Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041676

ARTICLE

BLECA: A Blockchain-Based Lightweight and Efficient Cross-Domain Authentication Scheme for Smart Parks

Fengting Luo, Ruwei Huang* and Yuyue Chen

School of Computer and Electronic Information, Guangxi University, Nanning, 530004, China

*Corresponding Author: Ruwei Huang. Email: ruweih@gxu.edu.cn

Received: 01 May 2023 Accepted: 13 September 2023 Published: 29 November 2023

ABSTRACT

Smart parks serve as integral components of smart cities, where they play a pivotal role in the process of urban modernization. The demand for cross-domain cooperation among smart devices from various parks has witnessed a significant increase. To ensure secure communication, device identities must undergo authentication. The existing cross-domain authentication schemes face issues such as complex authentication paths and high certificate management costs for devices, making it impractical for resource-constrained devices. This paper proposes a blockchain-based lightweight and efficient cross-domain authentication protocol for smart parks, which simplifies the authentication interaction and requires every device to maintain only one certificate. To enhance cross-domain cooperation flexibility, a comprehensive certificate revocation mechanism is presented, significantly reducing certificate management costs while ensuring efficient and secure identity authentication. When a park needs to revoke access permissions of several cooperative partners, the revocation of numerous cross-domain certificates can be accomplished with a single blockchain write operation. The security analysis and experimental results demonstrate the security and effectiveness of our scheme.

KEYWORDS

Cross-domain authentication; blockchain; smart parks; Certificate Authority (CA); distributed collaboration; Internet of Things (IoT)

1 Introduction

Intelligent devices have increased alarmingly with the development of the Internet of Things (IoT) and the advent of the 5G era [1]. Promoting the development of smart cities through IoT and cloud computing technology is a research hotspot nowadays [2]. Smart parks, as microcosms of smart cities, play an integral role in the era of innovation. Although extensive network infrastructure enables devices from various parks to interconnect, quite a few trust and security challenges have not been resolved [3]. Establishing secure communication for devices across different domains is still a crucial task.

Most of the existing identity authentication mechanisms are built upon the widely recognized Public Key Infrastructure (PKI) system, which involves a Certificate Authority (CA) [4]. The PKI system verifies the certificate to authenticate the identity of devices [5]. Traditional PKI-based cross-domain authentication schemes can be categorized into two types. The first type involves the

introduction of a trusted third party, namely the root CA, to serve as a trust center for all domains. However, if the root CA is compromised, the entire authentication system becomes paralyzed, leading to significant economic losses. The second type is inter-domain mutual authentication through the exchange of certificates between CAs. This authentication model results in overly complex authentication paths and escalates the cost of cross-domain cooperation. The traditional PKI-based cross-domain authentication schemes heavily rely on CAs, making them susceptible to the risk of single point failures [6].

The thriving blockchain technology offers a new solution to solve the security problem of IoT [7,8]. Blockchain possesses attributes like decentralization, tamper-proof, and data traceability, which can establish trust in an untrusted environment [9,10]. Blockchain has found widespread utility as a foundational technology for multi-party solutions [11,12]. The consortium blockchain is a kind of distributed ledger (DL) maintained by multiple cooperative peer nodes [13]. When multiple CAs collectively construct a consortium blockchain, they can realize decentralized certificate management and ensure reliable cross-domain trust transfer. This approach mitigates the risk of single point failures, and streamlines the complexity of cross-domain authentication processes [14,15].

In recent years, numerous scholars have leveraged consortium blockchain for cross-domain identity authentication [16–22]. Qiao et al. [23] devised an efficient cross-domain authentication scheme for the drone transportation industry by integrating consortium blockchain and PKI. Similarly, Rana et al. [24] employed blockchain technology to create a decentralized access control model for various healthcare sectors, enhancing interoperability among distinct industry systems. The solutions provided in [23,24] demand uniform identity management standards across all industries, which are not suitable for application in a range of industries with varying security levels. A smart park alliance is a cooperative mechanism aimed at fostering collaboration among various parks, promoting urban sustainability, and facilitating digital transformation. Members of smart park alliances may include government departments, urban planners, technology suppliers, businesses and industry groups, as well as social organizations. The security management of each park is typically different. Therefore, there is a need for a certificate management scheme that enables each park to issue and revoke cross-domain certificates in alignment with its specific security level. While many cross-domain authentication schemes address the requirement for autonomous certificate management by issuing cross-domain certificates to devices, the maintenance cost of multiple cross-domain certificates renders them impractical for resource-constrained devices. Moreover, large-scale enterprise parks typically oversee thousands of devices, and collaborative relationships between different organizations often undergo changes. In scenarios where an industrial park needs to terminate its cooperative relationship with other parks or replace a cooperative logistics company, existing solutions require substantial expenses to revoke cross-domain certificates for all relevant devices.

In order to solve the above problems, we proposed a blockchain-based lightweight and efficient cross-domain authentication scheme for smart parks (we call it BLECA). In contrast to the aforementioned cross-domain authentication schemes, BLECA provides a lightweight cross-domain authentication protocol and a practical certificate revocation mechanism specifically designed for smart parks. This solution not only guarantees cross-domain connectivity for resource-constrained devices but also enables each park to flexibly and cost-effectively revoke cross-domain identities according to its individual security requirements. The main contributions of this paper are as follows:

(1) We introduce a cross-domain authentication architecture based on the consortium blockchain, which consists of CA in each smart park. A smart contract for establish trust chains is deployed in the blockchain, allowing each park to dynamically adjust its partnerships.

(2) Drawing from the attributes of blockchain technology, this paper designs a low-cost, simplified cross-domain authentication and key agreement protocol. The protocol figures out the device's cross-domain certificate according to the device's intra-domain certificate and CA identity of the cross-domain park, enabling resource-constrained devices to maintain only one intra-domain certificate.
(3) We devise a complete certificate revocation mechanism for the dynamic partnership between parks. First, the device cancellation algorithm utilizes the trust chain to identify the relevant cross-domain certificates associated with the device that needs to be logged out, and revokes them synchronously, thereby enhancing system security. Secondly, the cross-domain certificate batch revocation feature is formatted in accordance with the requirement for terminating part of the partnerships. Only one blockchain write operation is required to revoke all relevant cross-domain certificates, which substantially reduces computational overhead and saves storage resources.

Organization: The remainder of this paper is organized as follows: Section 2 reviews the related work of cross-domain authentication based on blockchain. Section 3 introduces the system model and related theoretical knowledge. Section 4 elaborates on the design specifics of the proposed scheme, and Section 5 presents a security analysis. Section 6 includes extensive experiments and discussions to showcase the effectiveness of BLECA. Finally, Section 7 concludes the work.

2 Related Work

In this section, we begin by summarizing some basic knowledge of blockchain and smart contracts. Subsequently, we provide a comprehensive overview of recent research in blockchain-based identity authentication.

2.1 Blockchain and Smart Contract

Blockchain, initially proposed by Satoshi Nakamoto, is a distributed ledger system where data is managed in the form of blocks by a cluster of distributed computers [25]. Each block contains a block header and transaction data. The block header, including the hash value of the previous block, the hash value of the current block, and the timestamp, is used to link the previous block and calculate the hash value of the current block to ensure the continuity and tamper resistance. There are two common categories of blockchain: permissionless blockchain and permissioned blockchain. The first type refers to public blockchain, which allows any member to join freely. It always selects some competitive schemes as consensus protocol, such as Proof of Work (PoW), Proof of Stake (PoS), and Proof of Activity (PoA) [26–28]. The second category encompasses private blockchain and consortium blockchain. It only chooses trustful actors as peer nodes and usually adopts typical Crash Fault Tolerance (CFT) or Byzantine Fault Tolerance (BFT) consensus protocols. BFT is designed to ensure the fault tolerance and security of distributed systems by overcoming Byzantine faults like node failures, network delays, message loss, and malicious behaviors, enabling the system to reach a consensus in the face of these problems. Before transaction data is added to the blockchain, it must be confirmed by the consensus mechanism, ensuring the credibility and security of the data. Through the above-mentioned data organization method, the blockchain realizes the characteristics of decentralization, security, transparency and immutability, which can be used for credible reading and writing in distributed environments [29].

