applied sciences

MDPI

Article

A Blockchain-Based Cooperative Authentication Mechanism for Smart Grid

Yunfa Li [1,2,*], Di Zhang [1], Zetian Wang [3] and Guanxu Liu [1]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China; 202050282@hdu.edu.cn (D.Z.)
[2] Lishui Research Institute, Hangzhou Dianzi University, Hangzhou 310018, China
[3] HDU-ITMO Joint Institute, Hangzhou Dianzi University, Hangzhou 310018, China; wztcoco@hdu.edu.cn
* Correspondence: yunfali@hdu.edu.cn

Abstract: With the advancement of smart devices, the operation and communication of smart grids have become increasingly efficient. Many smart devices such as smart meters, smart transformers, and smart grid controllers are already widely used in smart grids. Thus, a series of complex architectures and a series of communication modes have been formed. However, these smart devices will be exposed to various cyber attacks such as distributed denial of service (DDoS) attack and replay attack. This is because they are open and dynamic. Therefore, there are serious security problems in the complex architectures and the communication modes. In this paper, we propose a multi-domain authentication mechanism based on blockchain cooperation to maintain the security of smart devices. In this mechanism, we propose a series of methods and algorithms, which include initialization method based on blockchain cooperative authentication, dynamic change method of intelligent devices and information, cross-domain authentication algorithm, and cross-domain key cooperative algorithm. To demonstrate the security and effectiveness of our proposed mechanism, we analysed its security and conducted a series of simulation experiments. The analysis and simulation experiments show that our proposed approach is secure and effective.

Keywords: smart grid; cross-domain; blockchain; security

Citation: Li, Y.; Zhang, D.; Wang, Z.; Liu, G. A Blockchain-Based Cooperative Authentication Mechanism for Smart Grid. *Appl. Sci.* **2023**, *13*, 6831. https://doi.org/10.3390/app13116831

Academic Editors: Ying Zhang, Zhengcheng Dong and Meng Tian

Received: 16 May 2023
Revised: 29 May 2023
Accepted: 1 June 2023
Published: 5 June 2023

1. Introduction

In general, a smart grid [1] is an energy management system that includes various energy production, transmission, and distribution devices. It can monitor and control energy usage in real time and can better integrate renewable and conventional energy sources. Smart devices for the smart grid, such as smart meters and smart charging posts, are being deployed in various locations to collect information or trigger events. This brings great convenience to people's lives and has become an indispensable part of our daily lives. For example, we can monitor our electricity consumption in real time through smart meters and optimize our electricity consumption plans to save energy costs [2]. We can charge our electric cars through smart charging posts to reduce carbon emissions. We can integrate renewable energy sources through smart grids for more sustainable energy development, and so on. Multi-disciplinary smart grid alliances can improve energy use efficiency and reduce costs, but they also face some security challenges.

Firstly, many of the devices used in the smart grid are resource-constrained, which means that they are unable to run sophisticated algorithms to improve their security and are therefore vulnerable to attacks. Secondly, because smart grids are composed of many heterogeneous components, the communication standards and data formats of these components are not uniform, which also increases their vulnerability to attacks and threats. When a smart grid is hacked and compromised, grid information can be stolen or corrupted, which can cause damage not only to the affected area but also to other areas in terms of

economic and property damage. Most seriously, this could lead to a collapse of the power system and even endanger public life and health.

In traditional smart grid authentication schemes, a trusted third party is usually relied upon to provide cross-domain authentication services. As with a traditional public key infrastructure (PKI) [3], there is a centralized certificate authority that can issue digital certificates for smart grid devices. During the cross-domain authentication process, devices can use this digital certificate to prove their identity. However, this method of cross-domain authentication also has shortcomings. Firstly, it is highly dependent on a third-party certificate authority. When this certificate authority is attacked, the entire system can become unavailable, leading to a single point of failure problem. Secondly, these trusted third parties may become untrustworthy at some point. For example, some organizations may compromise the privacy of their users for their own benefit, causing significant damage to the user. For these reasons, a decentralized, secure, and lightweight cross-domain authentication access mechanism is needed in a multi-domain smart grid.

Blockchain is essentially a decentralized, distributed, and tamper-evident model of data sharing and transmission [4]. Its main implementation model is to store asset and transaction information on a peer-to-peer network. Blockchain has three main features: decentralization, immutability, and traceability, which enable users or devices to achieve secure transactions in an untrusted network. Due to these characteristics, its application scenarios are no longer limited to the financial field [5]. Currently, we can use the programmable nature of smart contracts to apply blockchain to various scenarios and smart grid is one of its main application scenarios [6].

Based on the above-mentioned problems faced by cross-domains, we propose a blockchain-based cross-domain authentication mechanism for multi-domain smart grid devices by combining the features of blockchain. In this mechanism, smart grids from different domains form a virtual federation in which devices from different domains can access each other's data through cross-domain authentication.

The blockchain in our proposed authentication mechanism uses a federated chain. A federated chain is a type of blockchain that is formed by multiple institutions or organizations and is under common control. Individuals or organizations wanting to access a particular federated chain must be authorized to do so. The use of federated chains not only meets the scenario of a multi-domain smart grid but also the high efficiency of its consensus algorithm can meet the timeliness of cross-domain authentication. Our main contributions are as follows.

- We propose a blockchain-based cross-domain authentication mechanism for multi-domain smart grids. In this mechanism we store the hash of the device's certificate in the blockchain, which not only saves storage space but also improves security. With this mechanism, devices from different domains can be securely and efficiently authenticated across domains before they can access each other.
- We have a domain manager and a blockchain proxy node in each domain. In particular, the domain manager is mainly responsible for managing the smart grid devices in its domain, while the blockchain proxy node is mainly responsible for participating in the consensus process of the blockchain. In this way, we have improved the efficiency of cross-domain authentication.
- We performed a comprehensive safety analysis and conducted simulation experiments to verify the feasibility and efficiency of our proposed solution.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 is preliminaries, which mainly describe some theoretical knowledge and the overall structure of our proposed mechanism. Section 4 describes our proposed mechanism in detail. Section 5 performs a detailed security analysis to illustrate the security of our proposed mechanism. Section 6 conducts some simulation experiments to evaluate the efficiency of our proposed mechanism. Section 7 concludes this work.

