symmetry

MDPI

Article

EBAS: An Efficient Blockchain-Based Authentication Scheme for Secure Communication in Vehicular Ad Hoc Network

Xia Feng *, Kaiping Cui, Haobin Jiang and Ze Li

School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China; 2212004078@stmail.ujs.edu.cn (K.C.); jianghb@ujs.edu.cn (H.J.); 2222004131@stmail.ujs.edu.cn (Z.L.)
* Correspondence: xiazio@ujs.edu.cn

Abstract: A vehicular ad hoc network (VANET) is essential in building an intelligent transportation system that optimizes traffic conditions and makes traffic information conveniently accessible. However, malicious vehicles may disrupt the traffic order via propagating forged traffic/road information. Therefore, using digital certificates based on cryptography, some existing authentication schemes were proposed to manage vehicles' identities. At first glance, these schemes can effectively identify malicious vehicles. However, these schemes require more computation and storage resources to maintain certificates. This is because the data storage of the database increases in a near-linear trend as the number of certificates grows. In this paper, we propose an efficient blockchain-based authentication scheme for secure communication in VANET (EBAS) to address the aforementioned issues. In EBAS, the regional trusted authority (RTA) receives traffic messages uploaded by the vehicle, together with transactions constructed via the unspent transaction output (UTXO) model. The verifier checks the legitimacy of the single input contained in the uploaded transaction to verify the legitimacy of the message sender's identity. In terms of privacy preservation, a asymmetric key encryption technique, elliptic curve cryptography (ECC), is applied for constructing the transaction pseudonym, and users participate in the authentication process anonymously. In addition, our scheme guarantees the scalability of EBAS by proposing a transaction update mechanism, which can keep data storage at a stable level rather than near-linear growth. Under the simulation, the retrieving overhead remains at approximately 0.32 ms while the storage cost is stable at around 32.7 M for the blockchain state database. In terms of authentication efficiency, the average overhead of the proposed scheme is around 0.942 ms, which outperforms the existing schemes.

Keywords: vehicular ad hoc network (VANET); blockchain; authentication; asymmetric encryption; efficiency; scalability

Citation: Feng, X.; Cui, K.; Jiang, H.; Li, Z. EBAS: An Efficient Blockchain-Based Authentication Scheme for Secure Communication in Vehicular Ad Hoc Network. *Symmetry* **2022**, *14*, 1230. https://doi.org/10.3390/sym14061230

Academic Editor: Chin-Ling Chen

Received: 26 May 2022
Accepted: 7 June 2022
Published: 14 June 2022

1. Introduction

As the significant infrastructure of the Intelligent Transportation System (ITS), the vehicle ad hoc network (VANET) is the self-configuring network that has emerged as an advanced solution for improved driving safety and experience. VANET contains several heterogeneous entities, such as the trusted authority (TA), roadside units (RSUs), and vehicles [1]. Significantly, the vigorous development of wireless communication technology has allowed VANET to gain considerable attention from researchers in public and private sectors [2]. In VANET, each authorized vehicle can collect the time-critical road/traffic information and upload it to the TA or cloud server, to be utilized to analyze real-time road conditions.

However, a range of challenges and threats to information security and system availability are emerging [3–6], which is mainly reflected in three aspects. First, the messages should be authenticated by recipients. As mentioned in [7], malicious vehicles may propagate forged traffic/road messages via impersonating others, as well as fooling the trusted authority into accepting false or pointless information without being caught. Moreover, the privacy-preserving and anti-tracking aspects are not negligible in the authentication

process. Attackers may track a vehicle's trajectory to steal private information or fabricate traffic scenes [8]. Second, the proposed scheme must be efficient to satisfy the real-time nature of messages in VANET. Therefore, the algorithm contained in the proposed scheme must be efficiently executed while ensuring privacy security. Third, considering the system scalability, the proposed scheme should have an excellent information management mechanism to ensure that the scheme has stable performance and saves storage resources, especially based on the distributed system—the existing authentication scheme [9–11].

Numerous researchers have proposed various exploratory schemes to address information security issues [12–14]. For example, as mentioned in [12], one message is accepted by the TA if the same information is broadcast by at least τ vehicles, and the TA analyzes real-time road condition information with the assistance of received messages. However, the architectural design of these approaches brings unaffordable costs, such as more computing and storage resources.

Moreover, cryptography is utilized in privacy preservation during authentication [15–17]. Cryptography is a technique to convert plain text to ciphertext with the assistance of the key and algorithm [18], where the plain text is readable text and the ciphertext is unreadable text. Based on the cryptography technique, there are two types of authentication schemes: symmetric and asymmetric. In the schemes utilizing symmetric key cryptography, the same key is utilized in the encryption and decryption process of private information; on the contrary, in the asymmetric key cryptography scheme represented by public key infrastructure (PKI), the encryption and decryption processes are accomplished using public and private keys [19].

