ARTICLE TEMPLATE

BUA: A Blockchain-based Unlinkable Authentication Scheme for Mobile IoT

Yihong Wen[a], Mingxi Liu[b], Xiwen Yang[b], Tailong Yang[b], and Victor Chang[c]

[a]Department of No.54 Research Institute of China Electronics Technology Group Corporation, 589 Zhongshan West Road, Shijiazhuang City, Hebei Province, China; [b]Department of School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China; [c]Operations and Information Management, Aston Business School, Aston University, Birmingham, UK

ARTICLE HISTORY
Compiled March 17, 2023

ABSTRACT
As a special example of the Internet of Things, mobile IoT has made a pioneering contribution to the mobile-oriented data acquisition business. Mutual authentication is performed to ensure the identity validity of data acquisition devices and mobile edge computing (MEC) servers in the mobile IoT. Mobile IoT data collection nodes need unlinkability functionality during device authentication to prevent the visibility of movement tracks. However, the present unlinkable authentication techniques have drawbacks: 1. the Certificate Authority (CA) is under many computational burdens; 2. the blockchain is under a lot of storage and throughput strain. In this paper, we propose a blockchain-based unlinkable authentication scheme named BUA, which has applied vector commitment to reduce dependency on CA while also compressing the data on the blockchain. We create a binary tree-based auxiliary proof index structure named AEST to reduce the multiplicative redundant operation that is faced by auxiliary computing evidence to verify vector commitment. According to security and performance analysis, this solution can fulfill the security criteria mentioned in the security goals. Compared to similar solutions, the performance of this solution has improved by 13.24%.

KEYWORDS
Mutual Authenticaion; Vector Commitment; Auxiliary Proof Calculation Index; Mobile IoT.

1. Introduction

The mobile Internet of Things (mIoT) system's main architecture is as follows: sensor equipment gathers environmental or user data, delivers it to edge nodes for caching and preprocessing, and then sends it to the cloud for activities like data mining and knowledge discovery (Zebin et al. 2019). The notion of mobile IoT is based on the Internet of Things (Lu and Da Xu 2018)(Mendez Mena et al. 2018). A wider range of mobile sensor devices may be included in the sensor network by implementing multi-

Both Mingxi Liu and Victor Chang are the corresponding authors (e-mail: mingxiliu larry@163.com, vchang1@aston.ac.uk/victorchang.research@gmail.com)

Yihong Wen's e-mail: yihwen@139.com; Xiwen Yang's e-mail: yxw03123@163.com; Tailong Yang's e-mail: yangtailong01@163.com

access edge computing networks (MEC), such as intelligent vehicles, smartphones, wearable sensor devices, etc. (Porambage et al. 2018). The number of smart vehicles in 2030 will be close to 40 million, according to an estimate of the market status and future prospects of China's intelligent car sector in 2021(industry research institute). As a result, the spike in sensor devices will have a significant influence on the mobile Internet of Things (Khan and Salah 2018).

Mobile edge computing is the first consideration to address the management issue of booming in sensor devices (Mehrabi et al. 2019). A representative mobile IoT paradigm is shown as figure 1. The most conspicuous characteristic of mobile IoT compared with generic IoT paradigm is the mobility of IoT devices. In generic IoT paradigm, the location of IoT devices is fixed, while in mobile IoT paradigm, we should consider that such as smart vehicles switch between edge devices. The number of connections to each MEC node has risen dramatically; also, more IoT devices are transmitting data to the MEC node at the same time, increasing the bandwidth stress (Sun and Ansari 2016). In the face of assaults, the dramatic rise in computing demand has resulted in a drop in fault tolerance(Hassija et al. 2019).

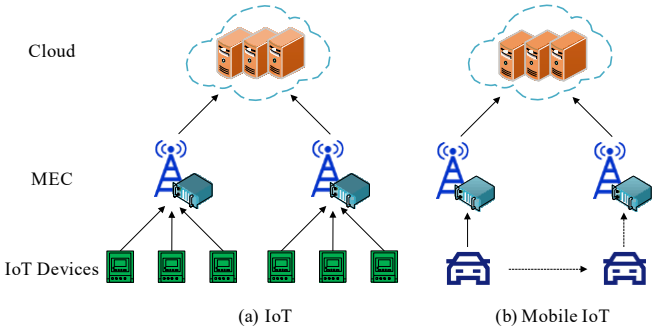

Figure 1. Significant difference between IoT and mobile IoT

If a MEC is disconnected owing to Byzantine fault tolerance, it will have an impact on the entire region's data transmission company. IoT devices may also upload private data to trustworthy intermediate nodes at the same time (Zhang et al. 2021). There will be privacy leaks if a middleman impersonates data user to get data.

There have been a lot of studies on the subject of applied cryptography. However, existing solutions are still inflexible in the face of mobile IoT scenarios (Morabito et al. 2018). As a result, authentication technology must be implemented to safeguard the security of communication between IoT devices and edge servers(Jia et al. 2019)(Li et al. 2020). To begin with, the Certificate Authority (CA) stores the root of trust for authentication, which means that when each mIoT device initiates authentication, the CA's participation is required. If the CA makes a Byzantine error in the face of a large number of mIoT devices, the entire authentication system is jeopardized. If CA offloads any of the computation and communication overheads to MEC, system stability will suffer. That is to say, if MECs connecting multiple mIoTs at the same time may exceed bandwidth capacity, which will be unable to authenticate (Rylands et al. 2021).

In recent years, several academics have developed blockchain-based authentication solutions to address the aforementioned flaws (Hammi et al. 2018). However, most of the solutions are underperforming owing to the blockchain's restricted on-chain resources (Yang et al. 2021). Reducing anonymity can directly enhance efficiency as a contradiction transfer tactic. The absence of anonymity in the mIoT situation will

expose the movement trajectory of mIoT devices, which is undesirable for devices like intelligent automobiles (Lin et al. 2019). Establishing many pseudonyms for each IoT device can effectively protect vehicle privacy. The constraint is that if the mIoT device changes its MEC connection, it must use a new pseudonym for a new round of authentication right away (Abbas et al. 2017). This technique requires the blockchain to retain pseudonymous data in order to enable proper authentication. However, frequent data uploads may cause a blockchain throughput issue because of the high frequency of mIoT-MEC replacement (Li and Wu 2021). Therefore, the prime problem we try to solve is how to efficiently achieving the aggregation of supplementary authentication information utilizing pseudonyms for blockchain-based authentication strategy.

We offer a vector commitment technique in this paper to accomplish effective compression of certification evidence on the blockchain and reduce the blockchain's storage load. On this foundation, we build a fast search method that leverages binary trees to provide a quick search for vector commitment verification, which is a high-complexity task in auxiliary computing information. Finally, we devise a vector commitment-based authentication system that successfully realizes unlinkable mutual authentication between mIoT devices and MEC.

The contributions of this paper are as follows:

- We present a blockchain-based authentication system that compresses the data that the blockchain must hold using vector commitment technology.
- A binary tree-based auxiliary information indexing approach is proposed to lower the computational cost of vector commitment verification.
- To accomplish effective authentication of mIoT devices, we offer an authentication technique based on vector commitment. Also, to realize the unlinkability of mobile IoT when traveling between various MECs, we create a pseudonym method.

2. Related Work

The work in this paper is a continuation of the existing work (Tsang et al. 2021)(Viriyasitavat et al. 2022)(Lu 2021)(Xu et al. 2022). The above works all refer to the introduction of blockchain in existing scenarios. In (Tsang et al. 2021), the authors divide the research on the application of blockchain in IoT into 9 categories. Our research belongs to the category of "data privacy and security for blockchain-based IoT systems", which provides distributed device identity authentication. In (Viriyasitavat et al. 2022), the authors propose a blockchain-oriented structure of public key infrastructure to distribute device certificates to users and store them in the blockchain. However, this scheme needs to store the certificate of each device, which makes the storage burden of the blockchain large. The scheme in this paper supplements the blockchain storage compression strategy, which can be applied for authentication. In (Lu 2021), this research can be categorized into the security application of blockchain, and fill in the information gap in authentication. The research in this paper helps to fill the above regrets. We show the existing research in the following, and analyze the uniqueness of the proposed scheme by analyzing the advantages and disadvantages of each. This paper Xu et al. (2022) studies a separate storage and authorization method for data sharing between organizations to ensure privacy protection and improve the convenience and efficiency of data access. This paper and we both use the alliance blockchain and smart contract, both adopt the separation of data on and

Table 1. Functional comparison between BUA and previous works

Scheme	Blockchain based	Evidence Compression	Unlinkable	Identity Privacy	Revocation
(Zhang et al. 2009)	✗	✗	✓	✓	✗
(Lin et al. 2015)	✗	✗	✓	✗	✓
(Aman et al. 2018)	✗	✗	✗	✓	✗
(Zhang et al. 2020)	✗	✗	✓	✓	✓
(Hammi et al. 2018)	✓	✗	✗	✗	✗
(Wang et al. 2019)	✓	✗	✗	✓	✓
(Cui et al. 2020)	✓	✗	✓	✗	✗
(Wang et al. 2020)	✓	✗	✗	✓	✗
BUA	✓	✓	✓	✓	✓

off the chain, and both support dynamic mode. The difference is that they focus on authorization, and we focus on authentication. They store the URL of medical information on the chain, and we store the auxiliary information related to authentication. They support dynamic authorization, and we support dynamic authentication after pseudonym switching.