Practical Byzantine Fault Tolerance (PBFT) and Tendermint are consensus protocols based on the principles and ideas of BFT, which are currently widely used in the consortium blockchain.

Both protocols can tolerate no more than 1/3 of malicious nodes to ensure that the system has strong fault tolerance to Byzantine faults. PBFT consensus requires three rounds of voting, including the pre-prepare, prepare, and commit phases. The Tendermint consensus uses a "proposal-vote" mechanism that involves only two rounds of voting, namely the pre-vote stage and the pre-commit stage, reducing the complexity of message transmission and the consensus process. Tendermint offers higher throughput and lower latency, making it more suitable for large-scale network applications that demand high-performance consensus. Peer nodes of the consortium blockchain are verified entities, resulting in fewer instances of malicious behavior. Smart park alliances generally only allow qualified organizations to join, hence it is more appropriate to select the consortium blockchain with Tendermint consensus protocol for large-scale park alliances.

The users can perform customized programs stored on the blockchain. When certain transactions invoke the predetermined functions, the program will automatically run [30]. These customized programs go by different names on various blockchain platforms. For example, on the Ethereum platform, they are referred to as smart contracts, on the Hyperledger Fabric platform, they are called chaincode, and on the Tendermint platform, they are known as ABCI applications. Since Ethereum platform is the most widely used, we commonly use "smart contract" as the generic term for customized blockchain programs.

2.2 Cross-Domain Authentication Based on Blockchain

The research of identity authentication based on blockchain has garnered the attention of many research institutions [31,32]. In 2014, the Conner team of MIT proposed the first blockchain-based identity authentication system for certificate management [33]. Kshetri [34] asserted that blockchain offers significant advantages in facilitating identity management and access control for IoT, which can enhance IoT security. Rana et al. [35] discussed the significance of blockchain in the context of IoT and introduced a specialized blockchain-based architecture for IoT networks. Dave et al. [36] presented a blockchain-based video surveillance framework designed for smart home environments. This framework utilizes smart contracts to define authentication levels and access rules for video footage, ensuring that only authorized users can access surveillance recordings.

Zhou et al. [16] and Huang et al. [17] established consortium blockchains consisting of several PKI organizations and developed cross-domain authentication protocols. Nevertheless, these two schemes retain the process of traditional identity certification, resulting in relatively complex authentication paths. Zhao et al. [18] proposed a double-layer cross-domain authentication architecture composed of authentication server nodes and several internal blockchains, enhancing the scalability of the PKI system without altering the internal structure. However, for the authentication schemes proposed by [16–18] necessitate devices to manage multiple cross-domain certificates, rendering them unsuitable for lightweight devices with limited storage capacity. In [19], the authors designed a cross-domain authentication and session key agreement protocol, enabling devices to manage just one certificate for authentication across all other domains. Nonetheless, in this scheme, each domain is unable to flexibly define its certificate management standards, potentially posing a security risk to the system. In [20], researchers designed a thoroughly cross-domain authentication scheme based on blockchain, accommodating participants from domains with entirely different configurations. But its authentication process requires the blockchain to perform intensive signature verification and hash calculation. Furthermore, none of the aforementioned studies offer an effective certificate revocation mechanism.

Gu et al. [21] proposed a blockchain-based authentication and certificate revocation scheme. In this scheme, revoked certificate records are appended to the certificate revocation list stored in the DL for the query. However, the certificate revocation mechanism of this scheme is relatively intricate, and the accumulation of revocation records can significantly impact authentication efficiency. Wang et al. [22] introduced a multi-CA authentication model based on blockchain. They designed a cross-domain certificate revocation mechanism using the Poisson distribution of the cross-domain access list. However, it requires substantial system resources to execute device revocation, and the related cross-domain certificates cannot be revoked synchronously.

While these blockchain-assisted cross-domain authentication schemes have achieved reliable authentication across multiple domains to some extent, they have not addressed the issue of certificate management for resource-constrained devices in multi-domain environments with varying security management settings. Furthermore, the existing certificate revocation mechanisms have not met the requirement for synchronously revoking all certificates when a device is deregistered and the bulk revocation of cross-domain certificates due to changes in inter-domain collaboration. Therefore, the current authentication schemes are not suitable for dynamic collaborative scenarios among smart parks with different security levels.

Novelty: The principal novel aspect of this article is that an effective blockchain-based cross-domain authentication scheme is proposed to meet the collaboration requirements among smart parks with varying security levels. Specifically, the distinctions from existing solutions are as follows: 1) The smart contract for managing cross-domain trust chains facilitates each park to flexibly adjust its collaborative relationships. 2) The process of cross-domain authentication and key negotiation is formulated, allowing each park to issue cross-domain certificates according to its security management regulations without adding to the certificate management burden of resource-constrained devices. 3) Utilizing the cross-domain trust chain as a foundation, we present an effective certificate revocation mechanism consisting of three algorithms: device cancellation, single cross-domain certificate revocation, and cross-domain certificate batch revocation. This mechanism further guarantees that each park can efficiently carry out certificate revocation tasks in different scenarios aligning with its own security standards.

3 Preliminaries

This section provides a summary of symbols used in the BLECA scheme, as presented in Table 1. Subsequently, the system model is described, followed by an introduction to the relevant technologies employed in this scheme.

Table 1: Symbol description of the BLECA scheme

Symbol	Description
CA_i	Certificate authority of *Park i*
$BPCA$	Consortium blockchain composed of CAs
AS_i/D_i	Authentication server/device of *Park i*
pk_X/sk_X	Public key/private key of entity X
Enc_X/Dec_X	Encryption/decryption algorithm
$Hash\,(m)$	Hash algorithm

(Continued)

Table 1 (continued)

Symbol	Description
$\delta = Sign\,(sk, msg)$	Signing algorithm
$Verify\,(pk, msg, \delta)$	Verification algorithm

3.1 System Model

The cross-domain authentication model of the proposed scheme is shown in Fig. 1.

(1) *BPCA* represents the consortium blockchain composed of CA from each park. Every CA node with the complete ledger can independently and transparently audit and verify blockchain transactions.
(2) CA_i represents the authority node issuing certificates for the devices of *Park i*. CA_i is also responsible for maintaining the trust chain of *Park i* and managing the certificates of devices.
(3) AS_i represents the authentication server node of *Park i*, which is responsible for receiving and processing the authentication request. AS_i is authorized to issue cross-domain certificates for request devices according to the trust chain. Furthermore, AS_i determines the abnormal behavior of a cross-domain device using predefined behavior criteria and recognition algorithms, and subsequently revokes the device's cross-domain certificate.
(4) D_i represents the smart device under the jurisdiction of *Park i*.

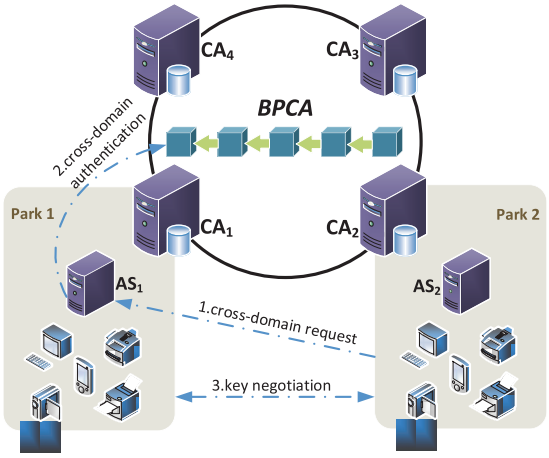

Figure 1: System model

In the proposed scheme, each park can autonomously maintain its own cross-domain trust relationships. The CA of each campus establishes its cross-domain trust chain according to its specific operational requirements. The trust chains of all domains collectively form an editable directed graph, as illustrated in Fig. 2. The vertexs in this graph represent the parks that join the consortium blockchain. Each park can only maintain its indegree edges. In Fig. 2, the value $Hash\,(CA_2,\ CA_1)$ and its status $TrustState$ = "*issue*" indicate that *Park* 2 has obtained cross-domain access authorization for *Park* 1. $TrustState$ = "*revoke*" means that the channel for *Park* 2 to access *Park* 1 has been closed.