2. Related Work

As the smart grid continues to evolve, its complexity continues to increase, while smart grid devices are also exposed to an increasing number of security threats. Malicious users may affect the operation of the entire smart grid by stealing grid data or through malicious nodes, which may have serious consequences. To address these security threats, Dipanwita et al. [7] proposed a novel mutual authentication scheme based on elliptic curve cryptography aimed at improving the security of the smart grid environment. Chim et al. [8] proposed a novel authentication scheme for protecting consumer privacy that uses hash-based message authentication codes, making the authentication process simple and effective. Li et al. [9] proposed a novel and secure message authentication scheme for mutual authentication and key establishment between smart grid devices, retaining the identity of the gateway during message transmission.

Zhang et al. [10] proposed a mitigated authentication protocol based on elliptic curve cryptography that uses a tamper-resistant device on the smart device side to achieve a delicate balance between performance and security for privacy protection in the smart grid. Hasen et al. [11] provide a novel authentication scheme between smart grid utility networks and home area network smart meters, and provide a new authentication scheme. A new key management protocol is provided for data communication between utility servers and customer smart meters. Mahmood et al. [12] proposed a hybrid lightweight authentication scheme based on Diffie–Hellman, which uses advanced encryption standard (AES) and RSA (Rivest–Shamir–Adleman) to generate session keys. Zhang et al. [13] proposed a lightweight anonymous authentication and key negotiation scheme for smart grids that allows mutual authentication between smart meters and servers, and improves authentication speed while increasing the anonymity and untraceability of smart meters.

With the development of blockchain, the combination of blockchain technology and smart grid is becoming more and more popular among researchers. Wang et al. [14] proposed a blockchain-based mutual authentication and key negotiation protocol for smart grid systems based on edge computing. Wang et al. [15] focus on solving some of the authentication problems that still exist in smart grids and combine blockchain, elliptic curve cryptography, and dynamic join and exit mechanisms to create a reliable and efficient authentication protocol for smart meters and utility centres.

Vasudev et al. [16] proposed a model for identification and authentication of IoT (Internet of Things) devices based on blockchain technology in smart grids, and gave a concrete implementation that enables authentication of devices in a trusted model using blockchain. Zhong et al. [17] proposed a new distributed authentication and authorization protocol for smart grids based on blockchain technology that integrates decentralized features such as centralized authentication and immutable ledgers to achieve identity authentication and resource authorization for smart grids. Badshah et al. [18] designed a lightweight authentication key exchange scheme for smart grids that allows secure communication between smart meters and service providers. In this scheme data is stored in a secure blockchain network.

Nyangaresi et al. [19] proposed an anonymous key and authentication protocol that not only meets the security requirements but also has low bandwidth and computational cost in order to address some of the security issues in the smart grid. Chaudhry et al. [20] proposed a new demand response managed authentication scheme (DRMAS) that provides all the necessary security requirements and is resistant to known attacks in order to ensure the security of the smart grid environment. Badar et al. [21] proposed an identity-based authentication protocol that can be used for monitoring the power supply network in a smart grid environment and that is resilient to various cyber attacks and to physical attacks on sensors. Bera et al. [22] designed a smart grid system supporting the Internet of Things, a new blockchain-based access control protocol in which data is securely brought from the respective smart meter to the service provider. Furthermore, the blockchain network consists of the providers of the service and the protocol is considered secure against various attacks.

In this paper we apply blockchain to the authentication of smart grid devices, which increases the security of the authentication process. In contrast to traditional PKI-based authentication mechanisms, our authentication mechanism does not rely on specific third-party applications and there is no threat of single point of failure. In contrast to blockchain-based authentication mechanisms, we use a federated chain and the hash of the certificate is stored in the blockchain. Our proposed mechanism is therefore more efficient in terms of authentication and does not require a lot of storage space.

3. Preliminaries

In this section, we will first introduce some theoretical knowledge. Then, we will detail the overall architecture of our proposed solution, and explain the roles and responsibilities of each entity in the architecture.

3.1. Basic Theory

3.1.1. Blockchain

The blockchain is essentially a decentralized distributed database that is tamper-evident and traceable. Since its emergence, blockchain has been extensively studied by scholars and is now used for more than just digital currency applications. Due to its decentralization, tamper-evident, and traceable nature, almost anything of value can be tracked and traded on a blockchain network. Furthermore, blockchain is not limited to the financial sector, as its programmable smart contracts can be applied to various scenarios. One of its main application scenarios is the Internet of Things.

Blockchain can be divided into public blockchain, private blockchain, licensing blockchain, and federated blockchain. Among them, federated blockchain is one in which there are multiple organizations that share the responsibility of maintaining the blockchain and its transactions are faster, thus making it ideal for such scenarios where the IoT crosses domains. Therefore, our proposed mechanism uses federated blockchain, which ensures efficiency and security at the same time.

3.1.2. Digital Signature

Digital signatures can be viewed as the inverse application of public key ciphers. A signature can be generated using a private key, which means only a user with the private key can create a signature. Verifying a signature requires the public key, which is public and can be used by all users to verify the signature. If the signature is successfully verified, we can confirm that the message was sent by the user with the private key and not by any other user. When there is a dispute between two communicating parties regarding the content or authenticity of a message, the digital signature becomes a strong piece of evidence.

3.1.3. Hash Functions

With hash functions, regardless of the input of any length of string, the output is a fixed length of the hash value. Moreover, the hash function has a one-way nature; it is difficult to get the input of the hash function from the given hash value.

3.2. System Model

This section details the overall architecture of our proposed blockchain-based collaborative authentication mechanism between multi-domain smart grid devices, as shown in Figure 1.

The architecture consists of multiple smart point grid domains, each with roughly the same structure, including smart meters, a domain manager, and blockchain agent nodes. At the top of each domain is a blockchain. The type of blockchain we have chosen in this paper is a federated chain. Each domain sends a blockchain agent node to jointly maintain the federated chain. The notation is described in Table 1 and each module of the architecture is described in detail below.