For the certification research, the evolution can be characterized as follows: from centralized certification to decentralized certification. The majority of early authentication research centered on public key infrastructure (PKI). Entrants and verifiers make up the basic mutual authentication system, and the procedure is divided into three parts. To validate the authenticity of the identification, the verifier first matches the entrant's identity information with the trusted root. The entrant then verifies the validity of the certificate issued by the verifier. Finally, the entrant and verifier collaborate to create a symmetric key for secure communication over the external channel. The function comparison between BUA and previous works can be shown in Table 1.

Zhang et al. increased the scalability of universal authentication techniques to address the authentication load of large-scale vehicle communication in 2009 (Zhang et al. 2009). They use the distributed computing concept and divide the computational load to each road sensor unit (RSU). Lin et al. proposed an anonymous authentication technique for applications to preserve entrants' identity privacy in 2015 (Lin et al. 2015). They also discovered that the opponent could take the data collector key and mimic its legal identity to capture sensor data. Aman et al. proposed a two-factor authentication approach in 2018 (Aman et al. 2018). It mostly relieves CA storage burden on trustworthy roots as the number of IoT devices grows. CA can utilize this as a factor and transfer it to the equipment maker for storage because the factory number and wireless signal characteristics of IoT units cannot be replicated. Zhang et al. proposed a batch authentication approach to alleviating the latency problem when a large number of cars connect to the network in 2020 (Zhang et al. 2020). The revocation credential in this technique is the hash chain seed of the created vehicle pseudonym. The length of the revocation list is solely determined by the number of cars that have been revoked. So far, authentication research has progressed at a steady pace. Still, there is a recurring flaw with these centralized authentications: they rely too heavily on CA's honesty, leading to the entire system failing if Byzantine problems arise in CA. As a result, the next development trend in authentication research will be decentralized authentication.

The key context for blockchain research is to enhance efficiency and scalability as much as feasible while maintaining consistent security. Hammi et al. proposed a blockchain-based IoT authentication solution in 2018 (Hammi et al. 2018). They encode the authentication process into smart contracts to accomplish fully distributed authentication. At the same time, they feel that one of the most important roles of an authentication system is to ensure message integrity, and they suggest a distributed key distribution technique. However, this system deployment strategy, which relies on intelligent contracts, has obvious throughput and waiting time flaws. Because of the significant wait period, the system will be in a waiting state most of the time. Wang et al. proposed a blockchain-based authentication strategy that included key management and increased anonymity in 2019 (Wang et al. 2019). They avoid the detrimental impact of the blockchain's inherent flaws on the authentication mechanism by using off-chain authentication. They utilize the blockchain to store the system settings and smart meter authentication results, which are simple and quick to cancel. Simultaneously, because the CA discloses the parameters, the system's reliance on it may be decreased even more. Cui et al. proposed an authentication approach based on an alliance chain, achieving a breakthrough in which the authentication process is independent of CA in 2020 (Cui et al. 2020). They provide authentication protection for various levels of access using the alliance chain's intrinsic authentication mechanism. It is compatible with external blockchain direct access and enhances the types of access. Wang et al. focused on blockchain-based authentication systems' batch authentication capacity and quick revocation process in the same year (Wang et al. 2020). The concept of quick consensus encourages nodes to enter and quit with little latency. The batch verification technique cuts the average authentication time in half.

The research on blockchain-based authentication techniques has progressed to this point. When dealing with large-scale device access, the primary issues are how to hide the trail when the accessor moves between various authentication nodes and how to lessen the storage load of blockchain systems. This study proposes a number of interesting solutions. We employ verified pseudonyms as authentication credentials and make the pseudonyms unlinkable to increase the unlinkability. Then, we use vector commitment technology to compress the authentication credentials on the blockchain to lessen the load on the blockchain.

3. System Model

The BUA's system model and description members are initially introduced in this section. The implementation process, threat model, and design goals are then described. The notations used in this paper are summarized in Table 2.

3.1. Design Goals

The performance and safety objectives that need to be considered for the designed BUA scheme will be introduced as follows:

- **Anonymity:** MIE should not reveal it's true identity throughout the certification process and should be referred to by a pseudonym.
- **Unlinkability:** MS cannot judge whether the pseudonym belongs to the same vehicle based on the pseudonym during the MIE authentication and cannot obtain the MIE trajectory information;

Table 2. Notations for BUA design.

Notation	Description
IoT	Internet of Things
MEC	multi-access edge computing networks
CA	Certificate authority
MS	multi-access edge computing server
MIE	Mobile IoT Equipment
$e(,)$	function of bilinear pairing
C	Commitment vector
h_i	Single pseudonym tag
$h_{i,j}$	Double pseudonym tag
PK_{MS}	Public key of MS
SK_{MS}	Private key of MS
ps_i^t	I-th pseudonym of MIE_t
ID_{MIE_t}	Identity of MIE_t
ADD_C	The address of the commitment vector C
ID_C	The identity of the commitment vector C
P_{Auth}	authentication proof for MIE_t
Λ_i	commitment evidence

- **Mutual authentication:** The process consists of two parts: MS verifies MIE's identity, and MIE verifies that the verification results received are correct;
- **Revocable:** When the MIE leaves the system, the pseudonym generated before the CA can no longer be used for authentication after the MIE is revoked.

3.2. System Model

In the existing blockchain-based authentication systems, *MS* is commonly used to upload authentication evidence to the blockchain. But they only assume that MS is "honest but curious" and ignore the fact that when MS is deployed in the virtual environment, it is easy to be attacked and disrupt the regular execution of the system (for example, tampering with evidence causes the authentication system to fail). Therefore, we input the certification evidence generated by CA into the smart contract and realize the evidence on the chain through the consensus between *MSs*. For the choice of blockchain, our recommendation is to deploy smart contracts based on the consortium blockchain such as Hyperledger Fabric. There are several reasons: 1) Compared with the public chain, the consensus algorithm adopted by the consortium blockchain has lower waiting delay. 2) The consortium blockchain itself has a user identity management mechanism, which can cooperate with the authentication scheme proposed in this paper to achieve on-chain authority management of MIEs. 3) The consortium blockchain has higher flexibility. The consensus strategy can be dynamically selected according to the number of online vehicles to balance the robustness of the system and consensus waiting time. When a new MIE enters the jurisdiction of the *MS*, the *MS* searches the blockchain based on the pseudonym information provided by the MIE to find the authentication evidence and conducts identity verification to determine the true identity of the MIE. Additionally, when the MIE is revoked, the *MS* can update the certification evidence in the blockchain according to the MIE's request. The system framework is shown in Figure 2.

The members of the system are introduced as follows:

- **Mobile IoT Equipment(MIE):** MIE (such as intelligent vehicles and mobile wearable equipment) have low storage capabilities, where the cached data are

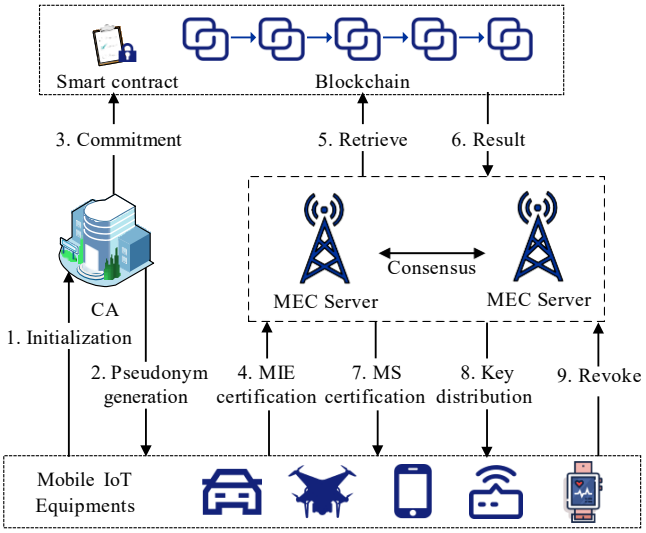

Figure 2. A workflow example for BUA

easily destroyed. Meanwhile, they have poor computing capabilities and can only perform simple data encryption, decryption, and signature operations.