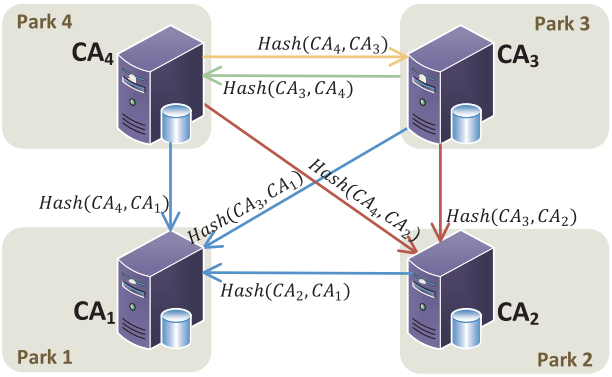

Figure 2: Editable cross-domain directed graph

3.2 Related Technologies

3.2.1 Computational Hard Assumptions

While constructing the scheme, we rely on two computational difficulties assumptions. The detailed definition is presented as follows:

Elliptic curve discrete logarithm (ECDL) assumption: Given an element $A \in \mathbb{G}$. It is impossible for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ to obtain $A = a \cdot G$, where $a \in Z_q^*$.

Elliptic curve computational Diffie-Hellman (ECCDH) assumption: Let $A = a \cdot G$, $B = b \cdot G$, where $A, B \in \mathbb{G}$. It is impossible for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ to obtain $a \cdot b \cdot G$.

3.2.2 Hash Function

A secure hash function [37] should possess the following three properties, where X represents the sets of all possible messages and Y represents the sets of all potential hash values.

Preimage stability: Given hash function $h : X \to Y, y \in Y$. It is difficult to find out $x \in X$ to make the equation $h(x) = y$ hold.

The second preimage stability: Given hash function $h : X \to Y, x \in X$. It is difficult to find out $x' \in X$ to make the equation $h(x') = h(x)$ hold, where $x' \neq x$.

Collision-resistance: Given hash function $h : X \to Y$. It is difficult to find out $x, x' \in X$ to make the equation $h(x') = h(x)$ hold, where $x' \neq x$.

3.2.3 Signature Scheme

A generic digital signature scheme consists of a set of probabilistic polynomial-time algorithms $SIG = (Gen, Sign, Verify)$ [38]. The specific definition is provided below:

Key generation algorithm $(pk, sk) \leftarrow Gen(1^{\lambda})$: Safety parameters λ is the input. The public key pk used for signature verification and the private key sk used for signing are the output.

Signing algorithm $\delta \leftarrow Sign(sk, msg)$: Use sk to sign msg to generate a signature δ.

Verification algorithm $result \leftarrow Verify(pk, msg, \delta)$: The signature δ, message msg and public key pk are the input parameters. If δ is the valid signature of msg, set $result = 1$, otherwise set $result = 0$.

Correctness. For every λ, $(pk, sk) \leftarrow Gen\,(1^{\lambda})$ and msg, if $Verify\,(pk, msg, \delta = Sign\,(sk, msg)) = 1$, proving that δ is a valid signature of msg.

Security. The security of the elliptic curve-based signature scheme relies on the computational complexity of ECDL assumption.

4 Design of the Proposed Scheme

4.1 System Initialization

The system initialization phase is carried out by all domains, as described below:

(1) **Basic initialization:** Every domain picks an elliptic curve $E\left(\mathbb{F}_p\right)$ defined on the finite field $\left(\mathbb{F}_p\right)$, where the generator is P and the prime order is q. A secure hash function $Hash : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ is selected by all the parks. The system parameters and the public key list of *CAs* are published in the DL. Then, every CA must process the registration application from the authentication server (AS) under its jurisdiction. Finally, the smart contract supporting the query of CA and AS nodes' information is deployed in the DL.

(2) **Initialization of trust chain:** Every CA sets its indegree edges. This process is called initialization of trust chain, as shown in Algorithm 1, where $self\_CA$ is the identity ID of the initiator, and $trust\_CA\,[n]$ are the partners of $self\_CA$. $BPCA$ stores the $TrustRoot = Hash\,(\,trust\_CA\,[x]\,, self\_CA)$ as the trust root and sets $TrustRoot$'s status $TrustState =$ "*issue*", where $TrustState =$ "*issue*" means that $trust\_CA\,[x]$ has gained the trust of $self\_CA$. Based on the initialization and update of the directed graph, every CA synchronously records the vertexs pointed to by its outdegree edge as the cross-domain list $cross\_CA\,[n]$ in the local database.

Algorithm 1: Initialization of Trust-Chain

Input: N_i , $Sign\left(sk_{self\_CA}, N_i\right)$, $self\_CA$, $trust\_CA\,[n]$
Output: *Implement result*
if $Verify\left(pk_{self\_CA}, N_i, Sign\left(sk_{self\_CA}, N_i\right)\right) == 0$ **then**
    return *error*
else
    for $(x = 0; x < n; x++)$
        $TrustRoot = Hash\,(\,trust\_CA\,[x]\,, self\_CA)$
        $TrustState =$ "*issue*"
        Store $(TrustRoot|\ TrustState)$
    return *success*

4.2 Device Registration

The smart device D_i is registered through the authentication server AS_i. The specific steps for identity registration are as follows:

Step1: $D_i \rightarrow AS_i : \left\{Enc_{pk_{AS_i}}\left(D_i, pk_{D_i}\right)\right\}$.

D_i generate a private random number x_i as its local private key and calculates its public key $pk_{D_i} = x_i \cdot P$. Then, D_i downloads $AS_i's$ public key pk_{AS_i}, and encrypt the message with pk_{AS_i}. Finally, D_i sends the encrypted message to AS_i to obtain the licensed certificate.

Step2: $AS_i \rightarrow CA_i : \left\{D_i, pk_{Di}, N_i, Sign\left(sk_{AS_i}, N_i\right)\right\}$.

After receiving the registration request from D_i, AS_i decrypts the message with its private key. AS_i checks the registration status of D_i and determines whether D_i is a legal user. If the verification is successful, AS_i send the message to the certification authority CA_i through a secure channel to apply for $D_i's$ certificate. Otherwise, the identity registration of D_i must be terminated.

Step3: $BPCA \rightarrow AS_i \rightarrow D_i : \left\{Cert_{D_i}\right\}$.

If the identity signature of AS_i is valid, CA_i issues intra-domain certificate $Cert_{D_i} = \left(CA_i, D_i, pk_{D_i}, ET_{D_i}\right)$ for D_i, where ET_{D_i} is the expiration time of the certificate. Besides, the hash value and the state of $D_i's$ certificate is written in the DL as the form of key-value pair (Hash $(Cert_{Di})$ | Cert$State$). Subsequently, CA_i return $D_i's$ certificate to AS_i. Eventually, AS_i encrypts $D_i's$ certificate in the local database and sends the certificate to D_i.