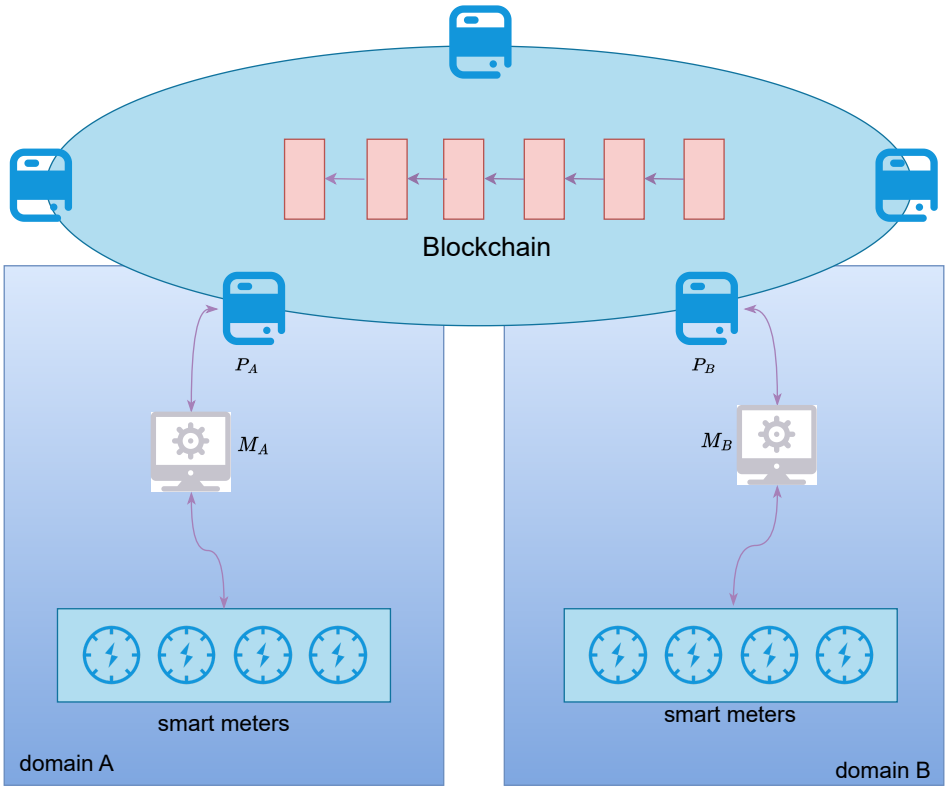

Figure 1. System architecture.

Table 1. Symbol description.

Symbol	Description
D_i	The ith device in domain A
D_j	The jth device in domain B
M_A and M_B	Administrator of domain A and B
P_A and P_B	Blockchain proxy nodes of domain A and B

Smart devices: Examples of smart devices are smart meters or smart switches. One of the main uses of smart meters is to monitor, record, and transmit electrical energy data. Unlike conventional meters, smart meters are able to monitor the electrical load and electricity usage in real time and transmit the data to a smart grid centre or relevant applications via a built-in communication module. Smart switches play a very important role in the smart grid, ensuring the safe and stable operation of the power system, and improving its reliability and self-healing capability.

Domain manager: The domain manager is the administrator for each domain and is unique within each domain with powerful storage and computing capabilities. The domain manager is responsible for managing smart devices such as smart meters in the domain, including the joining, updating, and exiting of devices. The domain manager performs system initialization at the start and generates the necessary parameters for the system. The domain manager is secure and trusted within the domain.

Blockchain agent nodes: Blockchain agent nodes can interact with domain managers and devices in the smart grid, with domain managers in each domain working together to maintain the federated blockchain. The blockchain we use is the federated chain, which is a permission blockchain where nodes trust each other. The identity information of the devices, including device certificates, is stored in the blockchain. As the blockchain is tamper-proof, this ensures that devices without certificates cannot forge them, ensuring that authentication is secure.

4. Our Proposed Mechanism

In this section we describe in detail the specifics of the authentication mechanism. The mechanism consists of four main components: initialization method based on blockchain cooperative authentication, dynamic change method of intelligent devices and information, cross-domain authentication algorithm, and cross-domain key cooperative algorithm.

4.1. Design Goals

Our proposed cross-domain authentication mechanism should satisfy the following design goals.

- Lightweight: Since smart devices in a smart grid have limited resources, our proposed authentication mechanism should require as few resources as possible to complete the authentication to meet the needs of most smart devices.
- Security: As the devices in the smart grid are connected via the Internet, it can be subject to many types of cyber attacks, so the mechanism we propose should be able to withstand these attacks.
- Mobility: Our proposed mechanism can satisfy the need for cross-domain access.

4.2. Initialization Method Based on Blockchain Cooperative Authentication

The initialization phase is performed by the domain manager of each domain and its purpose is mainly to generate the necessary parameters needed by the whole system. Elliptic curve cryptography (ECC) is a public key encryption algorithm based on elliptic curves. Its essence is to achieve encryption using the discrete logarithm problem. The main advantage of ECC is to provide faster performance and higher levels of security while using smaller keys. Therefore, we will use the elliptic curve cryptography algorithm to generate our keys. Let us take domain A as an example; the initialization method based on blockchain cooperative authentication is shown below.

First, M_A picks an elliptic curve $Ep(a, b)$, where p is a large prime and $a, b \in Z_p$. Then, M_A picks a base point G on the elliptic curve. M_A then picks a random number $r_A \longleftarrow Z_p$ in Z_p as its private key sk_A, which is kept strictly secret by M_A, and computes the public key $pk_A = sk_A.G$. Next, M_A chooses a hash function h_0, which can be $SHA - 1$ or a more secure version. Then, M_A signs the above parameters $\{p, a, b, G, h_0\}$ and publishes them to the blockchain proxy node P_A, which then publishes them to the federated chain.

The joining process for other domains is similar to domain A and will not be repeated here.

4.3. The Dynamic Change Method of Smart Devices and Information

In the smart device management phase, there are some smart devices joining and logging out, and some device information updates. Each domain administrator only manages the smart devices in its own domain and is responsible for exchanging information with other domain administrators. The dynamic change of smart devices and information mainly include three key activities: device join, device information update, and device logout.

