Hindawi
International Journal of Distributed Sensor Networks
Volume 2023, Article ID 2524051, 12 pages
https://doi.org/10.1155/2023/2524051

Research Article

Analysis and Design of Identity Authentication for IoT Devices in the Blockchain Using Hashing and Digital Signature Algorithms

Lei Wang, Ying Yuan, and Yan Ding

School of Electrical Engineering, Yellow River Conservancy Technical Institute, Kaifeng 475004, China

Correspondence should be addressed to Lei Wang; yrctiwl@163.com

Received 19 January 2023; Revised 10 October 2023; Accepted 25 October 2023; Published 15 November 2023

Academic Editor: Meng-Shiuan Pan

This paper proposes a blockchain-based identity authentication (BA) scheme for IoT devices to solve the authentication security problem of IoT devices. The BA scheme uses hashing and digital signature algorithms to achieve integrity and nonrepudiation of authentication messages. Blockchain technology is used to achieve decentralised and distributed storage and management of authentication data. Besides, the BA scheme uses the idea of trust domains and trust credentials to establish a master-slave connection between IoT devices. The BA scheme is then compared with the existing four schemes and analysed from six perspectives to show that the BA scheme has better security. Also, the results show that the BA scheme has reasonable computational and storage overhead. Finally, the advantages of the BA scheme over traditional centralised and existing blockchain-based authentication schemes are compared and analysed. The results show that it can perfectly solve the problem of overreliance on trusted third parties in traditional authentication schemes.

1. Introduction

In recent years, the development of sensing, computer control, embedded, and wireless network data communication technologies has enabled the emergence of the Internet of Things (IoT) [1]. This technology is now widely applied in various areas, such as smart grids [2], smart transportation [3], smart homes, smart healthcare facilities [4, 5], environmental monitoring [6], and industrial automation [7]. The edge nodes of the IoT are referred to as IoT devices, which vary from mobile payment handheld devices, surveillance devices for public places, home devices for smart homes, wearable devices for health services [8], and more. This growth is driven by current hardware products, such as ever-improving sensors and processors, advancing low-power narrowband networks such as LoRa [9] and NB-IoT, and evolving advanced technologies such as artificial intelligence and machine learning.

However, the massive number of IoT end devices accessing the IoT system for information exchange and data communication has exposed numerous security issues [10]. In particular, terminal devices that store and transmit important data to users can be attacked and become vulnerable to data leakage, leading to unnecessary damage and loss of privacy. Therefore, the security and privacy of IoT devices must be considered [11], mainly involving authentication [12], access control, data protection [13], trust management, and other issues. Identity authentication [14] is the first step of the whole IoT security, which is a security mechanism that identifies and authenticates the identity of devices that access the IoT system before data interaction takes place. On the one hand, the authentication mechanism can ensure that IoT devices in the system can use their legitimate identities to establish a trust relationship for end-to-end secure data communication. On the other hand, it can restrict the access of illegal devices to the IoT system to avoid a series of security risks, thus making the whole system safe and reliable.

Currently, authentication schemes [15] are usually based on a centralised system where a trusted third party confirms the legitimacy of an IoT device's identity. This centralised authentication approach has become increasingly unsuitable for the IoT ring with its wide range of devices and complex network structure [16]. Moreover, there is a potential single

point of failure in this approach. If the centre fails or is attacked, not only will the whole IoT system fail to operate normally, but there is even a security issue of private data leakage.

The current authentication schemes for IoT devices are mainly divided into symmetric key-based authentication, public key infrastructure- (PKI-) based authentication [17], and identity-based cryptograph- (IBC-) based authentication [18]. Symmetric key-based authentication is not a centralised authentication method and has the advantages of low computation and high efficiency, but it requires preshared symmetric keys and thus has problems such as key distribution and key leakage. PKI-based authentication and IBC-based authentication are both centralised authentication methods, and PKI-based authentication requires the establishment of a certificate authority centre to issue and query the device's certificate, so there is a cumbersome certificate management process. IBC-based authentication is based on traditional PKI and can solve the certificate management and delivery problem, but this type of solution not only requires a trusted third party to generate a private key for the device but also generally requires complex operations such as two-line pairs, which has a large computational overhead.

Blockchain technology uses a cryptographic chain-like block structure to verify and store data and a distributed node consensus algorithm to generate and update data, with features such as decentralisation, data tamper-proof forgery, and traceability [19]. Blockchain perfectly solves the problem of how distributed nodes that do not trust each other can reach a consensus state in a decentralised peer-to-peer network architecture [20], providing a new way of thinking about IoT security. Although blockchain technology was initially applied mainly in the financial services sector, in the past years, many researchers [21, 22] have tried to introduce blockchain technology into the IoT to solve the security problems faced.

Based on the above background, this paper proposes an IoT device identity authentication solution based on blockchain technology. Specifically, by leveraging the tamper-evident and traceability of blockchain, the key data to confirm the legitimacy of a device is stored in a blockchain ledger jointly maintained by multiple distributed blockchain nodes without being generated and managed by a trusted third party. The ultimate expectation is to build a decentralised IoT device identity authentication architecture that achieves bidirectional device-to-device identity authentication without the intervention of a trusted third party so that devices can verify each other's identities. The aim is to establish a trust relationship for the next step of data interaction.

The main work of the paper is: The paper presents a four-phase identity authentication scheme (BA scheme) that includes system initialization, trust domain creation, association of trust domains, and authentication. It is compared with existing blockchain-based authentication schemes, highlighting its better security in terms of two-way authentication, privacy protection, traceability, forgery attacks, power abuse, and replay attacks. The scheme is also shown to have low computational and storage overheads. Additionally, it is compared with centralised authentication solutions, demonstrating its ability to solve the problem of overreliance on trusted third parties. The paper introduces hash algorithms, digital signature algorithms, blockchain, and smart contract content in the second part; designs a comprehensive authentication scheme in the third part; analyses them in terms of security, computational and storage overheads, and comparison with traditional schemes in the fourth part, respectively; and concludes with the corresponding conclusions.