4.3 Cross-Domain Authentication and Session Key Agreement

Devices can apply for cross-domain certificates from the authentication server nodes of cooperative parks. For example, if a device D_A of *Park A* wants to obtain a cross-domain certificate issued by *Park B*, AS_B takes the identity of CA_B as a new parameter and combines it with the $D_A's$ intra-domain certificate to form a cross-domain certificate $CrossCert_{D_{A\rightarrow B}} = \left(CA_B, CA_A D_A pk_{D_A}, ET_{D_A}\right)$ for D_A. The specifics of function for issuing cross-domain certificates can be referred to Algorithm 2.

Algorithm 2: Cross-Domain Certificate Issuance

Input: $N_i, Sign\left(sk_{AS_B}, N_i\right), CA_B, Cert_{D_A} = (CA_A, D_A, pk_{D_A}, ET_{D_A})$
Output: *Implement result*
if $Verify\left(pk_{AS_B}, N_i, Sign\left(sk_{AS_B}, N_i\right)\right) == 0$ **then**
    return *error*
else
    $CrossCertKey = Hash\left(CA_B, Cert_{D_A}\right)$
    $TrustRoot = Hash\left(CA_A, CA_B\right)$
    $CrossCertState =$ "*issue*"
    Store $(CrossCertKey|\ TrustRoot, CrossCertState)$
    return *success*

When the device D_A from *Park A* need access to the resources from *Park B*, the process of cross-domain authentication and session key agreement is shown in Fig. 3. The specific steps are outlined below:

Step1: $D_A \rightarrow AS_B : \{request\}$.

D_A sends a cross-domain access request to AS_B.

Step2: $AS_B \rightarrow D_A : \{N_i\}$.

After receiving the request, AS_B generates a random number N_i and records the current timestamp t_N. Then, AS_B sends N_i to D_A.

Step3: $D_A \rightarrow AS_B : \left\{\text{Cert}_{D_A} = \left(CA_A, D_A,\ pk_{D_A}, ET_{D_A}\right), Sign\left(sk_{D_A}, N_i\right)\right\}$.

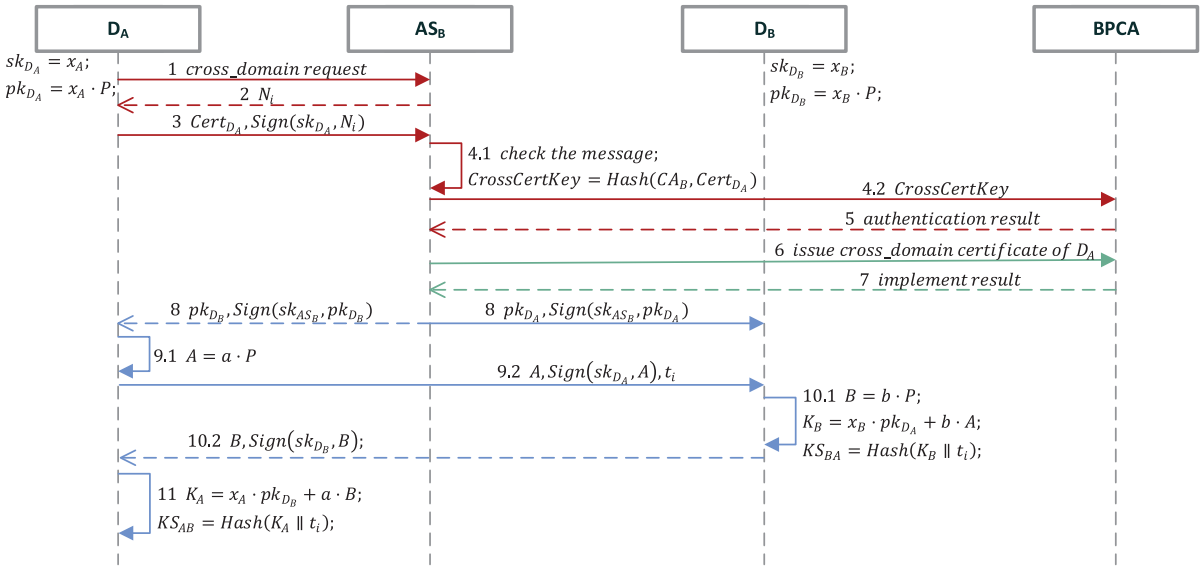

Figure 3: The process of cross-domain authentication and session key agreement

D_A uses its private key to encrypt N_i to generate a signature message. The signature and D'_As certificate is sent to AS_B.

Step4: $AS_B \rightarrow BPCA : \{CrossCertKey\}$.

First, AS_B checks the freshness of the message. If $\left|t'_N - t_N\right| < \Delta t$, where t'_N is the current timestamp, AS_B utilizes D'_As public key to verify D'_As signature. If the signature is valid, AS_B will calculate the hash value $CrossCertKey = Hash\left(CA_B, \text{Cert}_{D_A}\right)$ of cross-domain certificate and send it to $BPCA$ to verify D'_As cross-domain permission.

Step5: $BPCA \rightarrow AS_B : \{result\}$.

Algorithm 3 depict the cross-domain authentication function deployed in the blockchain. It will return the result to AS_B.

(a) If the query result is empty, it means that D_A does not have the cross-domain certificate of *Park B*. So AS_B needs to issue cross-domain certificates for D_A, i.e., Steps 6–7.
(b) If the status of *CrossCertKey* is *revoke* or the status of *TrustRoot* is *revoke* or *TrustRoot* does not exist, cross-domain authentication fails.
(c) If the status of *CrossCertKey* is *issue* and the status of *TrustRoot* is *issue*, it means that cross-domain authentication is successful and D_A can access the resources of *Park B* then. After verification, the session key is figured out for further reliable communication. Steps 8–11 describe a session key agreement protocol based on ECCDH assumption.

Algorithm 3: Cross-Domain Authentication

Input: *CrossCertKey*
Output: *Authentication result*
if *CrossCertKey* == *nil* **then**
//"*CrossCertKey* == *nil*" means that the query result is empty
return "CrossCertKey *does not exist*"

(Continued)

Algorithm 3 (continued)

else

if *CrossCertState* == "*revoke*" **then**

return "*Authentication fail*"

else if *TrustRoot* == *nil* ***or*** *TrustState* == "revoke" **then**

return "*Authentication fail*"

else *CrossCertState* == "*issue*" ***and*** *TrustState* == "*issue*"

then

return "*Authentication success*"

Step6: $AS_B \rightarrow BPCA : \{CA_B, Cert_{D_A}, N_i, Sign(sk_{AS_B}, N_i)\}$.

First, AS_B performs hash calculations $TrustRoot = Hash(CA_A, CA_B)$ and $CertKey_{D_A} = Hash(CA_A, D_A, pk_{D_A}, ET_{D_A})$ and send them to $BPCA$ to query their validity. If $TrustRoot$ is not licensed or $CertKey_{D_A}$ is not licensed, the application for a cross-domain certificate for D_A is terminated. Otherwise, the AS_B sends a request publishing $D_A's$ cross-domain certificate to $BPCA$.

Step7: $BPCA \rightarrow AS_B : \{result\}$.

After receiving the request, $BPCA$ will automatically execute Algorithm 2. If the signature of AS_B is valid, $BPCA$ sets $CrossCertState$ = "*issue*" and stores the tuple ($CrossCertKey|\ TrustRoot$, $CrossCertState$) into DL as the form of key-value pair, where $CrossCertKey = Hash(CA_B, Cert_{D_A})$, $TrustRoot = Hash(CA_A, CA_B)$.

Step8: $AS_B \rightarrow D_A : \{pk_{D_B}, Sign(sk_{AS_B}, pk_{D_B})\}$; $AS_B \rightarrow D_B : \{pk_{D_A}, Sign(sk_{AS_B}, pk_{D_A})\}$.