- **MEC server (MS):** MS is located at the network edge, close to the MIE, and has caching and computing capabilities. That is, it provides services such as computing offloading for multiple MIEs. By deploying a blockchain between MSs, it is possible to achieve cross-operator MS collaboration.
- **certification authority (CA):** CA is a trusted core organization with strict security protection standards, can issue certificates, and confirm the correctness of its identity based on the identity information provided by MIE. At the same time, CA has a true random number generator, which can generate credible pseudonyms. It is worth noting that CA has limited computing power and capacity, and frequent communication with CA should be avoided.
- **Blockchain (BC):** An decentralized consortium blockchain platform maintained by network miners. There are some authentication nodes in the blockchain nodes to dynamically manage the identity of the participants. The blockchain system adopts the DPoS consensus algorithm to reduce the number of validators required to reach consensus and enables faster validation times. At the same time, this platform is determined by consensus to upload information on the chain, the bandwidth is limited, and it is longer than the general point-to-point transmission.

The execution process of BUA is as follows:

- **Step 1: Initialization.** First, the CA authenticates the MS and allocates a public-private key pair for each MS. Then, the MIE takes part of the private information (such as the last 6 bits of the MIE ID) and encrypts it with the CA public key and sends it to the CA. Finally, CA submits a query request to the MIE management agency through a secure channel. If the query is passed, the vehicle management office returns the hash value after the exclusive OR of the complete information, and it generates a public and private key pair for the user.
- **Step 2: Generate pseudonyms.** The CA calculates the n pseudonyms of MIE, sends them to the user, and maintains a list of pseudonyms.

- **Step 3: Generate commitment.** CA takes the first q values from the pseudonym list and calculates the commitment. Then, CA communicates with the BC authentication node and registers the pseudonym and commitment. Finally, CA stores them on the blockchain through MS consensus.
- **Step 4: MIE authentication (processes 4-6 in figure 2).** MIE calculates the proof of commitment and sends the pseudonym, the proof of commitment, and the address of the corresponding commitment on the chain to the MS. Then, the MS forwarded the authentication evidence to the blockchain authentication node for interaction to obtain the blockchain smart contract access rights. The MS finds the corresponding commitment on the blockchain according to the address and constructs a verification formula for verification. The MS verifies that the MIE is successful if the verification is passed.
- **Step 5: MS authentication.** The vehicle randomly generates a data encryption key and uses the MS public key to encrypt the key and timestamp before sending it to the MS. After MS decrypts, it encrypts the timestamp with the private key and sends it to MIE. The MIE constructs a verification formula. If it succeeds, the MIE authenticates the MS.
- **Step 6: Key distribution.** The MIE randomly generates a symmetric key, encrypted using the MS's public key and sent to the MS. The MS decrypts the key using the private key and verifies correctness against the timestamp. The MS then returns a confirmation message.
- **Step 7: MIE revoke.** When the MIE is revoked, the CA first queries the commitment where the MIE pseudonym is located, calculates the new commitment, and stores the new promise and the updated address on the blockchain together.

3.3. Threat Model

The security assumption of each member in this model: CA is considered "honest", which means that MIE's pseudonym and calculated commitment are correct. MS is considered "semi-honest", which means that MS can perform some actions to analyze MIE's identity information. There may also be a collaboration amongst MSs, which can be classified into two scenarios. On the one hand, MSs may use authenticated pseudonyms to predict the vehicle's path. On the other side, collaborating MSs strive to pass the certification as a vehicle by fabricating phony identities and evidence. This is because they can communicate to collect verified pseudonyms to gain all of the pseudonyms contained in the commitment. MIE is regarded as "honest," and will supply the proper pseudonym in order to participate in the certification process. The communication connection is also secure, according to the scenario's assumptions.

4. Efficient Scheme of Unlinkable Authentication

In this section, firstly, we introduce the preliminary, including bilinear mapping and vector commitment. Secondly, we describe the AEST index and the corresponding retrieval algorithm. Finally, we introduce the protocol of unlinkable authentication.

4.1. Preliminary

4.1.1. Bilinear Mapping

The bilinear mapping is defined as follows: Let G_1, G_2, and G_T be a cyclic group of order p, and p is a prime number. The generators of G_1 and G_2 are g_1 and g_2, respectively. The bilinear pair $e : G_1 \times G_2 \to G_T$ should satisfy the following three attributes (He et al. 2011):

- Bilinearity: For $\forall u \in G_1, \forall v \in G_2$, equation $e(u^a, v^b) = e(u, v)^{ab}$ holds;
- Computability: For $\forall u \in G_1, \forall v \in G_2$, there is an effective algorithm for calculating $e(u, v)$;
- Non-degeneration: For g_1 and g_2, $e(g_1, g_2) \neq 1$.

4.1.2. Vector Commitment

There are two members in the process of vector commitment: promiser and verifier (Catalano and Fiore 2013). The promiser wants to prove that he possessed some data blocks some time ago. The vector commitment scheme includes five polynomial-time algorithms: $VC.KeyGen$, $VC.Com$, $VC.Open$, $VC.Ver$ and $VC.Update$:

Key generation: $VC.KeyGen(1^k, q)$**.** The algorithm takes the safety parameter k and the promised vector size $q(q = poly(k))$ as inputs, and output some public parameters pp.

Commitment generation: $VC.Com_{pp}(m_1, \ldots, m_q)$**.** The promiser executes this algorithm to generate commitments. He takes the message sequence $m_1, \ldots, m_q \in M$ as inputs, where M is the message space. Then, the algorithm outputs a commitment string C and an auxiliary information aux.

Proof generation: $VC.Open_{pp}(m_i, i, aux)$**.** When the promiser try to prove to verifier that he owns data m_i, he executes this algorithm to generate auxiliary proof Λ_i.

Commitment verify: $VC.Ver_{pp}(C, m_i, i, \Lambda_i)$**.** The verifier executes this algorithm to discriminate whether m_i is belonged to C by using the auxiliary proof Λ_i. If it is verified, the algorithm will output TRUE.

Commitment update: $VC.Update_{pp}(C, m_i, m'_i, i)$**.** The promiser executes this algorithm when he want to update the data m_i to m'_i. Beyond this, the algorithm uses the previous commitment C and data index i as input. The output is updated C'.

4.2. Commitment Evidence Generation Algorithm

4.2.1. Main Idea

In vector commitment, the $\Lambda_i = \prod_{j=1}^{q} tag^{m_i}$ is commonly linked to pseudonyms, and continuous modular exponentiation operations are usually required. The computational complexity of indexing and replicating is $O(n)$. When faced with densely connected road sections, the number of compressed pseudonyms in commitments will be enlarged to ensure that there is no redundancy at CA, resulting in a dramatic increase in the number of j. At this point, the initial strategy's computational overhead deficit will influence the entire system, slowing down CA's response time.

There are indexing technologies with the complexity of $O(1)$ and $O(\log n)$ to choose from for retrieval tasks to achieve high-efficiency (Lv et al. 2021)(Luo et al. 2018) retrieval of Bloom filters. The most efficient index has a complexity of $O(1)$. However,

it can only determine whether the search element is present in the set and provide a Boolean answer if it is. Although a Bloomier filter can produce element values, it does not support dynamic operations. Therefore, new pseudonyms cannot be added to the index in real-time, resulting in a gap (Chazelle et al. 2004). As a result, a binary tree with a $O(\log n)$ complexity is optimum.

According to the definition of vector commitment, $\Lambda_i = \prod_{j=1,j}^{q} h_{i,j}{}^{m_i}$, where i is a fixed value that is specified by The pseudonym, and j is the number of compressed pseudonyms in the commitment. There is a crucial link between i and the building of the index binary tree. First, according to i, the corresponding set of $\{h_{i,j}\}$ should be retrieved; second, the accumulation process includes omitted i elements, which should be excluded from the index construction. We suggest a piece of auxiliary evidence spanning tree (AEST) based on this, as shown in Figure 3.