2. State of the Art

2.1. Hashing Algorithm. The hashing algorithm is a mathematical function that garbles data and makes it unreadable [23]. A hashing algorithm is a cryptographic hash function. The rules for mapping an arbitrary-length binary value string to a fixed-length binary value string are the hash algorithm. The resulting binary value string is the hash value after the original data has been mapped.

Common hashing algorithms [24] include MD5, SHA1, SHA-2, NTLM, and LANMAN, which are part of the secure hash algorithm family published by the National Institute of Standards and Technology (NIST). Hashing algorithms are one-way programs, making it impossible to unscramble and decode the data. Among these algorithms, this paper selects the Keccak hash algorithm [25] as the method. Keccak is a versatile cryptographic function. Although best known as a hash function, it can nevertheless also be used for authentication, (authenticated) encryption, and pseudo-random number generation. Its structure is an extremely simple sponge construction, and internally it uses the innovative Keccak-f cryptographic permutation.

2.2. Digital Signature Algorithm. The digital signature algorithm (DSA) [26] is a public-key cryptosystem and Federal Information Processing Standard for digital signatures. It functions on the framework of modular exponentiation and discrete logarithmic problems, which are difficult to compute. It generates a digital signature from two 160-bit values using mathematical functions, with the private key and message digest used to create these numbers. DSA is a variant of the Schnorr and ElGamal signature schemes and is distinct from RSA, which relies on prime number factorization for secure communication and digital signatures.

ECDSA (Elliptic Curve Digital Signature Algorithm) [27] is a cryptographically secure digital signature scheme based on elliptic-curve cryptography. It relies on the difficulty of the ECDLP problem (elliptic-curve discrete logarithm problem), and its private key size is about twice the security level in bits. It is used by cryptocurrency traders to prove their identities and can be used to improve performance on the Internet. ECDSA is an emulation of the digital signature algorithm (DSA) using the elliptic curve cipher (ECC).

2.2.1. The Signing Process Is as Follows

(a) Choose an elliptic curve $Ep(a, b)$ and a base point G

(b) Select the private key $k(k < n)$ (n is the order of G) and compute the public key $K = k \bullet G$ using the base point G

(c) Generate a random integer $r(r < n)$ and compute the point $R = r \bullet G$

(d) Use the original data and the coordinate values x and y of the point R as parameters to compute SHA1 as the hash value, i.e., Hash = SHA1 (original data x and y).

(e) Compute the following:

$$s \equiv r - \text{Hash} * k (\text{mod } n), \tag{1}$$

where r and s are the signature values. If one of r and s is 0, reexecute from Step 3.

2.2.2. The Validation Process Is as Follows

(a) After receiving the message (m) and the signature value (r, s), the receiver performs the following operations

(b) Compute

$$\begin{aligned} sG + H(m)P &= (x_1, y_1), \\ r_1 &\equiv x_1 \bmod p. \end{aligned} \tag{2}$$

(c) Verify

$$r_1 \equiv r \bmod p. \tag{3}$$

(d) If Equation (3) holds, accept the signature. Otherwise, the signature is invalid

2.3. Blockchain Technology. Blockchain technology [28] integrates various components such as cryptography, peer-to-peer (P2P) networks, consensus algorithms, and smart contracts and is characterized by decentralisation, tamper-proofing, and traceability. At present, applications based on blockchain technology mostly use the distributed storage of blockchain and its properties such as tamper-proofing and traceability to achieve credible evidence and traceability. Among the many blockchain systems, Bitcoin, Ether, Hyperledger Fabric [29], and FISCO BCOS [30] are the more established and relatively well-studied public and federated chain systems. In this paper, Ether is chosen as the main system due to its suitability for the architectural system required. This system is mainly described as follows.

Ether's data storage structure is mainly based on Merkle Patricia Trees (MPT), which have distinct prefix identification and are particularly suitable for storing data in key-value pairs. Consequently, Ether utilizes LevelDB [31] as its main data storage system, as illustrated in Figure 1. The LevelDB stores block data, account data, receipt data, and index data. The block database is comprised of the block header and the block body, with the block header storing the previous block hash and information such as the root value and random numbers of the state tree, transaction tree, and receipt tree, and the block body mainly storing transaction data and block data.

Unlike Bitcoin, Ether does not rely on interblock connections when executing transactions and validating data. This allows Ether to use LevelDB to perform these functions efficiently and consume less storage space.

2.4. Ethernet-Based Smart Contract Design. Smart contracts are programs stored on a blockchain that run when predetermined conditions are met [32]. They are typically used to automate the execution of an agreement so that all participants can be immediately sure of the outcome without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met. Ethereum differs from blockchain platforms such as Bitcoin because the Ethereum Virtual Machine (EVM) [33] provides a Turing-complete operating environment for smart contracts. Smart contracts in Ether are composed of binary bytecodes (also known as EVM bytecodes). Then, EVM compiles smart contracts written in the high-level programming languages Solidity, Serpent, or LLL into EVM bytecode and stores them in the block with the smart contract address.

The process of deploying and invoking a smart contract on the Ethernet platform is illustrated in Figure 2.

The specific steps are illustrated as follows:

Step 1. Start an Ethernet node.

Step 2. Write the smart contract using a high-level programming language (typically Solidity).

Step 3. Convert the code for writing a smart contract into EVM bytecode.

Step 4. Deploy the smart contract on the blockchain network. After confirmation by the blockchain node, the EVM code will be stored in the blockchain, and the participants will get the returned contract address and interface information.

Step 5. Invoke the contract using the JSON RPC interface.

3. A Blockchain-Based Identity Authentication Scheme Design

A blockchain-based authentication scheme is designed by combining the digital signature algorithm, hash algorithm, and blockchain technology described in Section 2. This paper outlines the system model of the scheme and the scheme flow and describes the scheme in detail in four phases: system initialization, creation of trust domains, the association of trust domains, and identity authentication.