Receiving the successful authentication result returned by $BPCA$, AS_B uses its private key to individually sign the public keys of both D_A and D_B. Then, AS_B sends the signature $Sign(sk_{AS_B}, pk_{D_B})$ to D_A and sends $Sign(sk_{AS_B}, pk_{D_A})$ to D_B. After receiving the message, D_A and D_B verify its correctness using $AS_B's$ public key . If the verification is successful, proceed with Steps 9–11.

Step9: $D_A \rightarrow D_B : \{A, Sign(sk_{D_A}, A), t_i\}$.

D_A selects a private random number $a \in Z_q^*$ and records the current timestamp t_i. Subsequently, D_A calculates $A = a \cdot P$ and signs A with its private key. In the end, the message is sent to D_B.

Step10: $D_B \rightarrow D_A : \{B, Sign(sk_{D_B}, B)\}$.

If $|t_i' - t_i| < \Delta t$, where t_i' is the current timestamp, D_B utilizes $D_A's$ public key to verify whether the signature is valid. If the identity of D_A is legal, D_B selects a private random number $b \in Z_q^*$ and makes $B = b \cdot P$, $K_B = x_B \cdot pk_{D_A} + b \cdot A$, where x_B is the private key of D_B. Then, the hash calculation $KS_{BA} = Hash(K_B \parallel t_i)$ is performed, and the result KS_{BA} is regarded as the session key between D_A and D_B. Eventually, D_B sends the message $(B, Sign(sk_{D_B}, B))$ to D_A.

Step11: $D_A : \{KS_{AB}\}$.

If the identity signature of the D_B is valid, D_A performs the calculation $K_A = x_A \cdot pk_{D_B} + a \cdot B$, where x_A is the private key of D_A. Last, the value $KS_{AB} = Hash(K_A \parallel t_i)$ is recognized as the session key.

Correctness: From the above equations, we can get Eq. (1).

$$K_A = x_A \cdot pk_{D_B} + a \cdot B = x_A \cdot x_B \cdot P + a \cdot b \cdot P = x_B \cdot (x_A \cdot P) + b \cdot (a \cdot P)$$
$$= x_B \cdot pk_{D_A} + b \cdot A = K_B \tag{1}$$

Thus, we obtain $KS_{AB} = KS_{BA}$.

4.4 Certificate Revocation

4.4.1 Device Cancellation

When a device needs to be canceled, the cross-domain certificates associated with the device should be revoked simultaneously to enhance system security. Taking the device D_i as an example, when CA_i receives the cancellation request from AS_i, if $AS_i's$ identity signature is correct, CA_i invokes the device cancellation contract whose algorithm is shown in Algorithm 4. The status of $D_i's$ certificate is updated as $CertState$ = "*revoke*". Then, *BPCA* revokes all cross-domain certificates associated with D_i. *BPCA* performs the hash computations $CrossCertKey = Hash\left(cross\_CA[x], Cert_{D_i}\right)$, $TrustRoot = Hash\left(CA_i, cross\_CA[x]\right)$ and updates the information $(CrossCertKey|\ TrustRoot,\ CrossCertState)$, where $CrossCertState$ = "*revoke*".

Algorithm 4: Device Cancellation

Input: $N_i, Sign\left(sk_{CA_i}, N_i\right), Cert_{D_i} = \left(CA_i, D_i, PK_{D_i}, ET_{D_i}\right), cross\_CA[n]$
Output: *Implement result*
if $Verify\left(pk_{CA_i}, N_i, Sign\left(sk_{CA_i}, N_i\right)\right) == 0$ **then**
    return *error*
else
    $CertKey = Hash\left(Cert_{D_i}\right)$
    if $CertKey == nil$ ***or*** $CertState$ == "*revoke*" **then**
    //"$CertKey == nil$" means that the query result is empty
        return *error*
    else
        $CertState$ = "*revoke*"
        Update $(CertKey|\ CertState)$
        for $(x = 0; x < n; x++)$
            $CrossCertKey = Hash\left(cross\_CA[x], Cert_{D_i}\right)$
            if $CrossCertKey == nil$ ***or*** $CrossCertState$ == "*revoke*" **then**
                continue
            else
                $TrustRoot = Hash\left(CA_i, cross\_CA[x]\right)$
                $CrossCertState$ = "*revoke*"
                Update $(CrossCertKey|\ TrustRoot,\ CrossCertState)$
        return *success*

4.4.2 Single Cross-Domain Certificate Revocation

The revocation of a single device certificate typically occurs in the following two situations. If D_A no longer needs to access the equipment resources of *Park B*. To protect the cross-domain certificate from malicious attackers, D_A can request AS_B to revoke its cross-domain certificate. The second situation arises when *Park B* determines that D_A has violated its management regulation, it must revoke the cross-domain certificate of D_A. Algorithm 5 describes the mechanism for revoking a single cross-domain certificate.

Algorithm 5: Single Cross-Domain Certificate Revocation

Input: N_i, $Sign\left(sk_{AS_B}, N_i\right)$, $CrossCert_{D_{A\to B}} = \left(CA_B, CA_A, D_A, PK_{D_A}, ET_{D_A}\right)$
Output: *Implement result*
if $Verify\left(pk_{AS_B}, N_i, Sign\left(sk_{AS_B}, N_i\right)\right) == 0$ **then**
    return *error*
else
    $CrossCertKey = Hash\left(CrossCert_{D_{A\to B}}\right)$
    if $CrossCertKey == nil$ ***or*** $CrossCertState$ $==$ "*revoke*" **then**
    //"$CrossCertKey == nil$" means that the query result is empty
        return *error*
    else
        $TrustRoot = Hash\left(CA_A, CA_B\right)$
        $CrossCertState =$ "*revoke*"
        Update $(CrossCertKey|\ TrustRoot, CrossCertState)$
        return *success*

4.4.3 Cross-Domain Certificate Batch Revocation

According to Section 4.3, the *TrustRoot*, based on the cross-domain directed graph, is set as an attribute of the cross-domain certificate of the device. Suppose a park needs to disable the cross-domain access for some cooperative objects, it can efficiently revoke all relevant cross-domain certificates by updating its trust chain. The batch revocation function for cross-domain certificates is shown in Algorithm 6, where $revoke\_CA\ [n]$ are the objects whose cross-domain permission will be revoked by $self\_CA$.

Algorithm 6: Cross-Domain Certificate Batch Revocation

Input: N_i, $Sign\left(sk_{self\_CA}, N_i\right)$, $self\_CA$, $revoke\_CA\,[n]$
Output: *Implement result*
if $Verify\left(pk_{self\_CA}, N_i, Sign\left(sk_{self\_CA}, N_i\right)\right) == 0$ **then**
    return *error*
else
    for $(x = 0; x < n; x++)$
        $TrustRoot = Hash\left(\ revoke\_CA\,[x], self\_CA\right)$
        if $TrustRoot == nil$ ***or*** $TrustState ==$ "*revoke*" **then**
            return *error*
        else
            $TrustState =$ "*revoke*"
            Update $(TrustRoot|\ TrustState)$
    return *success*

5 Security Analysis

This section will provide a detailed explanation of how the BLECA scheme satisfies several standard security requirements for identity authentication.

5.1 Distributed Denial of Service (DDoS) Attack

DDoS Attack refer to multiple attackers located in different locations simultaneously launching attacks on one or more targets, or an attacker taking control of multiple machines located in different locations and utilizing these machines to simultaneously attack victims. This scheme uses consortium blockchain with Tendermint consensus mechanism as a distributed database. For every write transaction, all peers will receive them and save the data in the DL. Unlike traditional identity authentication solutions that rely on a central node, this scheme operates differently. Even if some nodes come under attack, the blockchain network can continue to function normally as long as the number of failed nodes is less than 1/3. When the paralyzed nodes is restored, it can retrieve the complete ledger from other nodes and resume its role as a normal node with a cross-domain authentication function. If the attacker tries to manipulate the system, it needs to control the computing power of $\frac{C_{\mathcal{A}}}{C_{\mathcal{S}}} \geq \frac{2}{3}$, where $C_{\mathcal{A}}$ and $C_{\mathcal{S}}$ represent the computing power of the attacker and the total computing power of the blockchain system, respectively. It is difficult for an attacker to possess such a huge computing power to disrupt Tendermint consensus protocol. The underlying blockchain technology and cryptographic primitives ensure the immutability of any identity information stored in the DL.