AEST has a two-layer structure, with the first layer recording h_i and the second layer storing $h_{i,j}$. The first layer is instantiated as an array to achieve $O(1)$-level indexing performance because the number of i is fixed when the vector commitment is defined. The accumulated multiplication value is the output result for the second layer. The multiplication operation is still $O(n)$ efficient when an array is employed. As a result, a binary tree structure is used, with the leaf nodes recording $h_{i,j}$ and the intermediate nodes recording the result of child node accumulation. AEST can retrieve $h_{i,j}$ using the breadth-first technique while obtaining Λ_i using intermediary nodes for the retrieval job $h_{i,j}$.

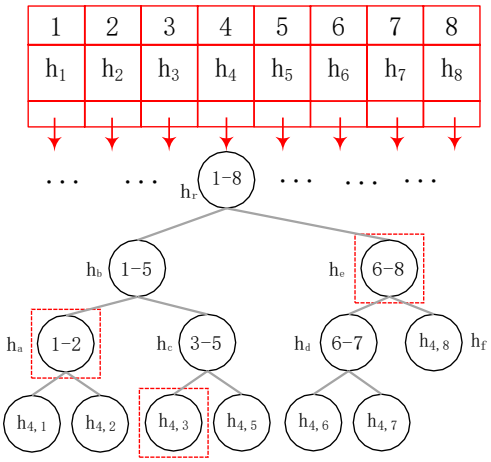

Figure 3. A simple example for AEST

4.2.2. AEST and Retrieval Algorithm

The following is a detail description of AEST: There are two layers, the first is an array, and the second is binary tree. The first layer of AEST's h_i array is made up of three parts: 1) obtain the value that corresponds to the value of i implemented the query function; 2) storage of h_i; 3) a pointer to $i-tree$'s root, the binary tree at the second level of AEST records the value of $h_{i,j}$ at the leaf nodes. The product of its two child nodes and the matching index range are recorded by the intermediate node. The root node stores all of j's value fields.

Figure 3 shows a simple example, with the retrieval command $\{h_{4,5}\}$. AEST extracts h_4 from the array and uses the pointer to locate the binary tree's root node. Then, using breadth-first traversal, modify the pointer point and use the right child node

as the output value multiplier; the left child node h_b contains the target value. The target value is contained in the right child node of the intermediate node h_b, the pointer pointing is modified again, and the left child node h_a is used as the multiplier of output value. Use $h_{4,3}$ as the multiplier of output value for the intermediate node h_c, and ultimately output $\Lambda_i = h_e \cdot h_a \ h_{4,3}$. The pseudo-code of the algorithm for retrieval and auxiliary evidence calculation is shown in the algorithm 1.

Algorithm 1 Auxiliary evidence generation algorithm

```
Input: TreeNode root, i
Output: Commitment evidence Λi
BEGIN
01. Queue(TreeNode) queue = new Linked List( ) ();
02. queue.add(AEST.root);
03. While(queue!=null)
04.   TreeNode Temp = queue.end;
05.   if(i in temp.index):
06.     while(queue!=null)
07.       TreeNode Temp2 = queue.end;
08.       Λi = Λi*Temp2.value;
09.   if(temp.left != null):
10.     queue.add(temp.left);
11.   if(temp.right != null):
12.     queue.add(temp.right);
13. return(Λi);
END
```

4.3. Unlinkable Authentication Protocol

4.3.1. Summary of Protocol

The protocol consists of seven steps. The execution algorithm of this protocol is defined as follows:

Initialization: $\{g, e, \{h_i\}, \{h_{i,j}\}, PK_{MS}, SK_{MS}\} \leftarrow Init_{CA}(1^k)$. This is a probabilistic algorithm executed by the *CA*. The input is security parameter 1^k. And the outputs are elliptic curve generator g, bilinear mapping function e, required parameter for commitment $\{h_i\}$ and $\{h_{i,j}\}$. In the end is *MS*'s key pairs $\{PK_{MS}, SK_{MS}\}$.

Pseudonym Generation: $ps_1^t, \ldots, ps_n^t \leftarrow PseuGen(ID_{MIE})_t$. This is a probabilistic algorithm executed by the *CA*. For a MIE, the *CA* generates n pseudonym for one time corresponding timestamp t.

Commitment Generation: $ComGen(ps^t_1, ..., ps^t_q)$. This is a deterministic algorithm executed by the *CA*. According to pseudonyms $\{ps_1^t, \ldots, ps_n^t\}$, the *CA* calculates commitment C, and update to the blockchain using smart contract. Finally, the outputs are the address Add_c in blockchain, commitment C, and identity ID_c.

MIE Authenticaion: $\{P_{Auth}\} \leftarrow MIEAuth(\Lambda_i, ps^t_i, Add_c, SK_{MS})$. This is a deterministic algorithm executed by the *MS*. It retrieves commitment C from blockchain using Add_c, and input the auxiliary information Λ_i to verify whether the pseudonym ps_i^t is true. Then, the *MS* generates authentication proof P_{Auth} for MIE_t.

MS Authentication: $\{TRUE, FALSE\} \leftarrow MSAuth(P_{Auth})$. This is a deterministic algorithm executed by the MIE_t. It verifies the correctness of MIE_t, and outputs TRUE or FALSE.

Key Distribution: $\{TRUE, FALSE\} \leftarrow Keydist(SK_{MS}, KeyInfo, time')$. This is a deterministic algorithm executed by the *MS*. It verifies key information *KeyInfo* by using $time'$. And then it decrypts the *KeyInfo* using SK_{MS}. Finally, it outputs TRUE or FALSE.

MIE Revocation: $\{C', ID_{C'}\} \leftarrow MIERevo(ID_{MIE_t}, Add_c)$. This is a deterministic algorithm executed by the MS. According to the Add_c, the MS retrieves the commitment C and updates to C' using ID_{MIE_t}. Finally, the MS insert the new commitment to the blockchain known as $ID_{C'}$.

4.3.2. Protocol Details

The specific implementation process of the agreement is as follows:

- $Init_{CA}(1^k)$: The CA sets G, G_T as the bilinear group of order q, and the bilinear mapping is defined: $e : G \times G \longrightarrow G_T$. Take $g \in G$ as the generator. Randomly take d numbers $m_1, \ldots, m_d \in Z_p$ as public parameters.
- $PseuGen(ID_{MIE_t})$: For the authentication request of MIE MIE_t, CA takes random numbers r_1 and r_2, and the result of performing hashing on r_1 is used as a pseudonym $ps_1^t = h(r_1)$ for MIE_t and The n-th pseudonym is $ps_n^t = (h(ps_{n-1}^t) \oplus r_2)$. Insert the pseudonym list in string order, and assign public and private key pairs $\{pk_{MIE}, sk_{MIE}\}$ to MIE_t, and return $\{ps_1^t, \ldots, ps_n^t, \{pk_{MIE_t}, sk_{MIE_t}\}\}$ to the MIE_t through the secure channel.
- $ComGen(ps_1^t, \ldots, ps_q^t)$: CA takes the first q values from the pseudonym list, calculates $h_i = g^{ps_i^t}$ for each pseudonym, calculates $h_{i,j} = g^{ps_i^t ps_j^t}$ for all $i \neq j$, and calculates the commitment $C = h_1^{m_1} \cdot h_2^{m_2} \cdot \ldots \cdot h_q^{m_q}$. Put $\{ID_C, C\}$ on the blockchain and send $\{h_i, \{h_{ij}\}_{i \neq j}, Add_C\}$ to the MIE's ps_i^t.
- $MIEAuth(\Lambda_i, ps^t, Add_c, SK_{MS})$: The MIE calculates the commitment evidence $\Lambda_i = \prod_{j=1, j \neq i}^{q} h_{i,j}^{m_j}$, and sign the ID_{MS} using its secret key sk_{MIE_t} to generate $ID_{MS}^{sk_{MIE_t}}$, and sends $\{ps^t, \Lambda_i, Add_C, ID_{MS}^{sk_{MIE_t}}\}$ to MS. The MS firstly verify the signature $ID_{MS}^{sk_{MIE_t}}$ by using pk_{MIE_t}. If correct, than finds the commitment C according to ADD_C, and finds the latest commitment C' backwards according to ID_C. If there is no update, then $C' = C$. Then MS construct the verification formula: $e(C' / (g^{ps_i^t})^{m_i}, h_i) =^? e(\Lambda_i, g)$, if the formula is established, the MS verifies the MIE successfully.
- $MSAuth(P_{Auth})$: MS sends $\{(C'^{h(time_{||}ps_i^t)})^{\alpha}, time'\}$ to the MIE, where α is the private key of MS. The vehicle structure verification formula $e((C'^{h(time_{||}ps_i^t)})^{\alpha}, g) =^? e(C'^{h(time_{||}ps_i^t)}, g^{\alpha})$, if the verification is passed, the MIE verifies MS successfully.
- $Keydist(SK_{MS}, KeyInfo, time')$: At this point, the authentication process is complete, and the key distribution steps will be performed. The MIE randomly generates a session key EN_k, and uses the MS's public key to encrypt $EN_{PK_{MS}}(EN_k || time')$ and sends it to the MS. MS retains the session key and uses the private key to encrypt $EN_{SK_{MS}}(time')$ and sends it to the MIE. If $DE_{PK_{MS}}(EN_{PK_{MS}}(time')) = time'$, it means that the MS has passed the verification of the MIE.
- $MIERevo(ID_{MIE_t}, Add_c)$: When the MIE is revoked, CA first queries the commitment corresponding to the MIE's pseudonym, calculates $C' = C \cdot h_i^{-m_i}$ respectively, and uploads the new commitment and the updated $\{C', ID_{C'}\}$ to the blockchain.