4.3.1. Device Joining

Devices in the smart grid need to rely on a specific domain before they can be accessed across domains, so devices need to join a specific domain. The domain manager is responsible for managing the joining of devices. Let us take device D_i as an example; if device D_i wants to join domain A, as shown in Figure 2, the join process is as follows:

First, the device D_i generates a public–private key pair. Device D_i picks a random number $r_i \longleftarrow Z_p$ in Z_p as its private key sk_i and then calculates its public key as $pk_i = sk_i.G$. Then, D_i sends its public key pk_i, identity ID_i, and timestamp t_i to M_A to apply to join domain A, where the identity is unique and is different for each device. After M_A receives a request from a device, it first checks the legitimacy of ID_i and, if it is not legal, it exits directly and the device fails to join. If it is legal, M_A will generate a certificate for device D_i with the following structure:

$$Cert_i = Sig(ID_i||pk_i||t_i)_{sk_A} \quad (1)$$

This means that M_A uses its private key to sign the identity ID_i, public key pk_i, and timestamp t_i sent by D_i, to obtain the certificate $Cert_i$ of device D_i. Then, M_A sets an expiration time T_i for the certificate, usually in days, to prevent the device certificate from expiring frequently. M_A saves the certificates $Cert_i$ and T_i to the local database, and calculates the hash value of the certificates to obtain $h_0(Cert_i)$. Next, M_A sends $\{T_i, h_0(Cert_i), state\}$ to the blockchain proxy node P_A. Here, $state$ indicates the status of the certificate, with $state = 1$ meaning the certificate is valid. After that, the blockchain proxy node P_A synchronizes this information to the federated chain through the consensus algorithm and reaches consensus. Finally, P_A feeds the synchronization result to M_A, which then sends the certificate $Cert_i$ to the device D_i. With this, the device is successfully joined to domain A.

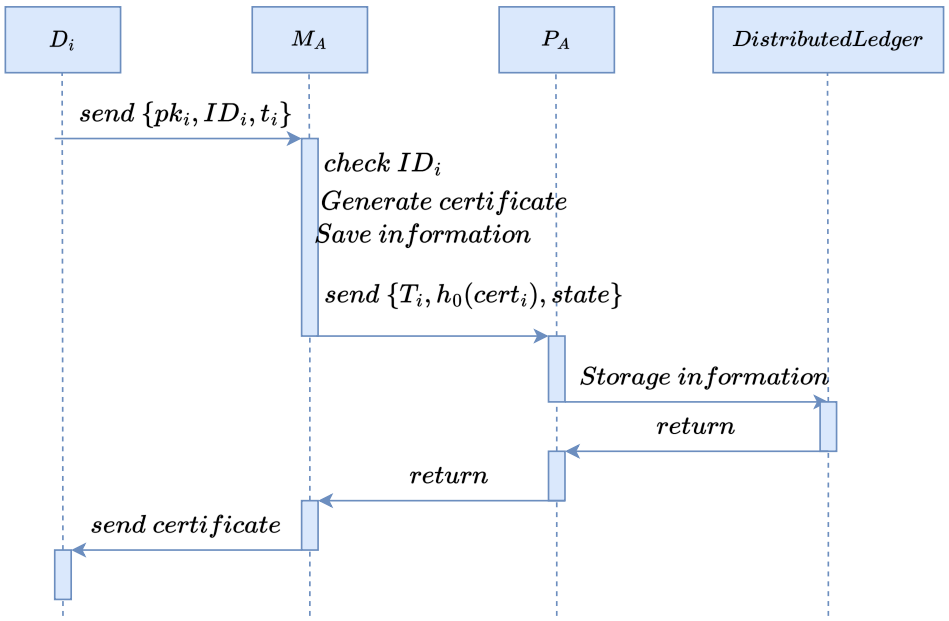

Figure 2. Device joining flow chart.

4.3.2. Device Information Update

When the certificate information in a device is lost or expired, it is necessary to perform an update operation to refresh the device information. For example, let us take the device D_i; its update process is as follows:

First, the device generates a new public–private key pair. Device D_i picks a random number $r_i' \longleftarrow Z_p$ in Z_p as its new private key sk_i'. Then, it computes its new public key as $pk_i' = sk_i' \cdot G$. Device D_i sends $\{sig(pk_i')sk_i, t_i', ID_i\}$ to M_A to perform the update operation for the device. Here, $sig(pk_i')sk_i$ means that device D_i signs the new public key with the old private key, t_i' is the new timestamp, and ID_i is the identity information of device D_i.

After receiving the request, M_A verifies the legitimacy of the signature with the old public key of device D_i. If it is not legitimate, M_A stops the update request. If it is legitimate, M_A generates a new certificate for device D_i with the following certificate structure.

$$Cert_i' = Sig(ID_i||pk_i'||t_i')_{sk_A} \quad (2)$$

The same M_A sets an expiration time T_i' for the new certificate and updates the old certificate in the local database to the new one, synchronizing the expiration time of the certificate. M_A then computes the hash $h_0(Cert_i')$ of the new certificate and sends $T_i', h_0(Cert_i'), state$ to the proxy node P_A of the blockchain. Here, $state$ is also set to 1, indicating that the certificate is valid.

After that, P_A synchronizes the new certificate information to the federated chain and reaches consensus by consensus algorithm. Once complete, P_A feeds the synchronization result back to M_A and M_A sends the new certificate $Cert_i'$ to device D_i. The device information is updated.

4.3.3. Device Logout

The device D_i can actively exit a domain. To do so, D_i signs the certificate with its own private key to obtain $sig(Cert_i)_{sk_i}$ and sends it to the domain manager M_A to request an exit from Domain A. M_A verifies the legitimacy of the signature and terminates the exit request if it is not legitimate. If the signature is legitimate, M_A forwards the request to the blockchain proxy node P_A.

Upon receiving the request, P_A updates the certificate status *state* in the federated chain to 0 and reaches consensus through the consensus algorithm. After that, P_A sends the update result back to M_A. M_A deletes the certificate information about device D_i in the local database upon receiving the update result and then sends the exit result back to the device, indicating that it has successfully exited Domain A.