5.2 Replay Attack

The authentication scheme of this paper adopts the question-response handshake mode, which integrates a random number along with a timestamp to ensure timeliness during message transmission. Receiving the cross-domain request of a device, the authentication server node will generate a random number N_i and record the current timestamp t_N. If the message delivery time surpasses the system-defined threshold, it is determined that the message has lost its freshness. Even if an attacker intercepts and attempts to resend the message, the timeliness of N_i will lead to failure, making it is resistant to replay attacks.

5.3 Man-in-the-Middle (MITM) Attack

The man-in-the-middle attack aims to intercept regular network traffic, sniff, and tamper with data without the knowledge of both parties. Suppose the attacker intercepts the random number N_i from AS_B during the cross-domain authentication process, but it does not have D'_As private key so that it can not generate D'_As identity signature. If the attacker attempts to sign the random number N_i using its private key and sends this signature along with its certificate to AS_B, it will fail authentication since the attacker is not a legitimate user in the DL. Successful cross-domain authentication requires the correct signature and a valid certificate. Therefore, this scheme is effective in thwarting man-in-the-middle attacks.

5.4 Insider Attack

Authorized users within the network can query the information about certificates stored in the DL. An insider attack occurs when an authorized user within the network behaves as an attacker by leaking certificates, thereby posing a threat to the system's security. In our scheme, the data stored in the DL is the certificate's hash value rather than the certificate's metadata. Due to the hash function properties described in Section 3.2, it is virtually impossible for an attacker to deduce certificate metadata from the hash value. As a result, this scheme exhibits resistance to internal attack.

5.5 Perfect Forward Secrecy

A secure session key agreement protocol is designed for communication between two devices. The value a and b are secret keys selected by both sides of the communication, so attackers cannot obtain these values. Attackers would need to break the ECCDH assumption to figure out the session key. The exchanged messages A and B are one-time keys, which ensure the forward secrecy of the session. In the event of a leak of the private keys of both parties, the session key KS_{AB}, KS_{BA} will not be compromised.

6 Performance Evaluation

6.1 Security and Functionality Performance Comparison

In this subsection, we choose some critical features to compare the scheme BLECA with Zhou et al. [16], Zhao et al. [18], Gu et al. [21], and Wang et al. [22]. As shown in Table 2, it is evident that BLECA is the only scheme that supports all of these features.

Table 2: Security and functionality system features comparison

	Papers				
Features	[16]	[18]	[21]	[22]	BLECA
DDoS attack resistant	✓	✓	✓	✓	✓
Replay attack resistant	✓	✓	✓	✓	✓
MITM attack resistant	✓	✓	✓	✓	✓
Insider attack resistant	✓	✓	✓	✓	✓
Forward secrecy	×	×	×	×	✓
Editable cross-domain directed graph	×	×	×	×	✓
Cross-domain certificate independence	✓	✓	×	✓	✓
Device'*s* global cancellation	×	×	✓	×	✓
Certificates batch revocation	×	×	×	×	✓
Number of *Device*'*s* certificates	n + 1	n + 1	1	m + 1	1

Note: ✓: The scheme supports this feature; ×: The scheme does not support this feature.

The aforementioned five schemes meet the basic security requirements, such as resisting DDoS attacks, replay attacks, MITM attacks, and internal attacks. Besides, BLECA provides a secure session key agreement mechanism to ensure the forward secrecy in communication. In the scheme proposed by Gu et al. [21], a device is only required to maintain one certificate, but this certificate also serves as a cross-domain certificate for other domains. As a result, this scheme is not suitable for scenarios where each park has different certificate management standards. In contrast, the cross-domain certificates in [16,18,22] are independent. However, devices in the studies [16,18] need to preserve $n+1$ certificates to access n parks, and those in [22] needs to retain $m+1$ certificates to connect m devices simultaneously. It is worth highlighting that BLECA requires only a single certificate for each device, which illustrates that an increase in the number of cooperation parks or communication devices will not elevate the certificate management overhead for the device. More importantly, BLECA offers several flexible features, including an editable cross-domain directed graph, global cancellation of devices, and batch revocation of cross-domain certificates, which enhance the scalability, security, and efficiency of the system.

6.2 Experiment Performance

Chen et al. [39] conducted performance evaluations of certain cryptographic operations used in the relevant protocols on both smartphones and desktop computers to simulate resource-constrained terminal devices and authentication servers. The smartphone is equipped with Android 10.0 system, HUAWEI Kirin 980 2.6 GHz CPU, and 6 GB RAM, while the computer has Windows 10 system, Intel (R) Core (TM) i5-7300 HQ 2.5 GHz CPU, and 8 GB RAM. The computational capabilities of the current devices from smart parks are similar to these types of devices. Therefore, when evaluating the computation costs of different solutions, we refer to the measured times of some cryptographic operations reported by Chen et al. [39]. Table 3 summarizes some of their experimental results. Most computation and communication overhead are incurred during the authentication period, so we ignore the cost of system initialization and certificate issuance. Our primary focus is on comparing the cost of cross-domain authentication. Table 4 provides a summary of the comparison between BLECA and the other four schemes. The computation cost of BLECA is equal to that of the scheme [16,18,21]. Compared with the scheme [22], BLECA places a slightly lower computational burden on devices. Furthermore, our scheme requires one less communication interaction than the other four schemes, which reduces communication costs. It can be seen that our scheme is the most lightweight among the mentioned schemes.

Table 3: Execution time of cryptographic operations

Operation	Description	Device	Server
T_h	Hash algorithm	0.0038 ms	0.0018 ms
T_s	Signing algorithm	3.8746 ms	1.9353 ms
T_v	Verification algorithm	1.2487 ms	0.0743 ms

Table 4: Computation and communication costs comparisons

Reference	Device	Server	Device + Server	Total cost	Interaction round
[16]	T_s	$T_v + T_h$	$T_s + T_v + T_h$	3.9507 ms	4
[18]	T_s	$T_v + T_h$	$T_s + T_v + T_h$	3.9507 ms	4
[21]	T_s	$T_v + T_h$	$T_s + T_v + T_h$	3.9507 ms	4
[22]	$T_s + T_h$	T_v	$T_s + T_v$	3.9527 ms	4
BLECA	T_s	$T_v + T_h$	$T_s + T_v + T_h$	3.9507 ms	3

For reference, we categorize the operations involved in the contracts of *BLECA* into two types: Query operations and Write operations. Write operations involve the creation and storage of new data into the blockchain network, which changes the ledger of every peer node. Query operations, on the other hand, involve retrieving data already stored in the blockchain, without modifying the distributed ledger. As outlined in the security analysis in Section 5.1, the consensus mechanism initiated by write operations ensures that the certificates of devices can be safely written into the ledgers of all parks without being tampered with. The query operations of blockchain provides a trust transfer service for the authentication server to verify the identities of cross-domain devices, and ensures that the cross-domain authentication service will not collapse due to the failure of some CA nodes. Write operations within this scheme encompass trust chain initialization, device registration, cross-domain certificate

issuance, and certificate revocation. Since the code calculation overhead of these write operations is similar, and all write operations within the same blockchain network carry the same time cost for executing the consensus protocol. We selected the smart contract for single cross-domain certificate revocation (Algorithm 5) as an experimental sample for write operations. Algorithm 3, described in Section 4.3, represents a query operation and serves as the core of the cross-domain authentication protocol. Therefore, this contract was chosen as the experimental sample of the query operation.