The queue mentioned in the protocol is a clever idea for pseudonym caching. The CA, faced with real-time MIE initialization requests, has to conduct concurrent processes to reduce The MIE wait time. The problem with this is that multiple MIE pseudonyms can be generated simultaneously. Multiple MIE pseudonyms should be integrated into the same commitment to achieving unlinkability. At this time, the queue is used to add the kana without order, and the pseudonyms cache can be effec-

tively managed after the commitment calculation.

Correctness The correctness of this protocol can be proved by the authentication equation in $MIEAuth(\Lambda_i, ps^t, Add_c, SK_{MS})$. The derivation of the formula is shown as follows:

$$
\begin{aligned}
& e(\Lambda_i, g) \\
=& e(\prod_{j=1, j=_i} h_{i,j}^{m_j}, g) \\
=& e((\prod_{j=1} h_{i,j}^{m_j}) / h_{i,i}^{m_i}, g) \\
=& e((\prod_{j=1} h_j^{m_j}) / h_i^{m_i}, g^{Ps_i^t}) \\
=& e(C' / (g^{ps_i^t})^{m_i}, h_i)
\end{aligned}
$$

The verification formula can be derived correctly, indicating the correctness of this protocol.

5. Secure Analysis

In this section, we first establish two collusion-oriented security models and then complete the reduction based on CDH difficult problems to prove the security of BUA.

5.1. Security Model

We construct a link attack model and a forgery attack model for the two collusion attacks that MS may implement.

Linking attack model. This model simulates the situation that a compromised MS_1 try to analysis where the MIE from assisted with a collusion MS_2, which means MS_1 could distinguish the last pseudonym according to the present. So we simply the attack: if the adversary can distinguish whether any two pseudonyms are generated from one seeds, it can win the game. Let MS_1 and MS_2 are adversaries, and CA is the challenger. The following is the specific interaction procedure between the challenger and the adversaries:

- **Setup:** Initialization and pseudonym generation algorithm are performed by the challenger. The challenger chooses a random number $v \in Z_p$, and generates set of hash chain seeds $\{r_{i,1}, r_{i,2}\}_{i\in\{1,v\}}$ at random and make them private.
- **Query:** The adversaries send a random value $u(u < v)$ to the challenger. The challenger creates u number of pseudonyms by using random seeds $\{r_{i,1}, r_{i,2}\}_{i\in\{1,v\}}$ where only two elements are generated by a same seed.
- **Output:** Following receipt of the set of pseudonyms, the adversary evaluates the correlation between them and chooses two pseudonyms to provide to the challenger. If these two pseudonyms are generated by the identical pair of hash chain seeds, the challenger determines that the attack is successful.

Forgery attack model. The compromised *MS* can collect a set of authenticated pseudonyms from collusive *MSs*, and then to frogery a new pseudonym, which could let a honest *MS* believe in it is a MIE. We define the adversary as a collusion MS collection. The challenger is the MIE and the ”virtuous” MS, which means that this MS did not participate in the collusion, and the authentication process is performed commonly. If the ”virtuous” MS is authenticated, the attack is successful. Below we define the interaction process of the security model:

- **Setup:** The challenger performs initialization and pseudonym generation algorithms. The challenger generates a set of commitments and corresponding sets of pseudonyms, all of which are made public.
- **Query:** The adversary asks the challenger for the pseudonyms in the set, and the challenger feeds back supporting evidence that can be verified.
- **Output:** The adversary forges pseudonym and auxiliary proof, then sends them to the challenger together. If the challenger passes the verification, the attack is successful.

5.2. Security Analysis

We first introduce the difficult assumption and then establish the security reduction of BUA.

Definition 1: (**Square-CDH Problem**) The Computational Diffie-Hellman problem is that, given $g, g^{\alpha} \in G$ for unknown $\alpha \in Z_p$, to compute g^{α^2}(Rabaninejad et al. 2019).

Theorem 1: (Anti-replay attack) If the signature generation algorithm is secure, then our scheme is anti-replay.

Proof: Assume that the success rate of the encryption algorithm being cracked is ϵ. If the attack is successful, the adversary needs to replay $\{ps_i^t, \Lambda_i, Add_3\}$ and forge $ID_{MS}{}^{sk_{MIE_t}*}$. The challenger verifies the discriminant as follow:

$$Decrypt_{pk_{MIE_t}}(ID_{MS}{}^{sk_{MIE_t}*}) =^{?} ID_{MS}$$

If the discriminant are equal, the attack is successful. Then the probability of a successful attack at this time is $P = \epsilon$. Therefore, when the probability of the encryption algorithm being cracked is negligible, this solution can resist the anti-replay attack.

Theorem 2: (Anti-linking attack) If the hash function is one-way, then our scheme is anti-linking.

Proof: Assume that the probability of one-way penetration of the hash function used in this scheme is ω. The challenger chooses a random number $v \in Z_p$, and generates v sets of hash chain seeds $\{r_{i,1}, r_{i,2}\}_{i\in(1,v)}$, calculate two pseudonyms for each hash chain $ps_1^t = h(r_{i,1})$, $ps_n^t = (h(ps_{n-1}^t) \oplus r_{i,2})$. The challenger randomly selects two pseudonyms from the 2v pseudonyms and sends them to the adversary each time. The attack is established if the adversary can distinguish whether the two pseudonyms belong to the same hash chain with a non-negligible advantage. Only when the adversary can infer another pseudonym on the same hash chain based on a pseudonym, the adversary has an advantage, so the advantage is as follow:

$$ad = Decrypt_{Hash}(ps_n^t) + P(getting(r_{i,2}))$$

Where P () represents the probability function. Since the acquisition of $r_{i,2}$ is directly related to the cracking of the hash function, $ad = 2\omega$. Since ω can be ignored, the probability of a successful attack can be ignored.

Theorem 3: (Anti-forgery attack) If the CDH assumption holds, our scheme is anti-forgery.

Proof: The challenger takes as input a tuple (g, g^{α}) and its goal is to compute g^{α^2}.

First, the challenger selects a random $r \in Z_p$ as a guess for the index i on which Adversary will break the position binding. Next, the challenger chooses $z_j \in Z_p, \forall j \in [q], j \neq i$, and it computes: $h_i = g^{\alpha ps_i^t}$. The Challenger sets as follows:

$$pp = (g, \{h_i\}_{i\in[q]}, \{h_{i,j}\}_{i,j\in[q],i\ j})$$

Notice that the public parameters are perfectly distributed as the real ones. The adversary is supposed to output a tuple $(C, h_i^{ps^t}, h_i^{ps^{t*}}, j, \omega_j, \omega_j^*)$.

If $i = j$, then the Challenger aborts the simulation. Otherwise it computes as follow:

$$g^{\alpha^2} = (\omega_i / \omega_i^*)^{(Ps_i^t - Ps_i^{t*})^{-1}}$$

To see that the output is correct, observe that since the two openings verify correctly, then it holds as follow:

$$e(g, g)^{(Ps_i^t - Ps_i^{t*})\alpha^2} = e(\omega_i^*/\omega_i, g)$$

Since $h_i = g^{\alpha Ps_i^t}$, one can easily sees that this justifies the correctness of Challenge's output. Notice that if Adversary succeeds with probability ϵ, then the Challenger has probability ϵ/q of breaking the Square-CDH assumption.

