natural Science Foundation of China (grant number LH2021F016), and the Fundamental

Research Funds for the Central Universities (grant number 3072021CF0608).

References

[1] J. Ding, T. R. Tang, Y. Zhang, and W. Chi, "Using intelligent ontology technology to extract knowledge from successful project in IoT enterprise systems," *Enterprise Information Systems*, pp. 1–27, 2021.

[2] S. Qu, L. Zhao, and Z. Xiong, "Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control," *Neural Computing and Applications*, vol. 32, no. 17, pp. 13505–13520, 2020.

[3] K.-H. Wang, C.-M. Chen, W. Fang, and T. Y. Wu, "On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags," *The Journal of Supercomputing*, vol. 74, no. 1, pp. 65–70, 2018.

[4] C.-M. Chen, B. Xiang, Y. Liu, and K. H. Wang, "A secure authentication protocol for Internet of Vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[5] E. K. Wang, J. Chen, Y. Peng, and L. Zhang, "Editorial: physical layer security and wireless access control (QSHINE 2017)," *Mobile Networks and Applications*, vol. 25, no. 1, pp. 1–3, 2020.

[6] K. Qiao, W. You, L. Wang, and H. Tang, "Data sharing scheme for 5G IoT based on blockchain," *Chinese Journal of Network and Information Security*, vol. 6, no. 4, pp. 45–55, 2020.

[7] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[8] C. Wang, D. Wang, G. Xu, and D. He, "Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0," *Science China: Information Sciences*, vol. 65, pp. 1–15, 2022.

[9] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017.

[10] S. Underwood, "Blockchain beyond bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016.

[11] Y. Qian, Y. Jiang, J. Chen et al., "Towards decentralized IoT security enhancement: a blockchain approach," *Computers & Electrical Engineering*, vol. 72, pp. 266–273, 2018.

[12] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," *Internet of Things*, vol. 1-2, no. 2, pp. 1–13, 2018.

[13] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: the case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 618–623, Kona, HI, USA, 2018.

[14] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: a lightweight scalable blockchain for IoT security and anonymity," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[15] Z. Li, D. Wang, and E. Morais, "Quantum-safe round-optimal password authentication for mobile devices," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[16] R. Reisman, "Blockchain serverless public/private key infrastructure for ADS-B security, authentication, and privacy," in *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019.

[17] D. Li, W. Peng, W. Deng, and F. Gai, "A blockchain-based authentication and security mechanism for IoT," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6, Hangzhou, China, 2018.

[18] J. Wang, L. Wu, K. K. R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1984–1992, 2020.

[19] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, Consultcd, 2008.

[20] H. U. A. N. G. Wei, "Potential risk and model construction of blockchain technology applied to derivatives market," *Financial Regulation Research*, vol. 2, pp. 97–111, 2019.

[21] Y. Yu, C. Taowei, and Z. Kun, "Data exchange model and application of certificate of origin based on blockchain technology," *E-Business Journal*, vol. 3, pp. 53–55, 2018.

[22] L. Yue, H. Junqin, Q. Shengzhi, and W. Ruijin, "Big data model of security sharing based on blockchain," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 117–121, Chengdu, China, 2017.

[23] C. Ellison and B. Schneier, "Ten risks of PKI: what you're not being told about public key infrastructure," *Computer Security Journal*, vol. 16, no. 1, pp. 1–7, 2000.

[24] C. Fromknecht, D. Velicanu, and S. Yakoubov, *Certcoin: A Namecoin Based Decentralized Authentication System 6.857 Class Project*, Unpublished class project, 2014.

[25] L. Axon and M. Goldsmith, "PB-PKI: a privacy-aware blockchain-based PKI," in *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, pp. 311–318, Madrid, 2017, SCITEPRESS - Science and Technology Publications.

[26] Z. H. U. Jian-ming and F. U. Yong-gui, "Supply chain dynamic multi-center coordination authentication model based on block chain," *Chinese Journal of Network and Information Security*, vol. 2, no. 1, pp. 27–33, 2016.

[27] Y. Chen, G. Dong, J. Bai, Y. Hao, F. Li, and H. Peng, "Trust enhancement scheme for cross domain authentication of PKI system," in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 103–110, Guilin, China, 2019.

[28] M. Al-Bassam, "SCPKI: a smart contract-based PKI and identity system," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pp. 35–40, New York, NY, USA, 2017.

[29] Z. Zhicheng, L. Lixin, and L. Zuohui, "Efficient cross-domain authentication scheme based on blockchain technology," *Journal of Computer Applications*, vol. 38, no. 2, pp. 316–320+326, 2018.

[30] W. Wang, N. Hu, and X. Liu, "BlockCAM: a blockchain-based cross-domain authentication model," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 896–901, Guangzhou, China, 2018.

[31] D. Liu, D. Li, X. Liu, L. Ma, H. Yu, and H. Zhang, "Research on a cross-domain authentication scheme based on consortium blockchain in V2G networks of smart grid," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, Beijing, China, 2018.

[32] M. Xiao-ting, M. Wen-ping, and L. Xiao-xue, "A cross domain authentication scheme based on blockchain technology," *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2571–2579, 2018.

[33] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical

access control in the IoT scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019.

[34] X. Jia, N. Hu, S. Su et al., "IRBA: an identity-based cross-domain authentication scheme for the Internet of Things," *Electronics*, vol. 9, no. 4, p. 634, 2020.

[35] H. Shu, P. Qi, Y. Huang, F. Chen, D. Xie, and L. Sun, "An efficient certificateless aggregate signature scheme for blockchain-based medical cyber physical systems," *Sensors*, vol. 20, no. 5, p. 1521, 2020.

[36] A. Meneghetti, M. Sala, and D. Taufer, "A new ECDLP-based PoW model," *Mathematics*, vol. 8, no. 8, article 1344, 2020.

[37] Y. X. Yan, L. Wu, W. Y. Xu, H. Wang, and Z. M. Liu, "Integrity audit of shared cloud data with identity tracking," *Security and Communication Networks*, vol. 2019, Article ID 1354346, 11 pages, 2019.

[38] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[39] X. Jiang, F. R. Yu, T. Song, Z. Ma, Y. Song, and D. Zhu, "Blockchain-enabled cross-domain object detection for autonomous driving: a model sharing approach," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3681–3692, 2020.

[40] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F. Y. Wang, "An overview of smart contract: architecture, applications, and future trends," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 108–113, Changshu, China, 2018.

[41] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, "A blockchain-based medical data sharing and protection scheme," *IEEE Access*, vol. 7, pp. 118943–118953, 2019.