4.4. Cross-Domain Authentication Algorithm

Cross-domain authentication means that, if a smart device D_i in one domain wants to access a device D_j in another domain and read some information, the identity of D_i must be confirmed by authentication before accessing it. Key activities of the device, including device registration, device update, and device logout, are prerequisites for cross-domain authentication. The overall process of cross-domain authentication is described in detail below.

Suppose device D_i from domain A wants to access device D_j from domain B. Before authentication, D_i must check whether its certificate has expired. If it has, D_i must perform the device update operation. If the certificate is not expired, the cross-domain authentication process can begin, which is shown in Figure 3. The cross-domain authentication algorithm is as follows:

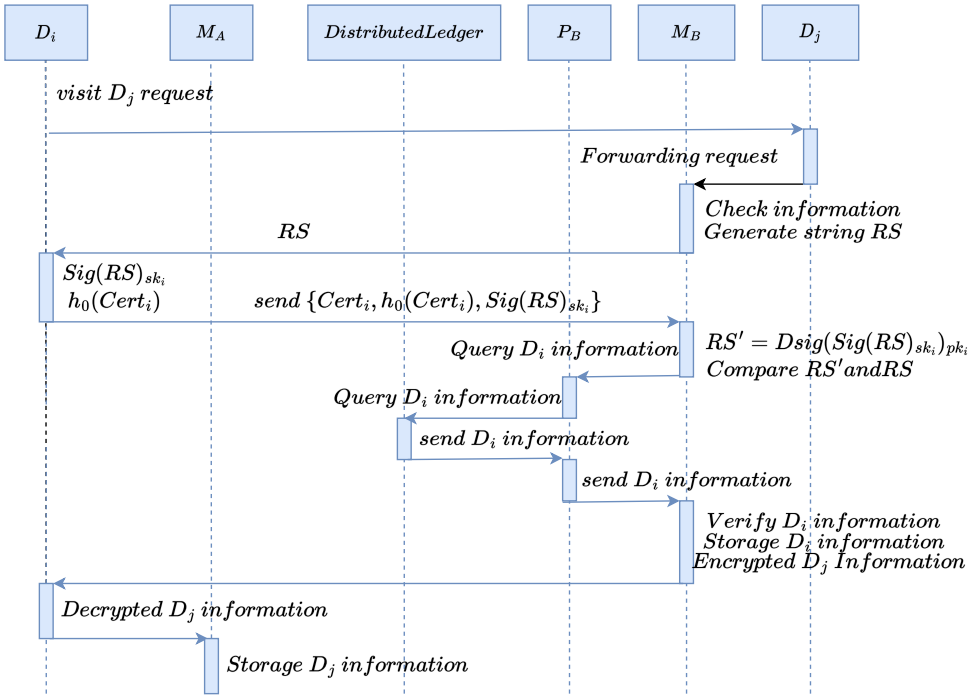

Figure 3. Cross-domain authentication flow chart.

Step 1: Device D_i in domain A sends a connection request to device D_j in domain B. After receiving the request, device D_j forwards this request to the domain manager M_B in domain B.

Step 2: After receiving the request, M_B checks whether the authentication information of device D_i is available in its local database; if so, it jumps to step 3; if not, it jumps to step 6.

Step 3: M_B will send a request for querying device D_i to the blockchain proxy node P_B of domain B. After receiving the request, P_B queries the blockchain for information about D_i and gets $\{T_i, h_0(Cert_i), state\}$ and sends it to M_B.

Step 4: M_B determines whether the hash value of the authentication information in the local database is equal to $h_0(Cert_i)$; if it is equal then execute step 5; if not then skip to step 7.

Step 5: M_B determines whether the certificate is expired, assuming NOW is the current timestamp; if $T_i < NOW$ then the authentication information has expired; go to step 6; if it has not expired then the cross-domain authentication is successful.

Step 6: M_B generates a random string RS and sends it to D_i.

Step 7: After receiving the random string RS, D_i uses its private key to sign to get $Sig(RS)_{sk_i}$. Afterwards D_i hashes his certificate to get $h_0(Cert_i)$, which prevents the certificate from being tampered with. Finally, D_i sends $\{Cert_i, h_0(Cert_i), Sig(RS)_{sk_i}\}$ to M_B.

Step 8: After M_B gets the public key pk_i of device D_i from the certificate, use its public key to verify the signature to get $RS' = Dsig(Sig(RS)_{sk_i})_{pk_i}$, then judge whether RS' is equal to RS; if it is equal then execute the next step, otherwise it means the cross-domain authentication fails so end the cross-domain authentication process.

Step 9: M_B sends a request to B_j to query the device D_i. B_j receives the request and queries the blockchain for information about D_i to get $\{T_i, h_0(Cert_i), state\}$ and sends it to M_B.

Step 10: M_B determines whether the certificate status is legal, whether the hash value of the certificate in the blockchain is equal to the hash value of the certificate sent from the device D_i, and whether the certificate is expired. If all three conditions are satisfied, the next step is executed, otherwise it means that the cross-domain authentication fails so end the cross-domain authentication process.

Step 11: M_B stores the certificate of device D_i, then encrypts the certificate of device D_j with the public key of D_i to get $Sig(Cert_j)_{pk_i}$ and sends it to device D_i.

Step 12: D_i decrypts the certificate $Cert_j = DSig(Sig(Cert_j)_{pk_i})_{sk_i}$ of device D_j with its private key and sends it to the manager M_A of domain A.

Step 13: M_A receives the certificate $Cert_j$ from D_j and stores it in the local database, thus ending cross-domain authentication.

4.5. Cross-Domain Key Cooperative Algorithm

After the devices D_i in domain A and devices D_j in domain B complete mutual authentication, they can negotiate a key together, after which they can communicate securely through that key. We will use the ECHDE key exchange algorithm to exchange keys, as shown in Figure 4; its specific algorithm is as follows.

Step 1: Share the same elliptic curve parameters $\{p, a, b, G\}$ by initializing the system as described above.