3.1. Scheme Overview. The scheme proposed in this paper is mainly based on the digital signature algorithm, hashing algorithm, and blockchain technology. Among them, the significance of introducing a digital signature and a hashing algorithm is to sign and verify the data messages between devices to ensure the integrity and nonrepudiation of data messages. The significance of introducing blockchain technology is to store the key data information in the

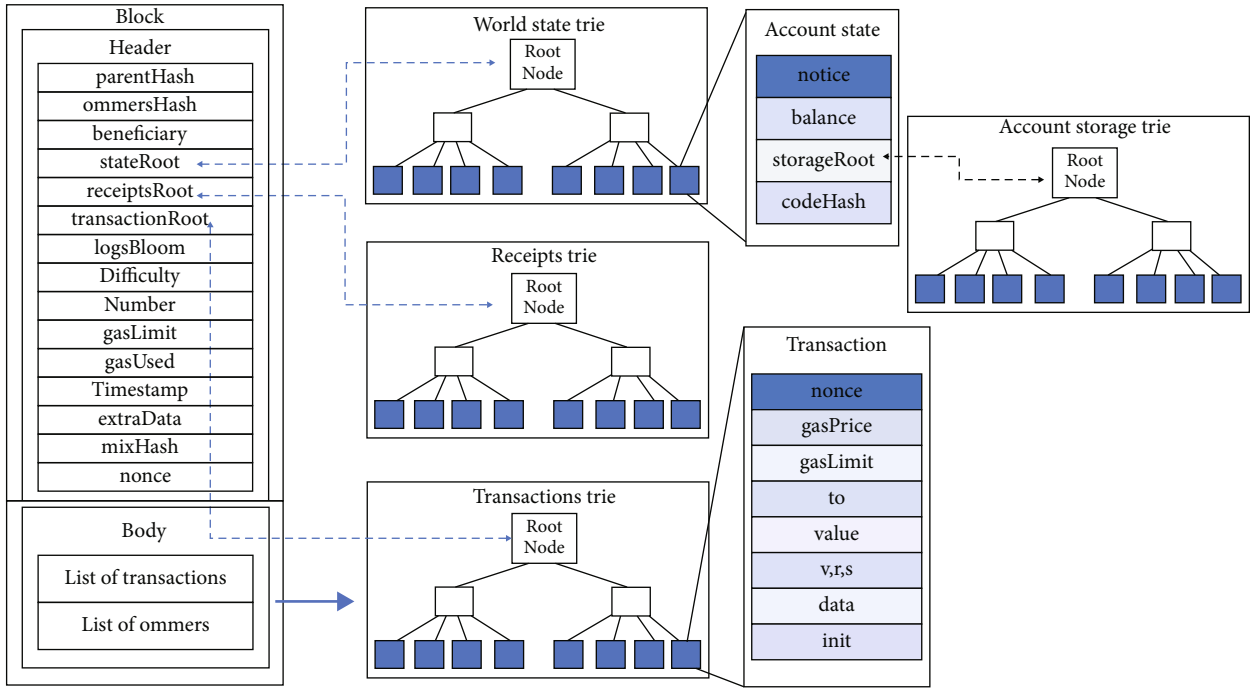

Figure 1: Ether's data storage structure.

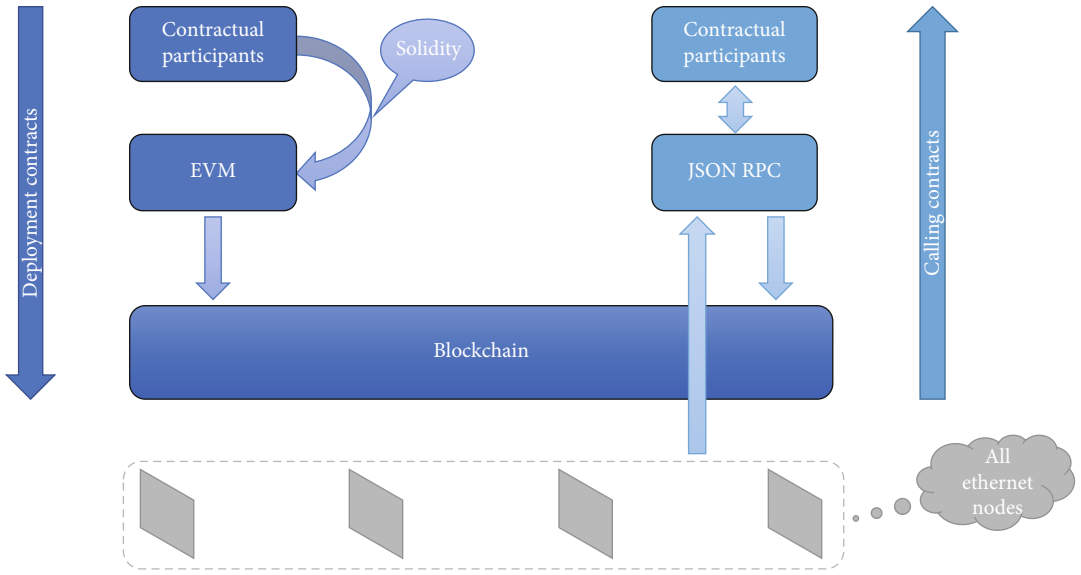

Figure 2: The deployment and invocation process for Ethernet smart contracts.

authentication mechanism in the form of blockchain transactions in a distributed ledger maintained by multiple blockchain nodes to ensure the nontamperability and traceability of the data information and ultimately to realise decentralised authentication. This section will provide an overview of the solution in terms of the system model and solution flow.

3.2. System Model. The system model of the program is divided into three parts from the bottom up, namely, the IoT trust domain, the blockchain network, and the blockchain, which are described as follows.