Blockchain technology originated from the paper "Bitcoin: A peer-to-peer electronic cash system", published by Satoshi Nakamoto in 2008 [20]. Blockchain has become an effective method of addressing vehicle management and data transfer security issues. Numerable researchers at home and abroad have proposed blockchain-based authentication schemes [11,21–23]. However, there are three limitations in existing approaches. First, due to the consensus mechanism, using smart contracts to accomplish the authentication process would incur extra time overhead. Second, the existing schemes are short of scalability, which may lead to a scheme without long-term stable performance. Third, the schemes utilize vehicles/RSUs as the mining nodes, which lacks consideration of the limited computing power and bandwidth.

From the above analysis, we can see that the existing approaches have the following restrictions. First, the existing methods have a prolonged delay, especially in authenticating and obtaining the related records. Second, the schemes lack sufficient scalability, which is an issue faced by most schemes, especially blockchain-based distributed authentication schemes. The data storage of the database increases as the number of certificates grows, which necessitates more storage resources. Third, there are linkability risks with real identities and certificates in the existing scheme. In this paper, to address the issues mentioned above, we propose *an efficient blockchain-based authentication scheme for secure communication in a vehicular ad hoc network* (EBAS), which provides the following functionalities:

- **Message authentication.** The vehicle generates a traffic message and uploads it to the Regional Trusted Authority (RTA) together with one transaction. The RTA can independently accomplish authentication by verifying the legality of the transaction. Furthermore, the RTA accepts the uploaded traffic message sent by vehicles successfully authenticated; otherwise, the message would be discarded.
- **Scheme scalability.** We proposed a transaction update mechanism in our scheme to enable scalability. With the assistance of the update mechanism, our scheme stabilizes the information retrieval efficiency of the system database and saves storage resources. Under the simulation, the retrieval overhead is maintained at around 0.32 ms while the storage cost is around 32.7 M.
- **Efficient authentication.** The transaction is generated based on the UTXO model. Therefore, based on asymmetric cryptography, the RTA checks the legitimacy of the message sender by verifying the validity of the single input contained in the transaction uploaded together with traffic messages. Under the simulation, one single

RTA can accomplish authentication within 0.942 ms. Moreover, compared with other related schemes, our proposal outperforms the existing common schemes.

- **Realization of the scheme prototype.** We simulate the proposed scheme on the *Hyperledger Fabric 2.0* and *Network Simulator 2*. In addition, we implement an exhaustive analysis of the efficiency and scalability of the proposed scheme. The security analysis shows that our scheme can defend against common attacks in VANET with the assistance of the asymmetric key encryption technique.

The remainder of this paper is organized as follows. Section 2 reviews the existing authentication schemes. Section 3 introduces the preliminaries and mathematical assumptions of the proposed scheme. The EBAS framework and the security model are formalized in Section 4. Section 5 introduces our EBAS scheme. In Section 6, we analyze the safety and efficiency of our EBAS scheme. Finally, Section 7 concludes the paper.

2. Related Research

Recently, numerous researchers have contributed a series of studies on the security and privacy-preserving issues in the VANET. In this section, we review some related research.

Numerous researchers have focused on addressing the reliability issues [12,24,25]. Threshold authentication is a promising technology to achieve reliability in VANET, which has received widespread attention. Specifically, this involves the transmission of information via a non-fully-trusted communication environment in VANET. The threshold mechanism allows the message to be accepted only when the number of confirmed vehicles exceeds the threshold value. Chen et al. [12] proposed a threshold anonymous announcement system. The recipient accepts the traffic/road messages when the number of vehicles that report the same message exceeds the threshold value. However, this scheme cannot revoke the certificates efficiently, which leads to the scheme being unable to resist frequent attacks. A one-time authentication and message-linkable group signatures scheme are proposed by Wu et al. [24], which can implement authentication efficiently. However, the scheme is inefficient for tracing doubtable messages because the process requires multiple expensive pairing operations. Lin et al. [25] proposed a roadside unit (RSU)-aided protocol to achieve the local detection and efficient traceability of malicious vehicles. However, the scheme is unsuitable for areas with sparse RSUs and cannot be bootstrapped by untrusted RSUs.

More and more safety threats [26–29] have drawn widespread attention. Zhang et al. [30] proposed a scheme of addressing the linkability issue in vehicular announcement networks with the assistance of a group signature. However, the same private key needs to be shared in one group, which is unsafe. Success et al. [31] proposed an autonomous privacy-preserving authentication scheme to guarantee the vehicle's traceability privacy. The vehicles can authenticate the messages securely and efficiently and renew their pseudonyms without interacting with trusted authorities. Jiang et al. [32] proposed a batch authentication scheme for message signatures based on a binary authentication tree. However, this scheme relies on the participation of semi-trusted RSUs. Ying et al. [33] proposed a lightweight authentication scheme. Based on the characteristic of fast calculation of a hash function, this scheme realizes mutual authentication among the OBU, RSU and TA. However, this scheme cannot effectively resist replay attacks and tampering attacks.