Step 2: Device D_i picks a random number $r_i \longleftarrow Z_p$ in Z_p as its private key sk_i, then calculates its public key as $pk_i = sk_i.G$ and sends its public key pk_i to device D_j afterwards.

Step 3: After the device D_j receives pk_i it also picks a random number $r_j \longleftarrow Z_p$ in Z_p as its private key sk_j, then calculates its public key as $pk_j = sk_j.G$ and sends pk_j to the device D_i.

Step 4: After that D_i and D_j can compute the shared key separately, both shared key $SK = pk_i.r_j = pk_j.r_i = r_i.r_j.G$.

After that, both parties have a common key and can use the common key for encrypted communication, which ensures the efficiency and the security of the information of both parties.

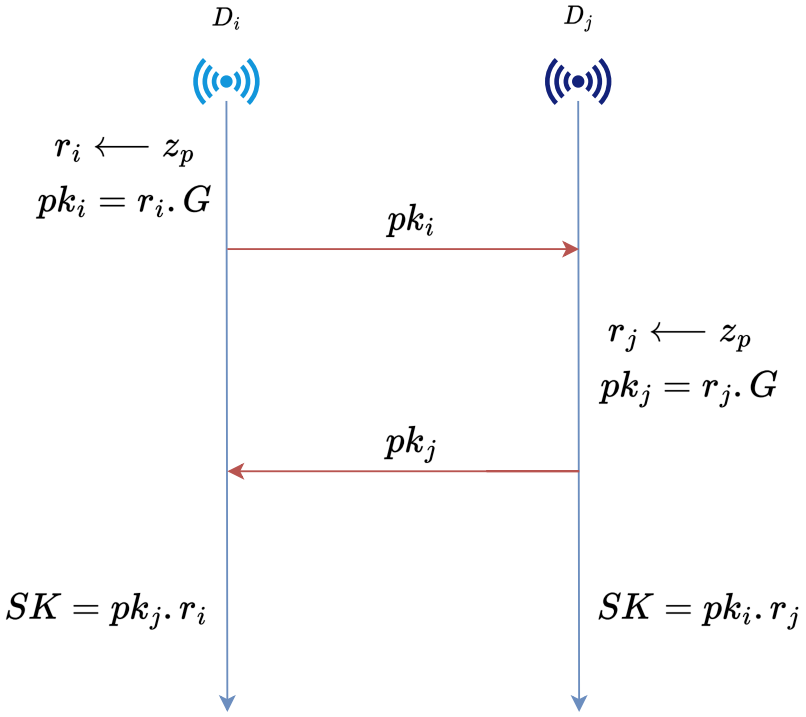

Figure 4. Schematic diagram of key negotiation.

5. Security Analysis

In this section, we will analyse the potential attacks and security features of our proposed mechanism in detail. We assume that the cryptographic principles and blockchain in our proposed mechanism are secure.

As smart grid devices in each domain are connected to the Internet, they are vulnerable to many attacks. An attacker may be able to obtain critical information about a device and launch a forgery attack. An attacker may also intercept information sent by the sender and send a different message to the receiver, which is known as a man-in-the-middle attack. In a replay attack, the attacker repeats the packets received by the destination host, spoofing the system and bringing it down. When an attacker operates multiple machines to attack one or more targets, this attack is known as a DDoS attack. In addition, an eavesdropping attack occurs when an attacker uses sniffing tools to steal communication information between two devices. Finally, an authorized user can launch an internal attack.

The following is an analysis of how our proposed mechanism protects against these attacks.

Forgery Attack: In our proposed mechanism, each device generates and saves its own private key. As long as the private key is not compromised, it is guaranteed to be infeasible to forge a signature. Even if the attacker knows the public key pk_i and the base G of the device, it is almost impossible for the attacker to compute the private key of the device, which is the classical discrete logarithm problem.

Man-in-the-Middle Attack: In our proposed mechanism, the communication between any two communicating parties is encrypted. Therefore, even if a man-in-the-middle hijacks the communication data, the attacker cannot obtain valid information without the key. In our mechanism, if an attacker steals the random character RS sent by M_B to the device, the attacker needs to sign this string with their private key. M_B then needs the attacker's certificate and goes to the blockchain to query and verify the legitimacy of this certificate. Due to the nature of the underlying blockchain technology that we use, an

attacker cannot tamper with the data in the blockchain and generate an illegal certificate. Therefore, our solution is able to resist man-in-the-middle attacks.

Replay Attack: In our proposed authentication mechanism, when performing cross-domain authentication, the domain manager M_B stores the certificate of the device D_i in domain A in its own local database. Therefore, even if the attacker repeatedly sends the same cross-domain authentication request, M_B will check in the local database whether the device has previously performed cross-domain authentication and the validity of the certificate. If the device has already performed cross-domain authentication and the certificate is legitimate, M_B will not repeat the cross-domain authentication.

DDoS Attack: In our proposed authentication mechanism, the authentication information is stored in the blockchain. The blockchain is distributed in nature, meaning that each blockchain proxy node has the ledger of the entire blockchain. In the federated chains that we use, the failure of less than one-third of the nodes does not affect the entire distributed ledger. Additionally, when the attacked ledger is restored to normal, the complete ledger information can be obtained from other nodes. Therefore, our proposed mechanism is fully resistant to DDoS attacks.

Eavesdropping Attack: In our proposed authentication mechanism, the information transmitted between the smart device and the domain management is non-private information, which means that it is not valuable even if it is intercepted by an eavesdropper. Other non-private information is encrypted with a key and, even if an eavesdropper obtains this information, it cannot be decrypted without the key. Therefore, our proposed authentication mechanism is fully resistant to eavesdropping attacks.

Insider Attack: In our proposed authentication mechanism, the blockchain stores an abstract of the certificate, rather than the complete certificate information. Even if an insider attacker queries the abstract of the certificate from the blockchain, it is almost impossible for the attacker to obtain the complete certificate information from the abstract, because the hash algorithm used has the anti-collision property. Therefore, our proposed authentication mechanism is fully capable of resisting insider attacks.

6. Implementation and Evaluation

In this section we perform a series of simulations of the proposed mechanism and analyse the key performance indicators involved.