3.2.1. IoT Trust Domain. Each trust domain comprises two device roles, which can be differentiated according to specific scenarios. Each trust domain consists of two device roles, namely, master device and slave device, where the number of master devices is unique and the number of slave devices is not limited. The master device role here is similar to that of a certificate authority in a PKI architecture, generating trust credentials for the slave devices in a specific trust domain. In the context of designating which devices can become the master device within a trust domain, this process occurs as a strategic procedure during system initialization. This procedure can be predefined based on the requirements

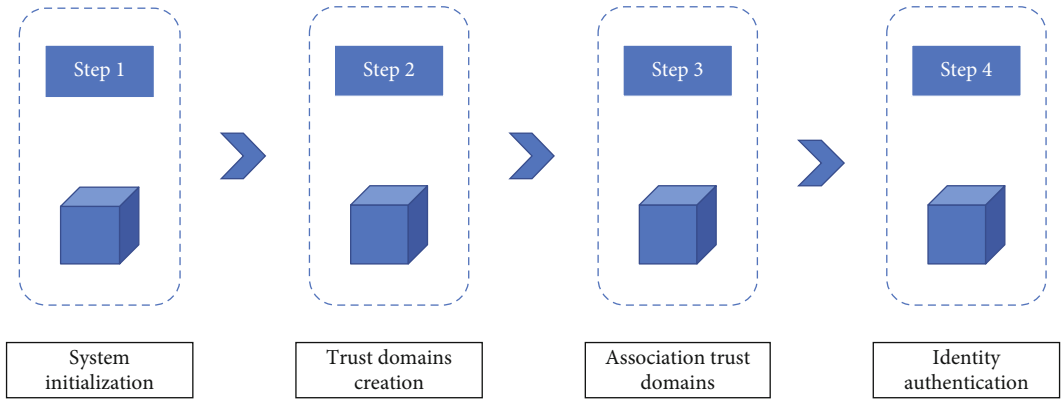

Figure 3: The blockchain-based IoT device identity authentication scheme.

and characteristics of the IoT environment in which the implementation takes place. Devices considered for assuming the role of the master device typically possess specific capabilities or attributes that render them suitable for this role. These attributes may encompass computational prowess, security functionalities, or administrative privileges. Additionally, trustworthiness plays a pivotal role. The chosen master device should inherently exhibit trustworthiness within the IoT ecosystem.

3.2.2. Blockchain Network. A blockchain network is maintained by multiple blockchain nodes, which can be acted upon by hosts, servers, etc. It is generally considered to be computationally powerful, always secure, and trustworthy. The blockchain nodes are mainly used to receive and verify the data information generated at the IoT device side and encapsulate it into corresponding data blocks for consensus, which will be linked to the previous block when the consensus is completed, thus forming the latest blockchain. The type of blockchain nodes employed in our proposed authentication scheme involves dedicated hosts positioned near IoT devices. These dedicated hosts serve as essential components within the IoT network and are instrumental in the functioning of our authentication framework. Specifically, they ensure the security and integrity of transactions within the IoT network.

3.2.3. Blockchain. The data blocks are combined in a chain in a specific data structure in chronological order and kept in each blockchain node, whose role is to store and record data information to authenticate IoT devices.

3.3. Solution Flow. The flow of this scheme is shown in Figure 3. The blockchain-based IoT device authentication scheme will be designed in four aspects: the system initialization phase, the trust domain creation phase, the association trust domain phase, and the identity authentication phase. Among them, the system initialization phase serves to divide the trust domain and determine the master and slave devices. The master device signs trust credentials for the slave devices. The trust domain creation phase is targeted at the master device, which requests the blockchain to create a trust domain. The associated trust domain phase is primarily for the slave device and is where the slave device uses the trust credentials to request the associated trust domain from the blockchain. The identity authentication phase is to confirm the identity of the slave device for the subsequent data interaction.

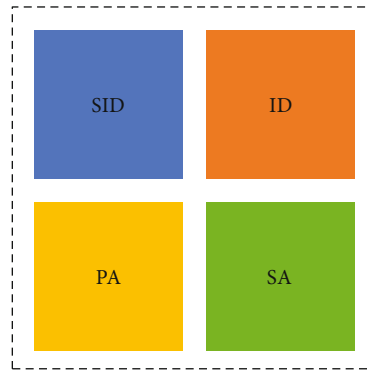

Figure 4: The structure of a trust credential ticket.

3.3.1. The System Initialization Phase Identifies a Unique Master Device for the Trust Domain. The master selects the trust domain identifier ID to be created and signs a trust credential ticket for the domain for all slave devices in the domain, the structure of which is shown in Figure 4:

(a) Trust domain identifier ID: this is the identifier of the trust domain to which the slave device belongs, is used to distinguish between multiple trust domains, occupies 1 byte, and is unique

(b) Slave device identifier SID: identifier of the slave device, used to distinguish multiple slave devices, occupies 1 byte, and is unique

(c) Slave address PA: the address of the slave device, the public key Pb of the slave device is hashed, and the first 20 bytes are taken to obtain the ID and SID for authentication of the slave device, with uniqueness

(d) Master device signature SA: the signature of the master device on the trust credential, signed by the

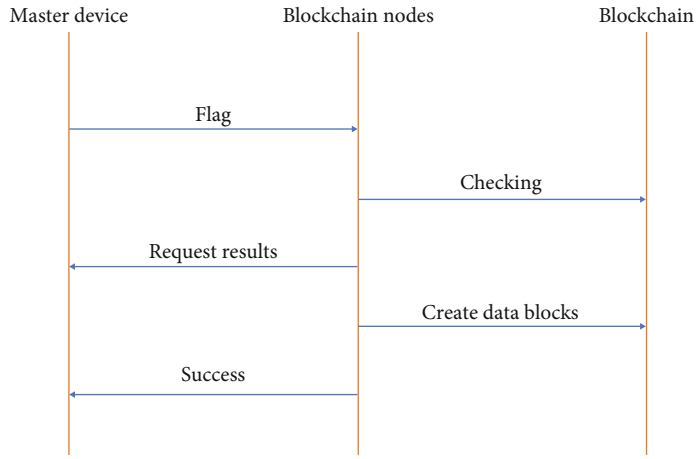

Figure 5: The exact flow of the trust domain creation.

master device's private key Pr on the ID, SID, and PA hash value of the concatenation, accounting for 64 bytes, with uniqueness

For the device object, the following points are noted:
For master devices:

(i) There is only one master device per trust domain

(ii) The trust domain identifier created must be unique and cannot conflict with other trust domain identifiers

(iii) Only the act of creating a trust domain and signing trust credentials is performed

For slave devices:

(i) They can only belong to a specific trust domain

(ii) They cannot play the role of a master device to create new trust domains

(iii) They only perform the act of associating trust domains and data interaction

For both master and slave devices:

(i) The identifiers of both must be unique

(ii) The public keys of both must be unique

(iii) Data interaction can only be performed in the same trust domain

(iv) All data messages and blockchain transactions generated must be verified by signatures

3.3.2. In the Trust Domain Creation Phase, the Master Device Sends a Flag Bit (Flag) to the Blockchain Node for the Blockchain Transaction for the Trust Domain Creation Request. When the blockchain node receives the transaction, it uses the master device's public key to sign and verify the integrity of the transaction. After the verification passes, it then queries whether the data block containing the ID and MID already exists in the blockchain, and if it does not, the creation request is passed. When the creation request is passed, the blockchain node encapsulates the requested ID transaction into the corresponding creation data block and sends it to other blockchain nodes, which store the transaction in a new block through the corresponding consensus algorithm to form the latest blockchain. The trust domain identifier ID and the master device identifier MID are stored in the blockchain. Finally, the blockchain node returns a successful creation message to the master device. The exact flow of the trust domain creation phase is shown in Figure 5.