Blockchain [20] has promising adaptability in many fields. Plenty of existing schemes utilize blockchain and asymmetric key cryptography to mitigate the privacy and security issues in the VANET. Yao et al. [11] proposed a blockchain-based lightweight anonymous authentication scheme. This scheme can satisfy security requirements such as anonymity, authentication, and integrity. However, this scheme does not consider the linkability of vehicles during the authentication process. Attackers can track vehicles based on static pseudonyms, leading to the disclosure of vehicles' private information. Lu et al. [21] propose an authentication protocol utilizing the Merkle Patricia Tree (MPT) as the underlying data structure, which is efficient and can save storage resources. However, the scheme cannot provide excellent scalability, and data processing in upper nodes may lead to more

time overhead following the data storage increase in MPT. Lei et al. [22] utilize blockchain to predigest the key management. Arora et al. [23] proposed a blockchain-based authentication scheme. However, this scheme relies on a centralized authority to implement the vehicle registration process, which is prone to cause a single point of failure problem.

3. Preliminaries and Mathematical Assumptions

3.1. UTXO Model

UTXO is a special pattern of currency circulation. The transaction constructed with the UTXO model mainly relies on an $InputSet$ and $OutputSet$ to accomplish the currency circulation. Furthermore, the $InputSet$ and $OutputSet$ contain several $Inputs$ and $Outputs$ separately. We elaborate the details in the following section. Tables 1 and 2 present the notations and definitions covered in this section.

Table 1. Basic notations of UTXO structure.

Notation	Definition
PID_t	Transaction index.
pk, sk	The key pair of the user.
σ^{sk}	A signature signed by the secret key.
H_{pk}	The hash value of public key.
V	The transaction value for one object.
N_{out}	The sequence number of the output.
$Hash_{lock}$	A representative value for the transaction output object.

Table 2. Cryptographic algorithms of UTXO structure.

Notation	Definition
$HashPubKey(pk) \rightarrow Hash_{lock}$	Calculating the hash value of the public key.
$Compare(Hash_{lock}, H_{pk}) \rightarrow 0, 1$	Comparing the public key hash value H_{pk} and transaction output object representative value $Hash_{lock}$ for consistency.
$Verify(pk, \sigma^{sk}, PID_t) \rightarrow 0, 1$	Verifying the signature. If the verification result is true, it returns 1; otherwise, it returns 0.

3.1.1. Output

$Output$ provides transaction object information, which contains two elements, V, H_{pk}:

$$Output \leftarrow (V, H_{pk}) \tag{1}$$

where H_{pk} is the hash value of the transaction object's public key, which represents the target of this output, and V represents the transaction value for one object. Notably, both H_{pk} and V are public information.

3.1.2. Input

The $Input$ of the current transaction is generated based on the previous transaction $Output$. We illustrate the specific composition of $Input$.

$$Input \leftarrow (PID_t, N_{out}, pk, \sigma^{sk}) \tag{2}$$

$Input$ contains four elements, PID_t, σ^{sk}. PID_t represents the retrieval index of the transaction. The $Input$ of the current transaction is generated based on the previous transaction $Output$. Therefore, N_{out} is utilized to mark the position of the $Output$ in the output set of the previous transaction. pk is the public key to verify the signature σ^{sk}, and the σ^{sk} is the signature of PID_t using the secret key corresponding the H_{pk}, which is utilized to prove the legitimacy of the output that is marked by N_{out}.

3.1.3. Verification Mechanism

Generally, the $InputSet$ consists of multiple $Input$. The verifier confirms the legitimacy of the transaction through the following operations. First, it compares whether the hash value of pk in the current $Input$ and H_{pk} in $Output$ that is marked by N_{out} are consistent. Second, it verifies the signature with the pk. We explain the verification operation in Algorithm 1.

Algorithm 1 Verification Algorithm.

Require: R.
Ensure: *true* or *error_code* .
1: $R \leftarrow (PID_t,\ pk,\ H_{pk},\ \sigma^{sk})$
2: **if** formal check on the R is ok **then**
3: $Hash_{lock} \leftarrow HashPubKey(pk)$
4: **else return** *error_code*
5: **if** $Compare\ (Hash_{lock},\ H_{pk}) \stackrel{?}{=} true$ **then**
6: **if** $Verify\ (pk,\ \sigma^{sk}) \stackrel{?}{=} true$ **then**
7: **return** *ture*
8: **end if**
9: **end if**
10: **end if**
11: **return** *error_code*

Algorithm 1 involves three main steps, described as follows.

Step 1-1 (Step 1 in Algorithm 1): The user sends the tuple R to the object located within its communication range.

$$R = (PID_t, pk, H_{pk}, \sigma^{sk}) \tag{3}$$

After receiving the message, the verifier performs a formal check on the R, which is to confirm the integrity of the message. After this, Equation (4) is leveraged to calculate the hash value of pk.

$$Hash_{lock} = HashPubKey(pk) \tag{4}$$

The function $HashPubKey$ is utilized to compute the hash value of the public key, which is constructed in a smart contract. Specifically, we implement the hash computation in the smart contract by considering the relevant functions in the standard cryptography library. After the smart contract is deployed to the blockchain, the corresponding public key hash value is obtained by considering the smart contract and using the pk as the input parameter of the smart contract.