6.1. Experimental Setup

In this simulation experiment, we simulated two different domains. In each domain a blockchain proxy node and a domain manager server were equipped along with two Raspberry Pi devices, where each Raspberry Pi simulates a device in a specific smart grid. The blockchain proxy node and domain manager were configured with an Intel Core i5 CPU@2.9HZ, 16 GB of RAM and running ubuntu 22.04, a 64-bit operating system. The blockchain we chose was a federated chain, so we implemented our solution on the HyperLedger Fabric platform; the version chosen was v2.4 and the consensus algorithm used was the draft consensus algorithm. The pc uses JDK 1.8 to write the code for the chain (smart contract) via Java and then installs Hyperledger Caliper v0.5 to perform performance tests on our proposed solution and analyse the corresponding performance metrics. We will install a virtual machine on a PC to simulate the corresponding smart grid device, choosing a Raspberry Pi 4 and allocating 4 GB of RAM. We will use prime256v1 elliptic curves to generate public–private key pairs and use the elliptic curve digital signature algorithm (ECDSA) as the signature algorithm to generate signatures.

6.2. Storage Overhead

The storage overhead refers to the authentication-related data information stored in the smart grid device. The consumption incurred by other information stored in the smart grid device is out of the scope of our discussion. Since we use prime256v1 elliptic curve to generate public–private key pairs, both the public and private keys are 256 bits in size, which is 32 bytes. In our proposed authentication mechanism, the certificate consists of an identity, a public key, a timestamp, and a signature. The identity is 128 bits and globally unique, the public key is 256 bits, the timestamp is 64 bits, and the signature is 72 bits maximum. Therefore, the size of the certificate is 520 bits. Moreover, the hash value of the certificate is saved in the blockchain and the certificates are saved in the off-chain database, which can reduce the storage pressure of the blockchain, improve the consensus efficiency, and also prevent the certificate from not being modified.

6.3. Communication Overhead

The communication overhead described here counts only the communication information related to cross-domain authentication. There are two possible cases when a device D_i in domain A authenticates with a device D_j in domain B across domains, which we will discuss separately here. In the first case, when the authentication information of device D_i is stored in the domain manager in domain B, only the abstract of device D_i certificate needs to be queried from the blockchain and the communication overhead is only the size of the abstract of device D_i stored in the blockchain, which is the size of the expiration time plus the size of the certificate abstract and a status flag bit, totalling 321 bits. In the second case, when the domain manager in domain B does not save the authentication information of device D_i, it is necessary not only to query D_i certificate digest from the blockchain, but also to send the certificate, certificate digest, and signature of device D_i. The size of the certificate is 520 bits, the size of the certificate digest is 256 bits, and the signature requires 72 bits, so the final communication overhead requires 1169 bits. In our proposed authentication mechanism, the certificate expiration time is generally in days. When a device is successfully authenticated, it does not need to be authenticated again, so the communication overhead in most cases is only 321 bits, and the communication overhead is 1169 bits only when a device is authenticated for the first time or when the authentication information is lost.

6.4. Authentication Efficiency

Authentication efficiency is an important metric to evaluate our proposed authentication mechanism. The information related to certificates in our proposed mechanism is stored in the blockchain, so frequent on-chain operations are required, so we mainly record the latency of running smart contracts, which mainly includes write latency and query latency. The write latency is the time required between the invocation of the write operation and the data being written to the blockchain, and the query latency is the time required between the invocation of the read operation to read data from the blockchain and the return of the data to the result. For a legitimate transaction, the average time taken for a single write is 126 ms and the average time taken for a single query is 51 ms. As shown in Figure 5, as the number of transactions increases, the read and write latencies are also increasing in an almost linear trend. The write latency increases faster due to the fact that the call to the chain code looks up the replica pool first, which saves time.

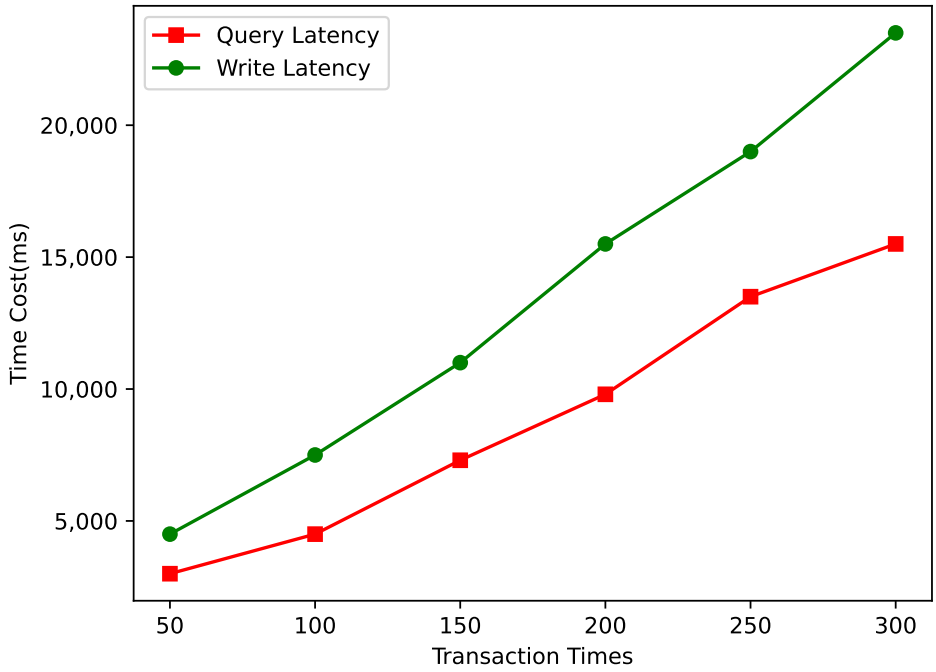

Figure 5. Time costs on writing and querying data.

7. Conclusions

In this paper, we propose a blockchain-based cooperative authentication mechanism for smart grid devices from different domains. The mechanism enables mutual authentication with security and we have performed a security analysis of the proposed mechanism to demonstrate its ability to meet security requirements. We have verified the mechanism through simulation experiments and the results show that it is efficient in terms of storage, communication, and authentication. The storage overhead is 520 bits, the communication overhead is 1169 bits for initial authentication, the query latency is 51 ms, and the write latency is 126 ms. This is perfectly suited to the scenario of mutual authentication between smart grid devices.