3.3.3. In the Association Trust Domain Phase, the Slave Device Sends a Blockchain Transaction with the Flag Bit Flag as the Association Trust Domain Request to the Blockchain Node for Trust Domain Association. When the blockchain node receives the transaction, it uses the public key of the slave device to sign and verify the integrity of the transaction. After the verification, it then queries whether the data block containing the SID already exists in the blockchain and checks the ticket's validity using the public key of the master device in the trust domain. The association request is passed if the data block does not exist and the ticket is valid. When the association request is passed, the blockchain node encapsulates the requested transaction into the corresponding associated data block and sends it to other blockchain nodes, where the transaction is stored in a new block by the corresponding consensus algorithm to form the latest blockchain. At this point, the hash value from the device identifier SID, the trust domain identifier that the slave device is in the ticket, and the address PA string of the slave device are stored in the blockchain. Finally, the blockchain node returns a successful association message to the slave device. The exact flow of the association trust domain phase is shown in Figure 6.

3.3.4. Identity Authentication. The first step in the identity authentication process must be the association trust domain phase. The first transaction sent by the slave device to the blockchain node must be a trust domain request transaction.

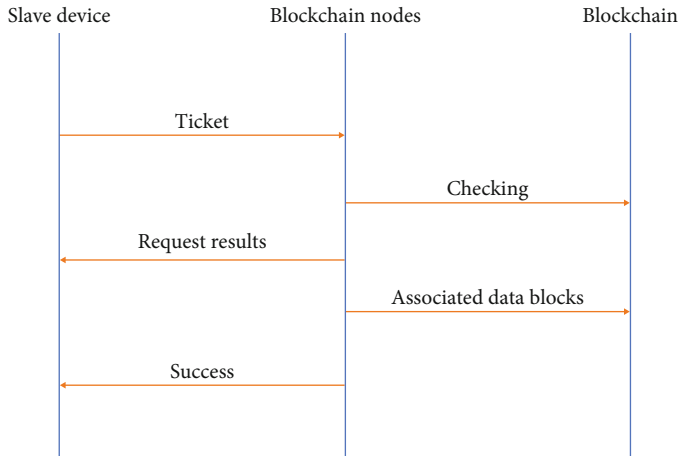

Figure 6: The exact flow of the association trust domain.

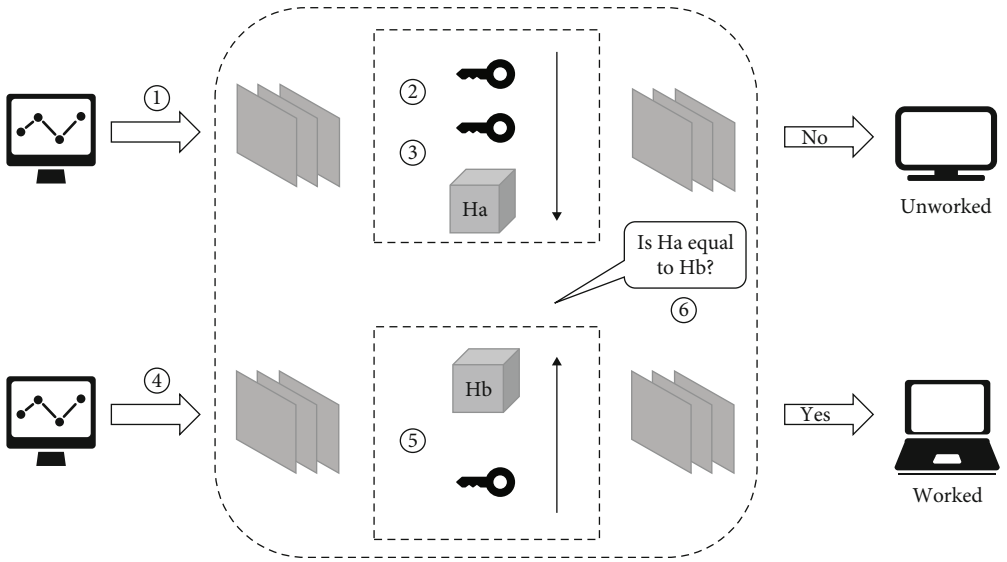

Figure 7: The identity authentication scheme. Notes: timestamp, flag, and signature SA are omitted from the transactions in all processes of this scheme.

Once the association is successful, this slave device does not need to use a ticket to verify its identity.

Take a slave device with a ticket provided by the master device as an example and interact with another slave device in the same domain, which requires authentication first. The identity authentication scheme includes 6 steps shown in Figure 7.

The authentication scheme described in the paper involves several steps to ensure secure communication between devices within a trust domain. The first step involves the slave device sending a signed ticket, which includes its ID and MID, to the blockchain node, requesting access to the associated trust domain. The ticket is signed with the slave device's private key Pr to ensure its authenticity. In the second step, the blockchain node uses the slave device's public key Pb to authenticate the transaction and ensure that it was indeed signed by the slave device's private key Pr.

Once the transaction is authenticated, the blockchain node uses the public key Pb of the master device in the trust domain to authenticate the ticket in Step 1. This authentication is necessary to ensure that only devices within the trust domain can access it. The ID, SID, and hash values from Step 1 are stored in the blockchain node after successful authentication.

In the next step, the slave device interacts with other slave devices within the same domain and generates data to be sent to the blockchain node for data interaction requests. This data is signed with the slave device's private key Pr and sent to the blockchain node for authentication. The blockchain node uses the slave device's public key Pb to authenticate the transaction in Step 5.