We conducted four sets of experiments to assess the performance of query and write operations across different numbers of domains (4, 8, 12, and 16). We employed the thread group of Apache-Jmeter-5.5 to simulate client connection requests and collected the throughput and average delay data using performance listener of Jmeter. Initially, we measured the average latency for both write and query operations with one request, as shown in Table 5. Then, we focused on performance metrics by increasing the number of concurrent requests within 1 s. For query operations, the number of requests increases from 100 to 1000, and the test result of average latency is shown in Fig. 4. In the case of write operations, the number of requests increases from 50 to 500, and the corresponding results are presented in Figs. 5a and 5b. Each condition was tested ten times, and the average values were recorded for the results.

Table 5: Average delay of a single operation

Operation	4 peer	8 peer	12 peer	16 peer
Write	523 ms	595 ms	623 ms	742 ms
Query	3.22 ms	3.44 ms	3.39 ms	3.61 ms

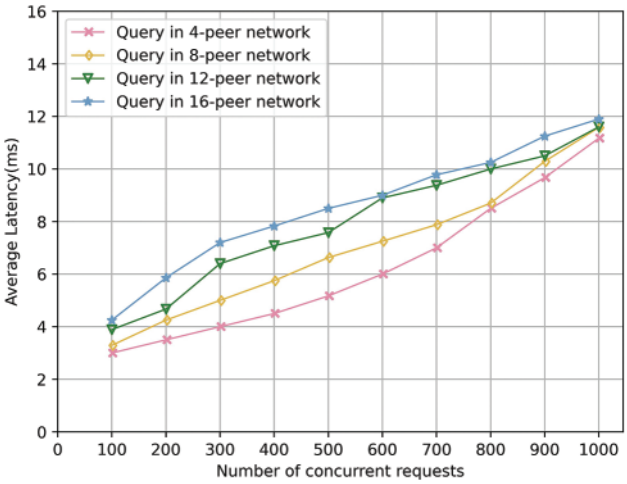

Figure 4: Average latency of query operation

Performance of query operations: Table 5 and Fig. 4 reveal that the average delay of query operations is not affected by the number of peers. Similar findings were also testified in the experiments conducted by [20,22]. This observation shows that scaling up the number of parks within the collaborative alliance has no discernible impact on the performance of the cross-domain authentication protocol. Fig. 4 shows that the average latency for query operations exhibits a gradual increase as the number of concurrent requests rises. Specifically, with every additional 100 query requests, the average

delay increases by a mere 0.88 milliseconds. It can be predicted that the time required for queries, even when numerous devices are simultaneously requesting cross-domain authentication, remains well within the acceptable limits for smart park construction. Based on these results, it is evident that the cross-domain authentication protocol of this scheme demonstrates outstanding performance and scalability.

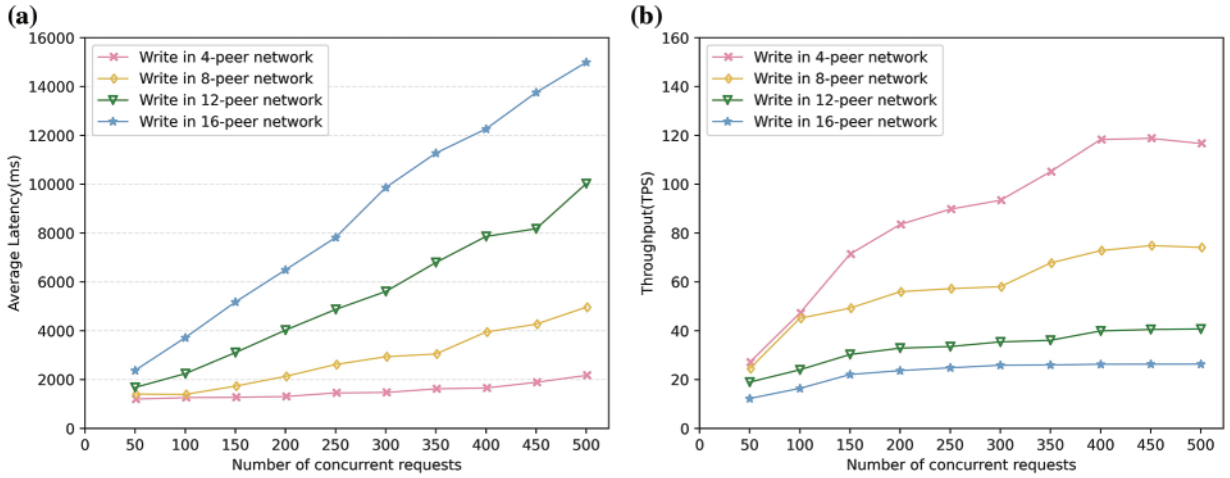

Figure 5: (a) Average latency of write operation. (b) Throughput of write operation

Performance of write operations: As depicted in Table 5, Figs. 5a, and 5b, as the number of peer nodes increases, the average latency rises, and the maximum throughput decreases. With the increase of concurrent requests, the average delay will also increase, while the throughput will gradually increase at the beginning and then tend to be steady with a maximum value.

The scheme [16,18] does not provide cross-domain certificate revocation service. Next, we will compare the cross-domain certificate revocation protocols of BLECA [21,22]. Table 6 summarizes this comparison results. Revoking a cross-domain certificate requires one write operation in both BLECA and [21]. However, if *Park A's* all cross-domain certificates authorized by *Park B* (assuming that the number of cross-domain certificates is n) require to be revoked, BLECA still needs only one write operation. In contrast, the scheme [21] needs *n* write operations. For the scheme [22], the authentication servers perform the cross-domain certificate revocation locally. Although local servers run much faster than the blockchain, revoking numerous cross-domain certificates still incurs significant costs. For BLECA, increasing the number of cross-domain devices in the park does not increase the cost of revoking cross-domain certificates when terminating park cooperation. From the above test results on write operations, it can be observed that the growth in the number of parks in the cooperative alliance can increase the cost of revoking cross-domain certificates resulting from terminating park cooperation. However, the protocol for cross-domain certificates batch revocation in this solution only requires one write operation, ensuring that the time cost does not increase significantly. Additionally, within the scheme [22], the process of revoking a device involves notifying the authentication servers in every domain to remove the relevant identity data, incurring substantial communication costs. This revocation process faces challenges in promptly synchronizing certificate information across all domains. In contrast, the device cancellation algorithm of BLECA ensures that the cross-domain certificates associated with the device can be revoked simultaneously when a device is canceled, which improves system security.

Table 6: Cross-domain certificate revocation cost comparison

Operation	[21]	[22]	BLECA
Single revocation	T_w	T_{ld}	T_w
Batch revocation	nT_w	nT_{ld}	T_w

Note: T_w: Write operation of blockchain; T_{ld}: Delete operation of local database.

In summary, our scheme can deliver lightweight and efficient cross-domain authentication services for smart parks, with a more secure and flexible certificate revocation mechanism.

7 Conclusion and Future Work

This paper introduces a blockchain-based cross-domain authentication scheme tailored for smart parks. This scheme designs the process of cross-domain authentication and session key agreement, simplifying the interaction of cross-domain authentication and reducing the certificate maintenance burden on devices, ensuring the feasibility of cross-domain interactions for resource-constrained devices. The mutual independence of cross-domain certificates aligns with the individual requirements of each park to manage certificates according to its own standards. A standout feature of this scheme is its provision of an editable cross-domain directed graph and a comprehensive certificate revocation mechanism, making it highly suitable for dynamic collaboration scenarios among smart parks. The batch revocation protocol of cross-domain certificates dramatically slashes computational and storage costs, which guarantees the efficiency of cross-domain authentication. The security analysis and performance evaluation manifest the practical security, high efficiency, and low cost of this scheme. In our future work, we will delve into blockchain consensus algorithms to develop a unique consensus mechanism tailored for IoT cross-domain authentication.