Step 1-2 (Step 5 in Algorithm 1): Comparing whether the hash value of pk in the current $Input$ and H_{pk} in $Output$ that is marked by N_{out} are consistent. Here, Equation (5) is leveraged to determine whether $Hash_{lock}$ and H_{pk} are equal.

$$Compare(Hash_{lock}, Hash_{pk}) \stackrel{?}{=} true \tag{5}$$

Step 1-3 (Step 6 in Algorithm 1): Next, Equation (6) is utilized to verify the validity of the signature based on the asymmetric cryptography.

$$Verify(pk, \sigma^{sk}) \stackrel{?}{=} true \tag{6}$$

Thus, the recipient can verify whether the transaction is valid with the verification mechanism.

3.2. Asymmetric Key Encryption

Symmetric and asymmetric encryption are the two basic forms of cryptographic encryption applications [34]. The asymmetric cryptographic system uses key pairs, the public

and private key. It is worth noting that the public key can be broadcast openly in the business. However, the private key is not publicly available information, and it is critical for the key owner to keep the private key secret. Generally, based on the large prime numbers, the asymmetric keys can be generated with cryptographic algorithms, which is further utilized in constructing the one-way cryptographic algorithm to achieve the asymmetric encryption [35]. Among various asymmetric encryption types, elliptic curve cryptography (ECC) is a lightweight asymmetric key cryptography method for data encryption and decryption that has received extensive attention and application.

3.3. Mathematical Assumptions

We elaborate the following assumptions as the basic requirements for our scheme.

- **Elliptic Curve Discrete Logarithm Problem.** Given a prime number q and one elliptic curve E, one point Q in the selected curve E satisfies $Q = xP$, where $P, Q \in G$. G is the additive cyclic group with prime order q, and P is the generator of G. If given parameter P and Q, it is difficult to determine x.
- **Computational Diffie–Hellman Problem.** Given a prime number p, q and one elliptic curve E, two points Q and V in the selected curve E satisfy $Q = xP$ and $V = yP$, where $P, Q, V \in G$. G is the additive cyclic group with prime order q, and P is the generator of G. It is difficult to compute $xyP \in G$ without $x, y \in Z_p^*$.

4. Definitions

4.1. Framework of EBAS System

In this section, we illustrate the architecture and design goals of our proposal, which is shown in Figure 1. Our proposal contains two layers, the computing layer and the user layer. The computing layer contains the RTAs and RA. The user layer contains the RSUs and OBUs.

- **The On-Board Unit (OBU).** The On-Board Unit (OBU) is the hardware support to assist vehicles in realizing wireless communication. The vehicle equipped with the OBU acts as the launcher of the authentication request. Moreover, the OBU contains a tamper-proof device, which is utilized to store confidential material, such as the secret key [36].
- **Roadside Unit (RSU).** The RSUs are stationary devices deployed along the road or intersections. RSUs are responsible for broadcasting essential messages within their communication range, such as the RTA's public key. Moreover, the RSU can support vehicles in accomplishing the process of traffic message uploading when the network environment is poor.
- **Regional Trusted Authority (RTA).** The RTA is responsible for checking the validity of received messages and authenticating the message sender's identity. Moreover, RTAs have sufficient computing power and act as the consensus nodes to maintain the blockchain. After the authentication, RTAs need to preprocess the traffic messages and issue license coins to vehicles.
- **Root Authority (RA).** The RA is the crucial institution that is responsible for analyzing real-time road conditions based on the traffic information preprocessed by the RTA. After this, the RA can make rational responses to critical traffic situations after being informed by the RTA [7]. Moreover, each vehicle participating in the VANET must be registered in the RA. Meanwhile, the RA is the only institute that can expose the real identity of vehicles. Together with the RTAs, the RA has sufficient computing power and acts as the consensus node to maintain the blockchain.

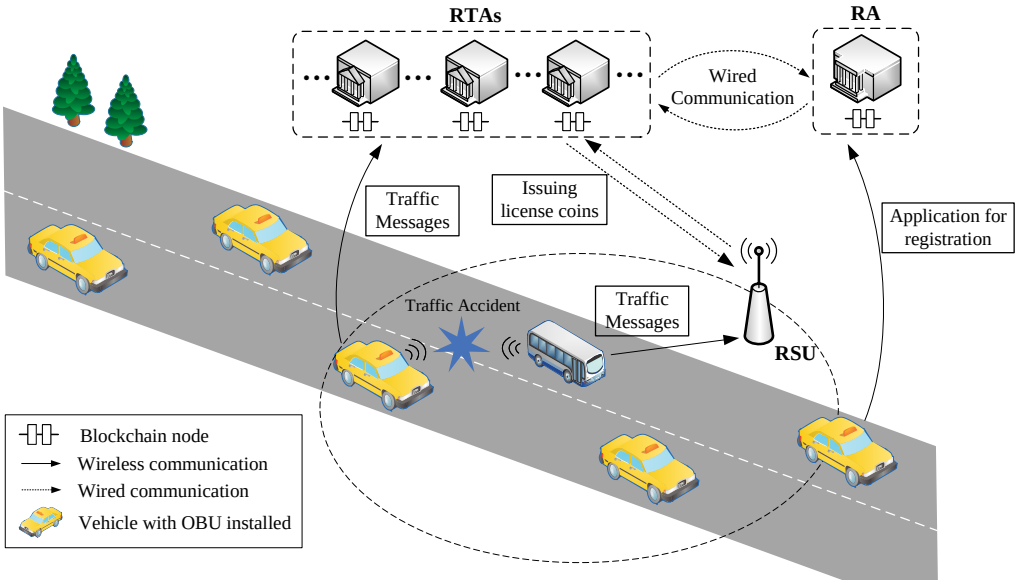

Figure 1. The overview of EBAS.