After the transaction is authenticated, the ID and SID of the transaction are extracted. The blockchain node then uses the public key Pb of the slave device to extract the hash value of the device and compare it with the hash value stored in Step 3. If they match, the authentication is successful, and data interaction with other slave devices within the same trust domain is allowed.

Input: IoT trust domain configuration, blockchain network, master device information, slave device information, and data for identity authentication.
Output: Identity authentication result.
① System Initialization: initialize the IoT trust domain; identify the master and slave devices.
② Generate Trust Credentials: master device generates trust credentials for slave devices; trust credentials include ID, SID, PA, and SA.
③ Trust Domain Creation: master device requests the blockchain to create a trust domain; the blockchain node verifies the request and creates the trust domain; trust domain data (ID and MID) is stored in the blockchain.
④ Association Trust Domain: slave device requests association with a trust domain; blockchain node verifies the request using the public key; if valid, association data (SID, ID, and PA) is stored in the blockchain.
⑤Identity Authentication: slave device initiates identity authentication; authentication includes multiple steps: (a) slave device signs a ticket; (b) blockchain node verifies the ticket; (c) slave device interacts with other devices; (d) Data is signed, sent, and verified; (e) hash values are compared; and (f) if successful, data interaction is allowed.
⑥ Aggregate Signature: multiple signatures can be aggregated into one using an algorithm; blockchain node verifies the aggregated signature; authentication continues as usual.

Algorithm 1: The authentication method proposed in this paper.

3.3.5. Aggregate Signature Authentication Scheme. To implement aggregate signature authentication in the ECDSA-based identity authentication scheme described in the steps provided, multiple signatures from different devices can be aggregated to form a single signature. The steps to achieve this are as follows:

(1) Each slave device signs the ticket with its own private key and generates a signature Pr

(2) The signatures from all slave devices are aggregated into a single signature using an aggregate signature algorithm

(3) The blockchain node verifies the aggregated signature using the public keys of all the slave devices

(4) If the verification is successful, the blockchain node proceeds with the authentication process as described in Steps 3–6 of the original authentication scheme

It is important to note that the choice of aggregate signature algorithm can affect the security and efficiency of the authentication scheme. One widely used aggregate signature algorithm for ECDSA is the Boneh-Lynn-Shacham (BLS) [34] signature algorithm, which allows for efficient aggregation of signatures without compromising security.

In summary, the steps to implement aggregate signature authentication in the ECDSA-based identity authentication scheme involve aggregating the signatures from multiple devices into a single signature using an aggregate signature algorithm and then verifying the signature using the public keys of all devices.

To sum up, the flow of the authentication scheme for IoT devices in this paper based on blockchain technology, combined with the Keccak hash algorithm and the ECDSA digital signature algorithm, is shown in Algorithm 1.

4. Analysis

4.1. Security Analysis

4.1.1. Two-Way Authentication. Our scheme confirms the identity of each slave device in each trust domain through a ticket and blockchain. This is achieved through digital signature algorithms to ensure message integrity and authenticity.

Two-way authentication establishes trust relationships between devices, ensuring the legitimacy of both parties' identities. This helps prevent man-in-the-middle attacks and impersonation attacks.

4.1.2. Privacy Protection. The ticket is passed during system initialization, similar to a third-party authority certificate. It is only passed during this phase, making it impossible for illegal devices to forge a correct ticket.

This step ensures that critical credentials are only passed during initialization, preventing identity deception by unauthorized devices and thus providing privacy protection.

4.1.3. Traceability. All generated data is permanently recorded in the form of blocks on the blockchain, with each data block containing all information, ensuring the traceability of information.

Traceability helps detect abnormal behavior and security incidents. In the event of an incident, its source can be traced, making it easier to respond to potential threats.

4.1.4. Resistance to Forgery Attacks. Every stage of our scheme relies on digital signature algorithms, ensuring that attackers cannot initiate or execute transactions due to a lack of device private keys.

This effectively prevents forgery attacks, ensuring data integrity and identity authenticity, thereby enhancing the overall security of the system.

4.1.5. Resistance to Power Abuse. All proposed transactions are publicly and immutably recorded in the blockchain; each transaction is signed with a private key, and a consensus

Table 1: The comparison results between the five schemes.

	KPSD	IBCCPA	EAAP	EPAW	BA
Two-way authentication	×	√	√	√	√
Privacy protection	√	√	√	√	√
Traceability	×	√	√	×	√
Resistance to forgery attacks	√	√	√	√	√
Resistance to power abuse	×	×	×	×	√
Resistance to replay attacks	×	√	×	√	√

algorithm is used to achieve consistency of data information across the blockchain ledger. Any changes to the data on a blockchain node are detected. As seen in Section 4.1.2, transactions and changes cannot be made without the private key in the ticket.

This design ensures transparency of data and consistency of data information, preventing power abuse and unauthorized access.

4.1.6. Resistance to Replay Attacks. For resistance to replay attacks, all blockchain transactions in this solution are passed as encapsulated blocks of data with timestamps. Once validated, they are stored in the blockchain and cannot be replayed by the attacker.

This helps prevent attackers from resending previous transactions, ensuring the one-time nature of transactions.

Based on the above analysis, the proposed scheme in this paper is noted as the BA scheme, and the results of this scheme are compared with the other four schemes (EPSD [35], IBCCPA [36], EAAP [37], and EPAW [38]) and shown in Table 1. After screening the existing authentication mechanisms, these four schemes are widely used as authentication methods.

Table 1 compares six key security attributes, including two-way authentication, privacy protection, traceability, resistance to forgery attacks, resistance to power abuse, and resistance to replay attacks, where √ indicates that the solution under consideration satisfies a specific security attribute and × indicates that the solution under consideration does not satisfy a specific security attribute. The KPSD and EPAW schemes do not involve the disclosure of the identity of entities within the architecture and therefore do not satisfy the traceability property. None of the above schemes considers authority constraints, so they do not satisfy the resistance to authority abuse. From the comparison results, it can be seen that the scheme in this paper satisfies more security attributes and is more secure than the other schemes.