However, the consensus algorithm used in this paper is the Raft algorithm and, although it can meet the demand for device authentication in smart grids, the efficiency of the Raft algorithm may not be able to meet the requirements of low latency and high throughput as the scale of smart grids continues to expand and application scenarios continue to increase. Therefore, further improvements to the consensus algorithm are needed in future work to improve efficiency in order to better meet the needs of smart grids.

Author Contributions: Conceptualization, Z.W.; methodology, D.Z.; software, Z.W.; validation, Z.W., Y.L., and D.Z.; formal analysis, Z.W.; investigation, D.Z. and G.L.; resources, Z.W. and G.L.; data curation, Z.W. and G.L.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W.; visualization, Z.W. and G.L.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Judge, M.A.; Khan, A.; Manzoor, A.; Khattak, H.A. Overview of smart grid implementation: Frameworks, impact, performance and challenges. *J. Energy Storage* **2022**, *49*, 104056. [CrossRef]
2. Mohammadi, F. Emerging challenges in smart grid cybersecurity enhancement: A review. *Energies* **2021**, *14*, 1380. [CrossRef]
3. Jiang, W.; Li, H.; Xu, G.; Wen, M.; Dong, G.; Lin, X. PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI. *Future Gener. Comput. Syst.* **2019**, *96*, 185–195. [CrossRef]
4. Zhuang, P.; Zamir, T.; Liang, H. Blockchain for cybersecurity in smart grid: A comprehensive survey. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3–19. [CrossRef]
5. Guo, S.; Hu, X.; Guo, S.; Qiu, X.; Qi, F. Blockchain meets edge computing: A distributed and trusted authentication system. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1972–1983. [CrossRef]
6. Khan, F.A.; Asif, M.; Ahmad, A.; Alharbi, M.; Aljuaid, H. Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development. *Sustain. Cities Soc.* **2020**, *55*, 102018. [CrossRef]
7. Sadhukhan, D.; Ray, S.; Obaidat, M.S.; Dasgupta, M. A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography. *J. Syst. Archit.* **2021**, *114*, 101938. [CrossRef]
8. Chim, T.W.; Yiu, S.M.; Hui, L.C.; Li, V.O. PASS: Privacy-preserving authentication scheme for smart grid network. In Proceedings of the 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), Brussels, Belgium, 17–20 October 2011; pp. 196–201.
9. Li, X.; Wu, F.; Kumari, S.; Xu, L.; Sangaiah, A.K.; Choo, K.K.R. A provably secure and anonymous message authentication scheme for smart grids. *J. Parallel Distrib. Comput.* **2019**, *132*, 242–249. [CrossRef]
10. Zhang, L.; Tang, S.; Luo, H. Elliptic curve cryptography-based authentication with identity protection for smart grids. *PLoS ONE* **2016**, *11*, e0151253. [CrossRef] [PubMed]
11. Nicanfar, H.; Jokar, P.; Leung, V.C. Smart grid authentication and key management for unicast and multicast communications. In Proceedings of the 2011 IEEE PES Innovative Smart Grid Technologies, Perth, WA, Australia, 13–16 November 2011; pp. 1–8.
12. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Shon, T.; Ahmad, H.F. A lightweight message authentication scheme for smart grid communications in power sector. *Comput. Electr. Eng.* **2016**, *52*, 114–124. [CrossRef]
13. Zhang, L.; Zhao, L.; Yin, S.; Chi, C.H.; Liu, R.; Zhang, Y. A lightweight authentication scheme with privacy protection for smart grid communications. *Future Gener. Comput. Syst.* **2019**, *100*, 770–778. [CrossRef]
14. Wang, J.; Wu, L.; Choo, K.K.R.; He, D. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1984–1992. [CrossRef]
15. Wang, W.; Huang, H.; Zhang, L.; Su, C. Secure and efficient mutual authentication protocol for smart grid under blockchain. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 2681–2693. [CrossRef]
16. Dehalwar, V.; Kolhe, M.L.; Deoli, S.; Jhariya, M.K. Blockchain-based trust management and authentication of devices in smart grid. *Clean. Eng. Technol.* **2022**, *8*, 100481. [CrossRef]
17. Zhong, Y.; Zhou, M.; Li, J.; Chen, J.; Liu, Y.; Zhao, Y.; Hu, M. Distributed blockchain-based authentication and authorization protocol for smart grid. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 2115641. [CrossRef]
18. Badshah, A.; Waqas, M.; Abbas, G.; Muhammad, F.; Abbas, Z.H.; Vimal, S.; Bilal, M. LAKE-BSG: Lightweight authenticated key exchange scheme for blockchain-enabled smart grids. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102248. [CrossRef]
19. Nyangaresi, V.O.; Abduljabbar, Z.A.; Refish, S.H.A.; Al Sibahee, M.A.; Abood, E.W.; Lu, S. Anonymous Key Agreement and Mutual Authentication Protocol for Smart Grids. In *Cognitive Radio Oriented Wireless Networks and Wireless Internet, Proceedings of the 16th EAI International Conference, CROWNCOM 2021, Virtual Event, 11 December 2021, and Proceedings of the 14th EAI International Conference, WiCON 2021, Virtual Event, 9 November 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 325–340.
20. Chaudhry, S.A.; Alhakami, H.; Baz, A.; Al-Turjman, F. Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure. *IEEE Access* **2020**, *8*, 101235–101243. [CrossRef]
21. Badar, H.M.S.; Qadri, S.; Shamshad, S.; Ayub, M.F.; Mahmood, K.; Kumar, N. An identity based authentication protocol for smart grid environment using physical uncloneable function. *IEEE Trans. Smart Grid* **2021**, *12*, 4426–4434. [CrossRef]
22. Bera, B.; Saha, S.; Das, A.K.; Vasilakos, A.V. Designing blockchain-based access control protocol in IoT-enabled smart-grid system. *IEEE Internet Things J.* **2020**, *8*, 5744–5761. [CrossRef]