4.2. Formal Definitions of the Transaction and License Coin

4.2.1. Transaction Structure

As in Figure 2, we illustrate several parts contained in the transaction. Specifically, the *TransactionPseudonym* represents the transaction index, which is generated based on the asymmetric encryption ECC. It would be utilized in retrieving transactions stored in the blockchain state database. Moreover, the Root Authority (RA) can expose the real identity of malicious vehicles based on the *TransactionPseudonym*. *ExpirationTime* represents the expiration time of the transaction. *TransactionType* represents the business type that the transaction would be used for, such as authentication, aggregating license coins, etc. *Timestamp* and *Nonce* denote the timestamp and sequence number of transaction generation, respectively. *InputSet* and *OutputSet* contain several *Inputs* and *Outputs* separately, as elaborated in Section 3.1. *Other* records extra information about the transaction.

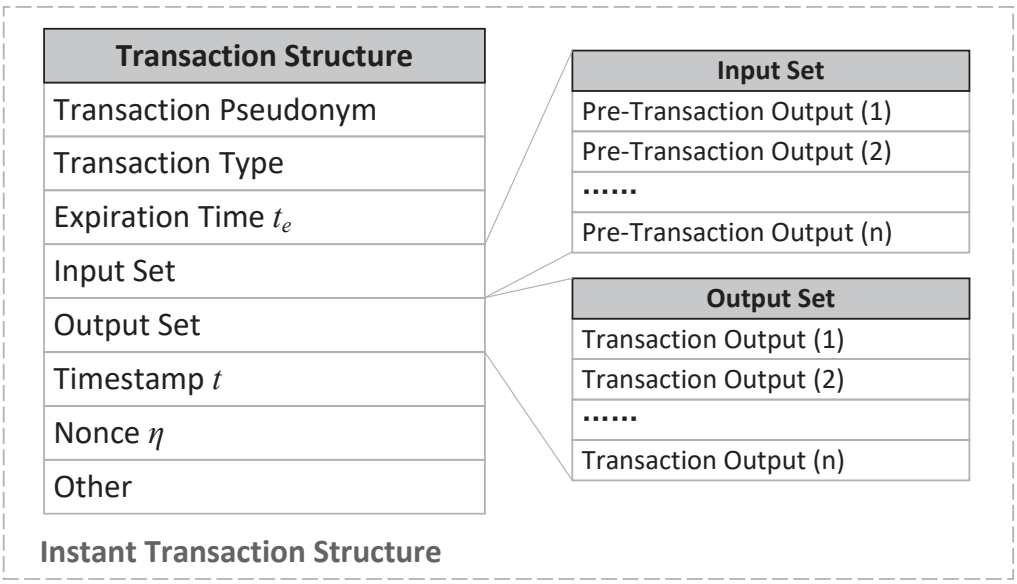

Figure 2. The instant transaction structure.

4.2.2. Transaction Function

There are four transaction categories: instant transaction, authentication transaction, aggregation transaction, and original transaction. Specifically, the instant transaction is

generated with no specific functions. When performing one business task, the instant transaction would be redefined as other transaction types. For instance, the instant transaction is utilized to participate in the authentication process, which would be redefined as an authentication transaction. The instant transaction is used to aggregate the license coins owned by users, which would be redefined as the aggregation transaction. After a specific business task, the RTA would upload the transaction to the blockchain state database. The transactions stored in the database are redefined as original transactions, which would be utilized to generate one newest instant transaction.

In our scheme, the aggregation transaction significantly affects the authentication efficiency. The transaction is generated based on the UTXO model, and the verifier needs to check the legitimacy of all $Input$ contained in the $InputSet$ during the authentication process. Therefore, the verification for the single $Input$ needs to be replayed multiple times, which consumes extra computing resources. Moreover, all $Input$ in the $InputSet$ are sourced from various previous transactions. These previous transactions need to be stored separately, which would consume additional storage resources. The aforementioned problems negatively affect the authentication efficiency.

A license coin aggregation mechanism is proposed to address the above issues. The user could generate one instant transaction to aggregate license coins. All licensed coins owned by the user from different outputs would be contained in the $Input$ of this latest instant transaction. It is worth noting that this latest instant transaction performs the aggregation function with one output. After this, this aggregation transaction would be stored in the blockchain state database. Obviously, based on this aggregation transaction with a single $Output$, the next instant transaction generated by the user would contain one $Input$, which decreases the computing resources and time overhead during the authentication process.