6. Performance Analysis

In this section, we first analyze BUA using theorem comparison, including overhead comparison, communication overhead comparison and storage overhead comparison on the blockchain. Then, simulation experiments were used to evaluate the actual performance of BUA.

6.1. Theory Analysis

To adapt to the massive MIE dynamic connections in the mobile Internet of Things, we need to lower computing overhead and communication load. As for the calculation overhead, we will analyze from four aspects: First, the authentication scheme is initialized. For each MIE, the long waiting time delay will delay the service data transmission. The second is the certificate generation stage. The authentication protocol that supports batch processing will pre-package the authentication evidence, which may cause calculation delays. The third is the authentication phase. The authentication process based on cryptographic protocols may cause high time overhead, and frequent communications will bring unacceptable waiting delays. Finally, it is the revocation stage. If the revocation of a single MIE affects the global authentication parameters, it will make the system inefficient. We compare the proposed scheme in this article

with the BAA (Wang et al. 2019) and SEMA (Wang et al. 2020) schemes because BAA and SEMA are both blockchain-based unlinkable authentication schemes with similar functions and security goals. The theoretical comparison of the specific process is shown in the table 3.

Since our scheme needs to calculate the commitment and verify the auxiliary information in the initialization phase, we need $m(n-1)$ as the coefficient. Only in it, n represents the number of aggregate pseudonyms in the vector commitment and is generally a fixed value. Therefore, the computational complexity of $m(n-1)$ is still $O(m)$. In addition, the computational cost of M_{Z_p} is significantly lower than that of M_G and E_G, which is enough to offset $(n-1)$. Comprehensive analysis shows that the proposed scheme still has efficiency advantages in the initialization phase. The evidence generation stage is a process specially considered for BUA because additional verification evidence is required when using vector commitment verification. In fundamental analysis, it can be included in the initialization overhead. Since we use the binary tree technology to calculate the cumulative multiplication result, the computational complexity is $O(\log n)$. Compared with other schemes, the mE_G calculation overhead is so small that it can be ignored. In the two-way authentication phase, the number of operations that our solution needs to calculate is less than that of other solutions, and the system's deployment can reduce the computing waiting time. There is only a bilinear pairing operation. As an auxiliary operation, the time cost used is significantly longer than M_G and M_{Z_p}. Fortunately, the overall number of calculations is small, which can offset part of the calculation pressure.

Table 3. Computational cost comparison

Schemes	initialization	Proof generation	Mutual authentication	revocation
BAA	$2E_G + 5mE_G + mM_G + 4mH_{Z_p} + mM_{Z_p}$	$NULL$	$12E_G + 3M_G + M_{Z_p} + 9H_{Z_p}$	$NULL$
SEMA	$3E_G + 2M_G + 3mE_G + mM_G + 2mH_{Z_p} + mM_{Z_p}$	$NULL$	$7E_G + 4M_G + M_{Z_p} + 4H_{Z_p}$	$NULL$
BUA	$m(n-1)M_{Z_p} + 2mE_G$	$log_2 m(n-1) + \lceil log_2 n - 1 \rceil M_G$	$3H_{Z_p} + 5E_G + 4P_G$	$M_G + E_G$

Next, we compared the communication overhead of the three schemes. In table 4, BUA has apparent advantages. It is most evident during the initialization phase. A mIoT device must generate multiple pseudonyms to achieve unlinkability. In the BAA and SEMA schemes, to resist forgery attacks, the cryptographic system must be initialized with the re-generation of pseudonyms, causing communication overhead to multiply as a function of m increases. The BUA scheme introduces hash chain technology to achieve effective separation between different pseudonyms. The disadvantages of BAA and SEMA in the authentication and revocation stage are that they store most of the registration information on the blockchain. As a result, during authentication, after the mIoT device sends the registration information to the MS, the MS also forwards the information to the blockchain smart contract. Execute the comparison that BUA only needs to retrieve the commitment from the blockchain and verify its correctness.

The last part of the theoretical analysis is the comparison of storage overhead on the blockchain. In the table 5, we can find that the BUA has a significant advantage in terms of efficiency. The main reason is that vector commitments have the nature of data compression. When each commitment carries m pseudonyms, only one G can realize batch authentication of multiple pseudonyms. The BAA and SEMA solutions

Table 4. Communication cost comparison

Schemes	initialization	Authentication	revocation
BAA	$m(4\|G\| + 8\|Z_p\|)$	$2\|G\| + 5\|Z_p\|$	$2\|G\| + 3\|Z_p\|$
SEMA	$m(5\|G\| + 4\|Z_p\|)$	$4\|G\| + 3\|Z_p\|$	$\|G\| + \|Z_p\|$
BUA	$\|G\| + (m + 3)\|Z_p\|$	$3\|G\| + 3\|Z_p\|$	$\|G\| + \|Z_p\|$

must instantiate a smart contract for each pseudonym, which will cause a storage burden on the blockchain.

Table 5. Storage load on the blockchain

Schemes	Storage on the blockchain
BAA	$4m(\|Z_p\|)$
SEMA	$2m(\|G\| + \|Z_p\|)$
BUA	$\|G\|$

6.2. Experimental Analysis

We constructed a simulation environment for protocol execution and compared it to SEMA and BAA to further analyze BUA's computational cost. This section introduces the simulated experimental environment as well as the results of the experiments. The simulation experiment was run on an Ubuntu virtual system with a 3.6GHz Intel i7-4790HQ CPU and 12GB RAM, and the virtual machine was given 4GB RAM and two processing cores. The authentication function based on bilinear pairing is implemented in the experiment using a password library (PBC). The MNTd159 curve is used to create a bilinear pairing with an 80bit safety parameter. The MS's communication range is 300 meters, while the MIE's typical travel in the city is 10 kilometers (Anbalagan et al. 2021)(Commission). The number of pseudonyms generated during setup is limited to between 0 and 50. Assume that the activities linking $||$ and XoR $\oplus$ are promised to be low-cost operations. Because the computing cost of these procedures is far smaller than that of other cryptographic operations, $||$ and $\oplus$ can be ignored throughout the process. In this subsection, we analyze the cost of the three steps of the protocol, namely the initialization step, the two-way authentication step, and the evidence generation step. The selection of the number of pseudonyms is 5-50, with five as the incremental value. We use Jing et al.'s proposal BAA (Wang et al. 2019), Weizheng et al.'s proposal SEMA (Wang et al. 2020) as a comparison, which is consistent with the comparison scheme used in the theoretical analysis to verify the theoretical analysis in the actual environment Accuracy. And at the end, comparing the total time cost between different schemes, there is not only the calculation cost of each step but also the waiting time of the device when switching between various operations, which is assumed to be 3*ms*.

The calculation overhead of initialization under different numbers of pseudonyms is compared in figure 3. In the figure, we can see that the BAA scheme is time-consuming

and volatile. In addition, the time cost of the BAA program has increased faster, which has been near twice as large as the program SEMA when the x-axis reaches 50. For the schemes SEMA and BUA, we can see that their growth is stable, and the broken line is steady. An interesting detail is that the growth rate of the BUA scheme is slightly lower than that of the SEMA, which can be noticed by comparing the area of the SEMA divided by the BUA scheme.

The figure 4 compares the computational overhead of the two-way authentication phase of different schemes. BAA is still the most time-consuming program, and its growth rate is on the rise, which is worrying. BAA is also unstable. The coordinate of 30 on the x-axis is significantly lower than the position we expected. For the SEMA and BUA schemes, we found that the BUA scheme is slightly higher than SEMA because of the high cost of the pairing operation. Both SEMA and BUA performed well in terms of growth rate and stability.