4.2. Calculation Overhead Analysis. Since the blockchain node is acted upon by the host, its computational overhead is not considered, and the computational overhead on the IoT device side is mainly considered. Based on the previously proposed authentication process, it is clear that one authentication process requires up to two signed transactions to be sent to the blockchain node, i.e., a maximum of two signature operations are required. As the ECDSA signature algorithm is used in this scheme, the computational overhead at the IoT device side can be known according to the principal generation process of the signature algorithm shown as follows:

$$T_C = 2T_E + 2T_H + 4T_M, \tag{4}$$

where T_C is the total operation, T_M is the modulo operation, T_H is the hashing operation, and T_E is the ECDSA operation.

4.3. Storage Overhead Analysis. The storage overhead mainly comes from the blockchain node side and the IoT device side. The total storage overhead algorithm is shown in the following:

$$S_T = S_I + S_B, \tag{5}$$

where S_T is the total storage overhead, S_I is the IoT device-side storage overhead, and S_B is the blockchain node side storage overhead.

Firstly, the storage overhead on the IoT device side consists of its own private key, ID, SID, and ticket, and the total storage is about 100 bytes.

For the blockchain node side, it includes the stored information content of three requests: trust domain creation, associated trust domain, and data interaction. For the first two types of requests, the length of information included is about 200 bytes, while for data interaction, the information length of the actual interaction data needs to be considered, which can be assumed to be *Ldata*. Then, the total length of data information for storage overhead is 300 + *Ldata* bytes.

4.4. Comparison with Traditional Authentication Schemes

4.4.1. Difficulty in Tampering with Data. The proposed scheme in this paper requires more arithmetic power to crack, and the attacker has a small probability of success in attacking nodes in the blockchain network to achieve tampering with the data by means of adapting blocks of data. This is the biggest advantage of this paper's blockchain-based authentication scheme over traditional centralised-based authentication schemes.

4.4.2. No Specific Trusted Third-Party Organisation Is Required. Currently, most traditional authentication schemes for IoT devices are based on centralised authentication methods. The authentication scheme proposed in this paper, however, leverages the decentralised nature of blockchain, uses distributed data storage for secure storage, and does not rely on a specific trusted third-party organisation.

This can avoid the risk of centralised structures being vulnerable to centralised malicious attacks.

4.4.3. Consistency of Data Information. Blockchain nodes encapsulate key data information used for device authentication into blocks of data at various stages and have this data permanently recorded in the blockchain distributed ledger by the remaining blockchain nodes through consensus algorithms. This allows it to be utilized for retrospective queries. The consensus algorithm achieves the consistency of data information in the blockchain ledger so that if a blockchain node's data information has been tampered with, it will be detected at the next consensus. This ensures that the data information queried by the device during the authentication phase is consistent with the previous phases.

5. Conclusion

In view of the lack of research solutions for introducing blockchain technology to solve the authentication of IoT devices and to address the problems of traditional centralised authentication solutions and the shortcomings of existing blockchain-based authentication solutions, this paper still proposes an authentication solution for IoT devices with blockchain as the technical support, combined with the Keccak hash algorithm and the ECDSA digital signature algorithm. This paper focuses on a blockchain-based authentication scheme and presents the system model of the authentication scheme, the scheme flow, and a comprehensive analysis of it, respectively. The details are described as follows.

(i) The paper describes the designed identity authentication scheme (BA scheme) in detail in terms of the scheme flow. This consists specifically of four phases: system initialization, the creation of trust domains, the association of trust domains, and authentication. The creation of the trust domain is the prerequisite for the association of the trust domain, while the association of the trust domain is a prerequisite for authentication

(ii) This paper compares the proposed scheme with existing blockchain-based authentication schemes and analyses the scheme in terms of two-way authentication, privacy protection, traceability, resistance to forgery attacks, resistance to power abuse, and resistance to replay attacks. Six perspectives show that it has better security, and an aggregate signature authentication scheme is proposed

(iii) The analysis also shows that this scheme has good computational and storage overheads

(iv) Finally, this paper compares and analyses the advantages of this solution with traditional centralised authentication solutions and shows that it can perfectly solve the problem of overreliance on trusted third parties in traditional authentication solutions

In conclusion, the IoT device identity authentication solution based on blockchain technology presented in this paper is a promising solution for ensuring the security and privacy of IoT devices. The solution not only solves the problems of traditional authentication schemes for IoT devices, such as key distribution and key leakage but also provides a secure and reliable identity authentication platform for IoT devices. In the future, more research and development should be done to improve the solution and make it more applicable to the real world.

Despite the innovative aspect of our study, there are still areas for further research and improvement:

(1) In terms of the authentication mechanism, only two-way authentication between devices within the same trust domain is currently taken into account, so in the next research work, we will focus on considering cross-domain authentication between multiple trust domains to make the authentication mechanism more comprehensive

(2) Also, the revocation stage is not currently included in the authentication process, so in the next research work, the revocation request blockchain transaction will be constructed to scrap the authentication data information of faulty or damaged devices

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Authors' Contributions

Lei Wang and Ying Yuan contributed to the writing of the manuscript and data analysis. Yan Ding supervised the work and designed the study. Lei Wang is the corresponding author. Ying Yuan provided great help in the revision of the final draft. All unanimously agreed to the above arrangement. All the authors have read and agreed on the final version to be published.

Acknowledgments

This work is supported by the On the Characteristics of Printed Circuit Board Internal Single-Layered Antenna (No. 21B510005).

References

[1] B. Acko, H. Weber, D. Hutzschenreuter, and I. Smith, "Communication and validation of metrological smart data in IoT-networks," *Advances in Production Engineering & Management*, vol. 15, no. 1, pp. 107–117, 2020.

[2] S. A. A. Abir, A. Anwar, J. Choi, and A. S. M. Kayes, "Iot-enabled smart energy grid: applications and challenges," *IEEE Access*, vol. 9, pp. 50961–50981, 2021.

[3] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and IoT in smart transportation," *Future Internet*, vol. 11, no. 4, p. 94, 2019.

[4] A. Ali, M. A. Almaiah, F. Hajjej et al., "An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network," *Sensors*, vol. 22, no. 2, p. 572, 2022.