4.2.3. License Coin

We define license coins as the circulating currency of the EBAS system. The license coins are generated by the RTA. The RTA issues license coins to newly registered vehicles or legitimate vehicles involved in the authentication via one transaction. Specifically, the RTA determines the amount of licensed coins to issue by setting the value of V in the transaction $Output$. It is worth noting that, based on the UTXO mechanism, the vehicle can only use the $Output$ information in the original transaction to generate the $Input$ of the newest instant transaction. Therefore, vehicles cannot issue or counterfeit license coins. Moreover, the vehicle can participate in the authentication process via consuming a certain amount of license coins. On the contrary, the vehicle cannot generate a valid transaction without license coins and cannot participate in the authentication process.

Generally, each transaction generated based on the UTXO model has multiple $Inputs$ and $Outputs$. The authentication is the outflow process, and the license coins are transferred from the vehicles to the RTAs. On the contrary, the new coins are issued, and the license coins are transferred from the RTAs to the vehicles, which is the inflow process. In addition, because the transaction $InputSet$ and $OutputSet$ contain multiple $Input$ and $Output$, respectively, the license coins can be transferred between various entities through one transaction.

4.3. Formal Security Definitions

Our scheme assumes that the adversary cannot break mathematical assumptions and standard cryptographic primitives. In order to ensure the communication security of the VANET, the proposed scheme should defend against several safety risks in the VANET and satisfy several design goals.

- **Robustness.** The RTAs and RA act as the consensus nodes to jointly maintain the blockchain state database in our proposal. Therefore, the system could deal with the business normally when one RTA is in a power failure or downtime condition.

- **Confidentiality.** The confidentiality issue of users contains two types: identity and location. The proposed scheme should guarantee that a vehicle's real identity information can be kept secret. Moreover, because of the open nature of VANETs, internal attackers can obtain other vehicles' action information by analyzing the broadcast messages in VANETs [37]. Considering the confidentiality of a user's location information, the proposed scheme should have anti-tracking capabilities and achieve unlinkability between real identities and transactions.
- **Efficiency and scalability.** Generally, there are a range of authentication requirements that need to be addressed by the RTA. Thus, the scheme should have an efficient transaction verification algorithm. Considering the scalability, the proposed scheme should have an effective data management mechanism to address the conflict between information retrieval efficiency and storage rises.
- **Replay attack.** The replay attack is a type of man-in-the-middle attack, where the adversary maliciously repeats a valid data transmission [38]. In our scheme, the adversary creates one instant transaction to participate in the authentication process, and the transaction's $Input$ is generated based on one previous transaction output that has been used in another transaction.
- **Sybil attack.** The adversary utilizes multiple identities to communicate with other entities simultaneously. Moreover, the adversary can use several identities to broadcast a series of false information and disturb the traffic order.
- **Identity revealing attack.** The attacker illegally obtains the vehicle's real identity information. Furthermore, the attacker can easily track the entity based on the real identity information.
- **Location tracking attack.** The attacker obtains the user's trajectory by analyzing location and path information, which causes a threat to users' privacy.
- **Repudiation attack.** The entity denies messages or facts that have been presented. Anonymous authentication techniques may be abused by malicious users to escape from their liabilities.

5. EBAS: An Efficient Blockchain-Based Authentication Scheme for Secure Communication in Vehicular Ad Hoc Network

Entities of the user layer must be registered in the RA. After the registration, the vehicles have the qualification to generate the instant transaction and participate in the authentication process. We illustrate the entity registration process and several mechanisms related to the authentication in this section. Moreover, the notations and definitions are listed in Table 3.

Table 3. Basic notations and definitions of system.

Notation	Definition
PS_{trans}	The retrieval index of the transaction.
$Cipher_{id}$	Encrypted ciphertext of identity information with master key.
E_{id}	Vehicle's pseudonym.
pk_m, sk_m	The master key pair of system.
pk_r, sk_r	The key pair of RTA.
pk_v, sk_v	The key pair of vehicles.
t	Timestamp of transaction execution.
C_r	Codes of the region where entities are located.
$Trans_{au}$	The instant transaction for authentication.
M	Uploaded traffic messages.
σ_v^{au}	The signature for t and M generated by vehicle's secret key.
$nonce$	The serial number of uploaded message.

5.1. Initialization

In the system initialization process, the RA is responsible for generating master key pairs and accomplishing user-layer entity registration.

5.1.1. The Master Key Pair Generation

In our scheme, the user's private information (e.g., identity) is encrypted with the master key pair (sk_m, pk_m). Moreover, the transaction pseudonym is also generated with the public key pk_m. The master key pair generation details are as follows:

- The RA selects an elliptic curve E: $y^2 = x^3 + Ax + B \ mod \ p$, where $p > 5$ is a prime, $A, B \in \mathbb{Z}_p$ and constants with $4A^3 + 27B^2 \neq 0$. Let $\mathrm{E}(\mathbb{Z}_p)$ denote the set of pairs $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ along with O, which is the point at infinity. The RA generates the master key pair (sk_m, pk_m) based on elliptic curve cryptography (ECC) [39].
- The RTA chooses a random number as its secret key $sk_r \in \mathrm{E}(\mathbb{Z}_p)$, and then computes the corresponding public key $pk_r = sk_r \times \mathrm{E}(\mathbb{Z}_p)$.
- The RTA stores the key pair (sk_r, pk_r) locally and broadcasts via the RSUs.