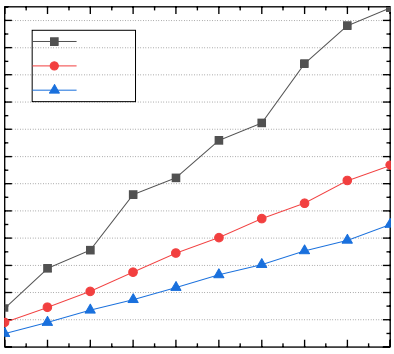

Figure 4. Comparison of the time cost of initialization

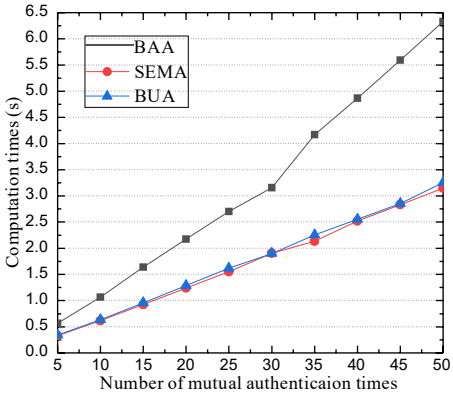

Figure 5. Comparison of the time cost of mutual authenticaion

The figure 5 is a notable comparison. We use this picture to illustrate the advantages of our proposed index AEST. The figure shows the comparison between the AEST-based commitment evidence generation algorithm and the general situation. Although we know through analysis that the retrieval complexity of *AEST* is $O(\log n)$, because, in BUA, the number of pseudonyms in each commitment is fixed, so it cannot reflect the number of pseudonyms in the contract The growing trend of $O(\log n)$. However, in the figure, the advantages of the calculation method used in the BUA scheme are still obvious. It maintains a meager growth rate while the calculation overhead is low when the amount of tasks is small. As the amount of tasks increases, it will soon be compared with the general proposal that opened the gap.

The last figure 6 can see the advantages of the BUA scheme compared to other schemes in terms of computational overhead. First of all, for BAA, we can see that its growth rate is steadily increasing, and its initial growth rate has already opened a gap with SEMA and BUA. As for SEMA, although it has excellent stability and high-efficiency characteristics at low tasks, there is still a gap between its growth rate and BUA, resulting in an increasingly obvious gap between time expenditure and BUA. In summary, based on the experimental data, we conclude that the BUA scheme's time cost is reduced by 13.24% compared to SEMA.

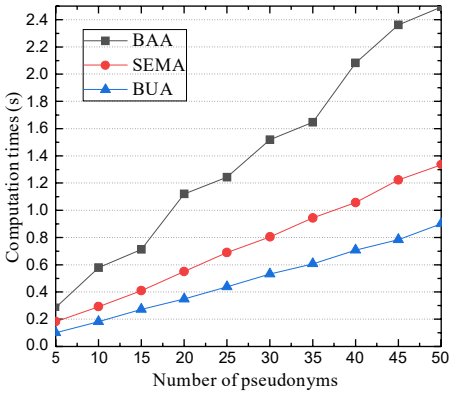

Figure 6. Comparison of the time cost of index

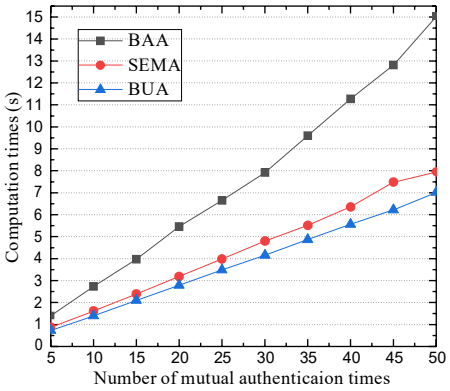

Figure 7. Comparison of the time cost of entire process

7. Discussion

We first discuss the applicability of BUA to the introduction of new technologies such as blockchain shardsLi et al. (2022) and DAGWang et al. (2022) to address the scalability bottlenecks faced by blockchain systems. Then, we discuss how our work is followed the trend current.

BUA can realize parallel authentication of different types of vehicles with the help of sharding technology. According to the information displayed on the license plate, we divided the vehicles into local and foreign vehicles and then subdivided the above two types by fuel type, including fuel cars and hybrid cars. (In China, the first Chinese character on a license plate indicates the provincial abbreviation, and the first letter represents the prefecture level city. Blue plates indicate gas-powered vehicles and green plates indicate electric or hybrid vehicles.) By deploying different vehicle pseudonyms in different network shards (for example, deploying ”Local-fuel vehicle” in Shard A, ”Local-electric/hybrid vehicle” in Shard B, ”foreign-fuel vehicle” in Shard C, and ”Foreign-electric/hybrid vehicle” in Shard D), parallel authentication of different vehicle classes can be achieved. In addition, since each vehicle belongs to only one category and different shards are independent of each other, there is no need to design a cross-shard communication mechanism and demonstrate the atomicity of cross-shard transactions.

When the blockchain adopts the DAG architecture, the retrieval speed of BUA can be improved. The blockchain of DAG architecture will have multiple chains existing at the same time, and each block on the chain may reference blocks on other chains, and the parent-child relationship between blocks is no longer single. Blockchain-based authentication schemes can achieve the purpose of authentication by using data structures such as Merkle trees, which store the promised hash values and pseudonyms in order of string size in different blocks of the DAG, and then query and verify them in smart contracts. Specifically, assuming we have a set of pseudonyms and corresponding promises, this information can be stored in a dag-based blockchain in such a way that each block stores the root hash of a Merkle tree. The leaf nodes of the Merkle tree store the hash of the pseudonyms and promises, and the parent block of each block points to the previous block. At validation time, the hash value of the corresponding block is queried through the smart contract, and the required commitments are retrieved in order from the root node to the leaf node. A well-designed retrieval procedure reduces

the cost of a single chain sequential retrieval from $O(n)$ to $O(logn)$. The hash value is then verified to match the hash value of the pseudonym and the promise to determine the consistency of the pseudonym and the promise.

The future trend of current blockchain technology development is as follows: 1) Reduce costs; 2) Increase confidentiality. 3) Improve customer loyalty; 4) Correct business strategic plan Zheng and Lu (2022). We are in line with these future trends.

(1) This paper uses blockchain technology to reduce maintenance costs. Because smart contracts supported by blockchain cannot be tampered with once they are released, there is no need to worry about the subsequent maintenance costs after the relevant standards are implemented and formulated. In addition, the stored evidence does not need to be centrally stored, which can reduce the storage cost for CA.
(2) The anonymous mechanism of the blockchain and the pseudonym mechanism cited in this article can well protect the privacy information of customers and increase the privacy protection capability of the system.
(3) If centralized authentication management is implemented, once the CA is attacked, the service will be paralyzed, reducing the customer's confidence in the business and causing customer loss. When the number of users is huge, the decentralized system has stronger stability.
(4) Develop the correct business strategy. On the question of which blockchain to choose, we choose the alliance blockchain. The low latency consensus mechanism in the alliance blockchain helps to authenticate the real-time requirements. The choice of alliance blockchain is also more flexible. The consensus strategy can be dynamically selected according to the number of online vehicles to balance the robustness of the system and the consensus waiting time.

8. Conclusion

The mobility of nodes in the mobile Internet of Things must remain unlinkable. The method that employs the blockchain as an anonymous technique has an issue with the chain's storage capacity. In this paper, we present a vector commitment-based identity authentication technique that decreases the storage overhead on the blockchain while lowering the dependency on CA. We propose a binary tree-based auxiliary evidence spanning-tree index and the accompanying evidence calculation technique to improve the generating efficiency of auxiliary evidence. Our method can withstand MS collusion assaults, according to security analysis. We show that the suggested proposal has a lower time cost than similar schemes through performance analyses. The limitations of this work are obvious: the blockchain is only proposed in the model as a concept, and we does not consider the technical choices during actual deployments, such as the choice of consensus protocols.

For future works, there are three aspects:

Fundamental research: we plan to design dynamic commitments to compress more pseudonyms in the future. Improve blockchain efficiency by reducing the number of times on the chain. In addition, we consider redesigning the authentication protocol and adding the function of re-authenticaion, so that MIE can authenticate with other MSs more quickly after the initial authentication. A MIE has authenticated by a MS, when it transfers to another MS, it should execute re-authentication process.

Application: the BUA has divergent application scenarios, such as cross-domain

authentication of smart medical care. When the user transfers from the local hospital to other hospitals, the authentication technology in this paper can be used. BUA can also be deployed in mobile Internet scenarios. This authentication method can be used when smartphones are transferred between base stations. This type of authentication requires higher latency. A more efficient consensus mechanism should be designed. If the blockchain is used as an open and trusted database.

System development: we plan to develop a BUA authentication system to construct a basic blockchain system using Hyperledger Fabric, where we set the CA as Endorser, and the MIE as Committer. Then, we develop the client for MIE. In the early stage, we only targeted mobile users, such as the Android platform. There are two main difficulties: the MIE side needs to store the pseudonym issued by the CA for a long time. If the user clears the cache, the pseudonym on the blockchain will be unavailable, wasting public storage space. On the other hand, CA, as an independent entity, may face a large number of MIE initialization requests in a short period of time when the system is first deployed. If an error causes the program to crash, it will destroy the system's availability.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos.62072092 and 62072093); the Key Research and Development Project of Hebei Province (No.20310702D); the Natural Science Foundation of Hebei Province (No.F2020501013) and VC Research (VCR 00000172). 2023 Hebei Provincial doctoral candidate Innovation Ability training funding project (No.CXZZBS2023168).