[5] A. Ali, H. A. Rahim, M. F. Pasha et al., "Security, privacy, and reliability in digital healthcare systems using blockchain," *Electronics*, vol. 10, no. 16, p. 2034, 2021.

[6] V. Tanasiev, G. C. Pătru, D. Rosner, G. Sava, H. Necula, and A. Badea, "Enhancing environmental and energy monitoring of residential buildings through IoT," *Automation in Construction*, vol. 126, p. 103662, 2021.

[7] Z. Fatima, M. H. Tanveer, Z. S. Waseemullah et al., "Production plant and warehouse automation with IoT and industry 5.0," *Applied Sciences*, vol. 12, no. 4, p. 2053, 2022.

[8] M. A. Almaiah, F. Hajjej, A. Ali, M. F. Pasha, and O. Almomani, "A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS," *Sensors*, vol. 22, no. 4, p. 1448, 2022.

[9] R. S. Sinha, Y. Wei, and S. H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017.

[10] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, and V. R. Kebande, "A review of security standards and frameworks for IoT-based smart environments," *IEEE Access*, vol. 9, pp. 121975–121995, 2021.

[11] F. Alshohoumi, M. K. Sarrab, A. AlHamadani, and D. Al-Abri, "Systematic review of existing IoT architectures security and privacy issues and concerns," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 232–251, 2019.

[12] T. D. Dinh, T. D. Le, K. Q. Dang, V. Vishnevsky, and R. Kirichek, "Blockchain-driven hybrid model for IoT authentication," in *International Conference on Next Generation Wired/Wireless Networking*, pp. 557–573, Springer Nature Switzerland, Cham, 2023.

[13] S. Divadari, J. Surya Prasad, and P. Honnavalli, "Managing data protection and privacy on cloud," in *In Proceedings of 3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2022*, pp. 383–396, Springer Nature Singapore, Singapore, 2023.

[14] J. Long and X. Su, "Anonymous chaotic-based identity authentication protocol in IoT," *International Journal of Embedded Systems*, vol. 14, no. 2, pp. 194–200, 2021.

[15] Y. Zhou, T. Liu, F. Tang, and M. Tinashe, "An unlinkable authentication scheme for distributed IoT application," *IEEE Access*, vol. 7, pp. 14757–14766, 2019.

[16] Y. Zhao, C. Lian, X. Zhang, X. Sha, G. Shi, and W. J. Li, "Wireless IoT motion-recognition rings and a paper keyboard," *IEEE Access*, vol. 7, pp. 44514–44524, 2019.

[17] Z. Siddiqui, J. Gao, and M. K. Khan, "An improved lightweight PUF–PKI digital certificate authentication scheme for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19744–19756, 2022.

[18] X. Jia, N. Hu, S. Yin, Y. Zhao, C. Zhang, and X. Cheng, "A2 chain: a blockchain-based decentralized authentication scheme for 5G-enabled IoT," *Mobile Information Systems*, vol. 2020, Article ID 8889192, 19 pages, 2020.

[19] J. P. Howard and M. E. Vachino, "Blockchain compliance with federal cryptographic information-processing standards," *IEEE Security & Privacy*, vol. 18, no. 1, pp. 65–70, 2020.

[20] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, "Peer-to-peer trading in electricity networks: an overview," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3185–3200, 2020.

[21] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," *Internet of Things*, vol. 1-2, pp. 1–13, 2018.

[22] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the IoT and industrial IoT: a review," *Internet of Things*, vol. 10, article 100081, 2020.

[23] F. E. De Guzman, B. D. Gerardo, and R. P. Medina, "Implementation of enhanced secure hash algorithm towards a secured web portal," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, Singapore, 2019.

[24] B. U. I. Khan, R. F. Olanrewaju, M. A. Morshidi, R. N. Mir, M. L. B. M. Kiah, and A. M. Khan, "Evolution and analysis of secured hash algorithm (SHA) family," *Malaysian Journal of Computer Science*, vol. 35, no. 3, pp. 179–200, 2022.

[25] H. Mestiri, F. Kahri, M. Bedoui, B. Bouallegue, and M. Machhout, "High throughput pipelined hardware implementation of the KECCAK hash function," in *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*, Tunis, Tunisia, 2016.

[26] L. X. Van and L. H. Dung, "Constructing a digital signature algorithm based on the difficulty of some expanded root problems," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, Hanoi, Vietnam, 2019.

[27] J. Doerner, Y. Kondi, E. Lee, and A. Shelat, "Secure two-party threshold ECDSA from ECDSA assumptions," in *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2018.

[28] Y. Cheng and H. Shaoqin, "Research on blockchain technology in cryptographic exploration," in *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, Bangkok, Thailand, 2020.

[29] T. Yan, W. Chen, P. Zhao, Z. Li, A. Liu, and L. Zhao, "Handling conditional queries and data storage on Hyperledger fabric efficiently," *World Wide Web*, vol. 24, no. 1, pp. 441–461, 2021.

[30] F. Ma, M. Ren, Y. Fu et al., "Security reinforcement for Ethereum virtual machine," *Information Processing & Management*, vol. 58, no. 4, article 102565, 2021.

[31] L. Wang, G. Ding, Y. Zhao, D. Wu, and C. He, "Optimization of LevelDB by separating key and value," in *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, Taipei, Taiwan, 2017.

[32] V. Singla, I. K. Malav, J. Kaur, and S. Kalra, "Develop leave application using blockchain smart contract," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, Bengaluru, India, 2019.

[33] Z. Yang, H. Liu, Y. Li, H. Zheng, L. Wang, and B. Chen, "Seraph: enabling cross-platform security analysis for EVM and WASM smart contracts," in *ICSE '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceeding*, New York, 2020.

[34] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *International conference on the theory and application of cryptology and information security*,

pp. 514–532, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[35] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: an effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[36] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1711–1720, 2016.

[37] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[38] S. Jegadeesan, M. Azees, N. R. Babu, U. Subramaniam, and J. D. Almakhles, "EPAW: efficient privacy preserving anonymous mutual authentication scheme for wireless body area networks (WBANs)," *IEEE Access*, vol. 8, pp. 48576–48586, 2020.