In the master key pair generation procedure, the RA first selects an elliptic curve E. Based on the curve E, the RA generates the master key pair (pk_m, sk_m), and the RTA generates the key pair (pk_r, sk_r). The master key is mainly utilized to encrypt the user identity, and the RTA's key is mainly utilized in the transaction generation.

5.1.2. User-Layer Entity Registration

In effect, each RTA has a disjoint management region. Each entity participates in the registration process by submitting its real identity information to the RA. The RA generates the entity's pseudonym via encrypting the received real identity information with the master key.

After this, the RA forwards the registration results to the corresponding RTA. The RTA utilizes one registration transaction to issue n license coins to the registered entity to initialize authentication permissions. Moreover, this registration transaction would be uploaded to the database as the original pioneer transaction. The details of the user-layer entity registration are as follows:

- The entity in the user layer submits its real identity material to the RA.
- The RA verifies the legitimacy of the received material. After this, the RA generates the E_{id} via encrypting the entity's real identity information with the master key.
- The RA gives authorization to the user for the transaction generation.
- The RA forwards the registration results to the corresponding RTA. The RTA utilizes one registration transaction to issue n license coins to the registered entity to initialize authentication permissions, which can be utilized in the authentication process.

We define the generation of E_{id} as follows:

$$Cipher_{id} = E_{pk_m}(m_{id} || t) \tag{7}$$

$$E_{id} \overset{def}{=} Slice(Cipher_{id}) \tag{8}$$

where t is the timestamp of the entity registration. m_{id} represents the entity's real identity information. E_{pk_m} represents the encrypted operation. $Cipher_{id}$ is ciphertext. The operation of the function *Slice* is to intercept the first 20 characters of the target field (hexadecimal), which is utilized to generate E_{id}.

Finally, the RA stores the entity's E_{id} and $Cipher_{id}$ as key value pairs in the blockchain state database. It is worth noting that the RTAs and RA are the consensus node, so each RTA can retrieve the E_{id} in the blockchain state database.

5.2. Transaction Generation

There are two computation operations in the transaction generation process, key pair generation and transaction pseudonym generation. The details are as follows.

5.2.1. Key Pair Generation

In our scheme, the newest instant transaction is generated based on the original transaction. The license coin information stored in the original transaction $Output$ would be utilized in the newest instant transaction $Input$. The details of key pair generation are as follows:

- The vehicle calls the instant transaction generation algorithm.
- The OBU chooses a random number as its secret key $sk_v \in \mathrm{E}(\mathbb{Z}_p)$, and computes the corresponding public key $pk_v = sk_v \times \mathrm{E}(\mathbb{Z}_p)$.
- The vehicle stores the key pair (sk_v, pk_v) locally.

After this, the secret key is stored in the tamper-proof device. Moreover, we assume that this phase has no privacy disclosure and security attack risk threat.

5.2.2. Transaction Pseudonym Generation

The transaction pseudonym acts as the retrieval index of the transaction. Moreover, the transaction pseudonym is utilized for conditional privacy protection and non-reputation. The vehicle generates the transaction pseudonym using the RTA public key broadcast periodically via the RSU. The details of transaction pseudonym generation are as follows:

$$EC_{trans} = E_{pk_r}(E_{id}||t||C_r) \tag{9}$$

$$PS_{trans} \stackrel{def}{=} Slice(EC_{trans}) \tag{10}$$

C_r is the regional code where the vehicle is located. E_{pk_r} is an asymmetric encryption operation based on the RTA public key pk_r. PS_{trans} is the transaction pseudonym. PS_{trans} and the ciphertext EC_{trans} are recorded in the blockchain state database as key value pairs.

5.3. Message Authentication

In the following section, we illustrate the authentication procedure and issuance of license coins. The details are as follows.

5.3.1. Authentication Operation

We elaborate the authentication between the entity of the user layer and the RTA.

- **Step1:** The vehicle sends the message $\langle Trans_{au}, t, nonce, M\rangle$ to the RTA.
- **Step2:** Based on the received $Trans_{au}$, the RTA retrieves the original transactions involved in the $Input$ in the blockchain state database. Suppose that the retrieval procedure is successful, as described in Section 3.1.3. In this case, the RTA checks the validity of the signature based on the asymmetric cryptography and compares the hash value of the public key with the $Hash_{lock}$ stored in the original transaction.

M is ciphertext for the traffic report, which is generated with pk_r. The signature σ_v^{au} for $(t||M)$ is generated using sk_v. The σ_v^{au} and sk_v are contained in the $Input$ of $Trans_{au}$. Afterward, based on the transaction update mechanism, the vehicle generates one newest instant transaction to implement the next authentication or license coin aggregation.