References

Tahmina Zebin, Patricia J Scully, Niels Peek, Alexander J Casson, and Krikor B Ozanyan. Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition. *IEEE Access*, 7:133509–133520, 2019.

Yang Lu and Li Da Xu. Internet of things (iot) cybersecurity research: A review of current research topics. *IEEE Internet of Things Journal*, 6(2):2103–2115, 2018.

Diego Mendez Mena, Ioannis Papapanagiotou, and Baijian Yang. Internet of things: Survey on security. *Information Security Journal: A Global Perspective*, 27(3):162–182, 2018.

Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, 20(4):2961–2991, 2018.

Prospective industry research institute. Analysis of the market status and development prospects of china's smart car industry in 2021. https://bg.qianzhan.com/trends/detail/506/210 902-168ca088.html.

Minhaj Ahmad Khan and Khaled Salah. Iot security: Review, blockchain solutions, and open challenges. *Future generation computer systems*, 82:395–411, 2018.

Mahshid Mehrabi, Dongho You, Vincent Latzko, Hani Salah, Martin Reisslein, and Frank HP Fitzek. Device-enhanced mec: Multi-access edge computing (mec) aided by end device computation and caching: A survey. *IEEE Access*, 7:166079–166108, 2019.

Xiang Sun and Nirwan Ansari. Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 54(12):22–29, 2016.

Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. A survey on iot security: application areas, security threats, and solution architectures. *IEEE Access*, 7:82721–82743, 2019.

Yiwen Zhang, Jie Pan, Lianyong Qi, and Qiang He. Privacy-preserving quality prediction for edge-based iot services. *Future Generation Computer Systems*, 114:336–348, 2021.

Roberto Morabito, Vittorio Cozzolino, Aaron Yi Ding, Nicklas Beijar, and Jorg Ott. Consolidate iot edge computing with lightweight virtualization. *Ieee network*, 32(1):102–111, 2018.

Xiaoying Jia, Debiao He, Neeraj Kumar, and Kim-Kwang Raymond Choo. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. *IEEE Systems Journal*, 14(1):560–571, 2019.

Yuting Li, Qingfeng Cheng, Ximeng Liu, and Xinghua Li. A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing. *IEEE Systems Journal*, 15(1):935–946, 2020.

Leanne Rylands, Jennifer Seberry, Xun Yi, Andrei Kelarev, Joe Ryan, and Yuqing Lin. Collusion-resistant protocols for private processing of aggregated queries in distributed databases. *Distributed and Parallel Databases*, 39(1):97–127, 2021.

Mohamed Tahar Hammi, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni. Bubbles of trust: A decentralized blockchain-based authentication system for iot. *Computers & Security*, 78:126–142, 2018.

Hui Yang, Bowen Bao, Chao Li, Qiuyan Yao, Ao Yu, Jie Zhang, and Yuefeng Ji. Blockchain-enabled tripartite anonymous identification trusted service provisioning in industrial iot. *IEEE Internet of Things Journal*, 2021.

Chao Lin, Debiao He, Neeraj Kumar, Xinyi Huang, Pandi Vijayakumar, and Kim-Kwang Raymond Choo. Homechain: A blockchain-based secure mutual authentication system for smart homes. *IEEE Internet of Things Journal*, 7(2):818–829, 2019.

Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2017.

Bengang Li and Faguo Wu. Post quantum blockchain with segregation witness. In *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pages 522–527. IEEE, 2021.

YP Tsang, CH Wu, WH Ip, and Wen-Lung Shiau. Exploring the intellectual cores of the blockchain–internet of things (biot). *Journal of Enterprise Information Management*, 2021.

Wattana Viriyasitavat, Li Da Xu, Assadaporn Sapsomboon, Gaurav Dhiman, and Danupol Hoonsopon. Building trust of blockchain-based internet-of-thing services using public key infrastructure. *Enterprise Information Systems*, pages 1–24, 2022.

Yang Lu. Implementing blockchain in information systems: a review. *Enterprise Information Systems*, pages 1–32, 2021.

Boyi Xu, Li Da Xu, Yuxiao Wang, and Hongming Cai. A distributed dynamic authorisation method for internet+ medical & healthcare data access based on consortium blockchain. *Enterprise Information Systems*, 16(12):1922757, 2022.

Lei Zhang, Qianhong Wu, Agusti Solanas, and Josep Domingo-Ferrer. A scalable robust authentication protocol for secure vehicular communications. *IEEE Transactions on vehicular Technology*, 59(4):1606–1617, 2009.

Xi-Jun Lin, Lin Sun, and Haipeng Qu. Insecurity of an anonymous authentication for privacy-preserving iot target-driven applications. *computers & security*, 48:142–149, 2015.

Muhammad Naveed Aman, Mohamed Haroon Basheer, and Biplab Sikdar. Two-factor authentication for iot with location information. *IEEE Internet of Things Journal*, 6(2):3335–3351, 2018.

Jing Zhang, Hong Zhong, Jie Cui, Yan Xu, and Lu Liu. An extensible and effective anonymous batch authentication scheme for smart vehicular networks. *IEEE Internet of Things Journal*, 7(4):3462–3473, 2020.

Jing Wang, Libing Wu, Kim-Kwang Raymond Choo, and Debiao He. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. *IEEE Transactions on Industrial Informatics*, 16(3):1984–1992, 2019.

Zhihua Cui, XUE Fei, Shiqiang Zhang, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. A hybrid blockchain-based identity authentication scheme for multi-wsn. *IEEE*

Transactions on Services Computing, 13(2):241–251, 2020.

Weizheng Wang, Huakun Huang, Lejun Zhang, and Chunhua Su. Secure and efficient mutual authentication protocol for smart grid under blockchain. *Peer-to-Peer Networking and Applications*, pages 1–13, 2020.

Daojing He, Chun Chen, Sammy Chan, and Jiajun Bu. Secure and efficient handover authentication based on bilinear pairing functions. *IEEE Transactions on Wireless Communications*, 11(1):48–53, 2011.

Dario Catalano and Dario Fiore. Vector commitments and their applications. In *International Workshop on Public Key Cryptography*, pages 55–72. Springer, 2013.

Xianwei Lv, Zhenfeng Shao, Xiao Huang, Wen Zhou, Dongping Ming, Jiaming Wang, and Chengzhuo Tong. Bts: a binary tree sampling strategy for object identification based on deep learning. *International Journal of Geographical Information Science*, pages 1–27, 2021.

Lailong Luo, Deke Guo, Richard TB Ma, Ori Rottenstreich, and Xueshan Luo. Optimizing bloom filter: Challenges, solutions, and comparisons. *IEEE Communications Surveys & Tutorials*, 21(2):1912–1949, 2018.

Bernard Chazelle, Joe Kilian, Ronitt Rubinfeld, and Ayellet Tal. The bloomier filter: an efficient data structure for static support lookup tables. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 30–39. Citeseer, 2004.

Reyhaneh Rabaninejad, Mahmoud Ahmadian, Maryam Rajabzadeh Asaar, and Mohammad reza Aref. A lightweight auditing service for shared data with secure user revocation in cloud storage. *IEEE Transactions on Services Computing*, 2019.

Sudha Anbalagan, Ali Kashif Bashir, Gunasekaran Raja, Priyanka Dhanasekaran, Geetha Vijayaraghavan, Usman Tariq, and Mohsen Guizani. Machine-learning-based efficient and secure rsu placement mechanism for software-defined-iov. *IEEE Internet of Things Journal*, 8(18):13950–13957, 2021. .

Shanghai Municipal Transportation Commission. 2019 shanghai traffic operation monitoring annual report. http://jtw.sh.gov.cn/xydt/20200514/e815c562c3 6a491ba741d576a9bcac1f.html.

Canlin Li, Huawei Huang, Yetong Zhao, Xiaowen Peng, Ruijie Yang, Zibin Zheng, and Song Guo. Achieving scalability and load balance across blockchain shards for state sharding. In *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, pages 284–294. IEEE, 2022.

Qin Wang, Jiangshan Yu, Shiping Chen, and Yang Xiang. Sok: Dag-based blockchain systems. *ACM Computing Surveys*, 2022.

Xian Rong Zheng and Yang Lu. Blockchain technology–recent research and future trend. *Enterprise Information Systems*, 16(12):1939895, 2022.