Acknowledgement: Not applicable.

Funding Statement: This work was supported in part by the National Natural Science Foundation Project of China under Grant No. 62062009 and the Guangxi Innovation-Driven Development Project under Grant Nos. AA17204058-17 and AA18118047-7.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Fengting Luo, Ruwei Huang; data collection: Fengting Luo; analysis and interpretation of results: Fengting Luo, Ruwei Huang, Yuyue Chen; draft manuscript preparation: Fengting Luo. Yuyue Chen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, pp. 107930–107946, 2021.

[2] Z. F. Zeng, Y. Yuan, J. Zhang and Y. Liu, "Blockchain in smart park: Application scheme design," in *Proc. of the First Int. Electronics Communication Conf.*, Okinawa, Japan, pp. 76–83, 2019.

[3] H. Du, J. Zeng, Y. An, J. Zhang and J. Zhao, "Exploration on the application of blockchain in the security system of smart park," in *Proc. of the First Int. Electronics Communication Conf.*, Okinawa, Japan, pp. 146–153, 2019.
[4] L. Chen, H. W. Lim and G. Yang, "Cross-domain password-based authenticated key exchange revisited," *ACM Transactions on Information and System Security*, vol. 16, no. 4, pp. 1–32, 2014.
[5] M. Wang, C. Qian, X. Lin, S. Shi and S. Chen, "Collaborative validation of public-key certificates for IoT by distributed caching," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 92–105, 2020.
[6] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI around with decentralized automated incentives," in *IEEE Symp. on Security and Privacy*, San Jose, CA, USA, pp. 410–426, 2017.
[7] S. Biswas, K. Sharif, F. Li, B. Nour and Y. Wang, "A scalable blockchain framework for secure transactions in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4650–4659, 2018.
[8] J. Huang, L. Kong, G. Chen, M. Y. Wu, X. Liu *et al.,* "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.
[9] F. M. Amine, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras *et al.,* "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2018.
[10] L. Xue, H. Huang, F. Xiao and W. Wang, "A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2409–2420, 2022.
[11] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du *et al.,* "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1229–1241, 2020.
[12] M. Shen, J. Zhang, L. Zhu, K. Xu and X. Tang, "Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5773–5783, 2019.
[13] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, Chicago, IL, USA, pp. 1–4, 2016.
[14] W. Cai, L. Yu, R. Wang, N. Liu and E. Deng, "Research on application system development method based on blockchain," *Journal of Software*, vol. 28, no. 6, pp. 1474–1487, 2017.
[15] Z. Huang, "The application of blockchain in edge computing and IoT," *Cyberspace Security*, vol. 9, no. 8, pp. 25–30, 2018.
[16] Z. Zhou, L. Li and Z. Li, "Efficient cross-domain authentication scheme based on blockchain technology," *Journal of Computer Applications*, vol. 38, no. 2, pp. 316–320, 2018.
[17] Y. Huang, Y. Wang, W. Chen and Z. Zhang, "PKI cross-domain authentication model based on alliance chain," *Computer Engineering and Design*, vol. 42, no. 11, pp. 3043–3051, 2021.
[18] G. Zhao, B. Di and H. He, "A novel decentralized cross-domain identity authentication protocol based on blockchain," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 1, pp. e4377, 2022.
[19] J. Zhang, X. Li, X. Zeng, Y. Zhao, R. Duan *et al.,* "Cross domain authentication and key agreement protocol based on blockchain in edge computing environment," *Journal of Cyber Security*, vol. 6, no. 1, pp. 54–61, 2021.
[20] H. Zhang, X. Chen, X. Lan, H. Jin and Q. Cao, "BTCAS: A blockchain-based thoroughly cross-domain authentication scheme," *Journal of Information Security and Applications*, vol. 55, pp. 102538–102547, 2020.
[21] P. Gu and L. Chen, "An efficient blockchain-based cross-domain authentication and secure certificate revocation scheme," in *2020 IEEE Sixth Int. Conf. on Computer and Communications*, Chengdu, China, pp. 1776–1782, 2020.
[22] M. Wang, L. Rui, Y. Yang, Z. Gao and X. Chen, "A blockchain-based multi-CA cross-domain authentication scheme in decentralized autonomous network," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2664–2676, 2022.

[23] G. Qiao, Y. Zhuang, T. Ye and Y. Qiao, "BCDAIoD: An efficient blockchain-based cross-domain authentication scheme for internet of drones," *Drones*, vol. 7, no. 5, pp. 302–327, 2023.
[24] S. K. Rana, S. K. Rana, K. Nisar, A. A. A. Ibrahim, A. K. Rana *et al.,* "Blockchain technology and artificial intelligence based decentralized access control model to enable secure interoperability for healthcare," *Sustainability*, vol. 14, no. 15, pp. 9471–9495, 2022.
[25] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, 2008.
[26] W. Wang and C. C. Su, "A system with high embedding capacity for covert communication in bitcoin," in *Proc. of the IFIP Int. Conf. on ICT Systems Security and Privacy Protection*, Maribor, Slovenia, pp. 21–23, 2020.
[27] L. Zhang, Z. Zhang, W. Wang, R. Waqas, C. Zhao *et al.,* "A covert communication method using special bitcoin addresses generated by vanitygen," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 597–616, 2020.
[28] Z. Li, Z. Yang, S. Xie, W. Chen and K. Liu, "Credit-based payments for fast computing resource trading in edge-assisted internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6606–6617, 2019.
[29] H. Huang, S. Zhou, J. Lin, K. Zhang and S. Guo, "Bridge the trustworthiness gap amongst multiple domains: A practical blockchain-based approach," in *ICC 2020–2020 IEEE Int. Conf. on Communications*, Dublin, Ireland, pp. 1–6, 2020.
[30] A. Singh, R. M. Parizi, Q. Zhang, K. K. R. Choo and A. Dehghantanha, "Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities," *Computers & Security*, vol. 88, pp. 101654–101669, 2020.
[31] R. P. Sukumaran and S. Benedict, "Survey on blockchain enabled authentication for industrial internet of things," in *2021 Fifth Int. Conf. on IoT in Social, Mobile, Analytics and Cloud*, Palladam, India, pp. 1510–1516, 2021.
[32] X. Wei, X. Y. Wang, Z. Yu, S. Y. Guo and X. S. Qiu, "Cross domain authentication for IoT based on consortium blockchain," *Journal of Software*, vol. 32, no. 8, pp. 2613–2628, 2021.
[33] C. Fromknecht, D. Velicanu and S. Yakoubov, "A decentralized public key infrastructure with identity retention," *Cryptology ePrint Archive*, 2014.
[34] N. Kshetri, "Can blockchain strengthen the Internet of Things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.
[35] A. Rana, S. Sharma, K. Nisar, A. A. A. Ibrahim, S. Dhawan *et al.,* "The rise of Blockchain Internet of Things (BIoT): Secured, device-to-device architecture and simulation scenarios," *Applied Sciences*, vol. 12, no. 15, pp. 7694–7715, 2022.
[36] M. Dave, V. Rastogi, Miglani M., P. Saharan and N. Goyal, "Smart fog-based video surveillance with privacy preservation based on blockchain," *Wireless Personal Communications*, vol. 124, pp. 1677–1694, 2022.
[37] J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya, "Merkle-Damgård revisited: How to construct a hash function," in *25th Annual Int. Cryptology Conf.*, Santa Barbara, California, USA, pp. 430–448, 2005.
[38] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, Germany, pp. 387–398, 1996.
[39] Y. B. Chen, C. R. Zhong, C. R. Zhou, L. Y. Xue and H. P. Huang, "Design of cross-domain authentication scheme based on medical consortium chain," *Computer Science*, vol. 49, no. s1, pp. 537–543, 2022.