5.3.2. Issuance of License Coins

The RTA will issue license coins to vehicles successfully authenticated, and the details are as follows:

- **Step1:** The RTA implements the authentication operations.
- **Step2:** The RTA generates one instant transaction locally and calculates the *Lockinghash* based on the vehicle's public key.
- **Step3:** The RTA sends the instant transaction to the RTAs and stores it in the blockchain state database. The RTAs broadcast the received instant transaction in the management region.
- **Step4:** Vehicles accomplishing the authentication process execute the license aggregation procedure.

5.4. Transaction Confirmation

We define the process of checking the legitimacy of the transaction as transaction confirmation, which contains several operations, such as retrieve, verification, and storage. After the transaction confirmation process, the original transaction would be replaced by the latest verified transaction. The details are as follows:

- In the blockchain state database, the previous transaction involved in *Input* would be retrieved through the transaction pseudonym.
- The system verifies the signature and compares the hash value of public key, which has been illustrated in Section 3.1.3.
- The transaction would be stored in the database as the latest original transaction to replace the previous one.

5.5. Aggregation Transaction

The RTA will issue license coins to the corresponding vehicle successfully authenticated through one instant transaction. The vehicle utilizes the aggregation transaction to aggregate all owned coins existing in other transactions. Based on the other transactions, the vehicle constructs several *Input* contained in the aggregation transaction. Afterward, the vehicle constructs aggregated transaction *Output* based on the local key pair. This aggregation transaction is uploaded to the RTA, and the RTA executes the transaction authentication process. Finally, this aggregation transaction would be stored in the blockchain state database as the newest original transaction.

5.6. Transaction Update

As shown in Figure 3, we define the conversion between different types of transactions as an update process. Based on the transaction update mechanism, our proposal can guarantee scalability. The details of the transaction update are as follows:

- The instant transaction is utilized in authentication or in aggregating license coins, and the instant transaction is redefined as an authentication transaction or aggregation transaction.
- Authentication or aggregating license coins would be accomplished via the transaction confirmation process, and the transaction would be the newest original transaction stored in the blockchain state database.
- Based on the original transaction, the user can construct the *Input*, which would be contained in one newest instant transaction.
- Afterward, the newest instant transaction can be utilized in the next authentication process or license coin aggregation.

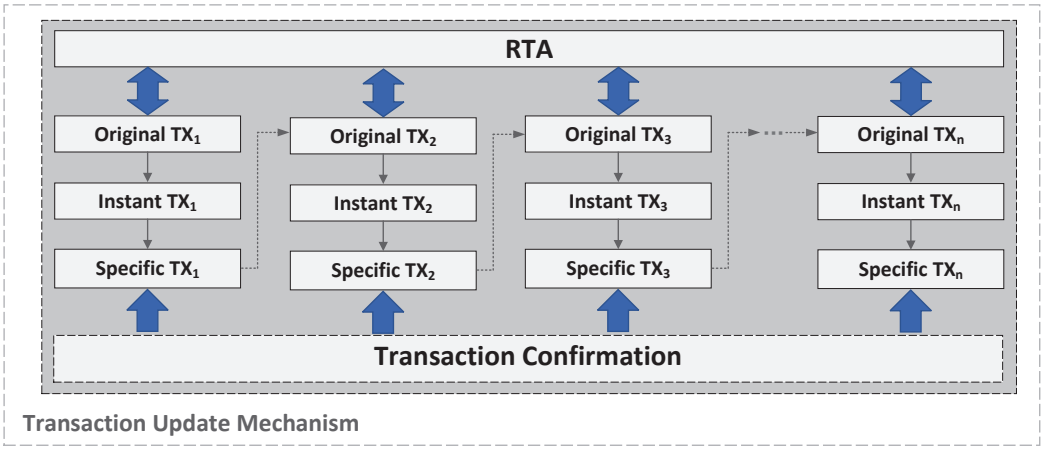

Figure 3. The transaction update mechanism.

Through the transaction update process, we achieve the replacement of old and new transaction data stored in the blockchain state database, thereby guaranteeing the scalability of the storage level.

6. Security Analysis and Performance Evaluation

In this section, we implement the performance evaluation and prove that our scheme can effectively defend against several common security threats.

6.1. Security Analysis

Theorem 1. *Our proposal can resist the replay attack and the Sybil attack.*

Proof. During the transaction confirmation procedure, the RTA needs to retrieve the original transaction. Moreover, the original transaction would be replaced after the transaction confirmation process. Therefore, the attacker cannot utilize one previous $Output$ from the transaction that is not stored in the blockchain state database to generate one valid transaction. Moreover, each transaction has its expiration time and timestamp. Our scheme can effectively resist replay attacks.

Based on the transaction update mechanism, the newest instant transaction generation is executed based on the original transaction. Therefore, the attacker cannot forge multiple transactions to participate in the authentication process. Therefore, our scheme can effectively resist Sybil attacks. □

Theorem 2. *Our scheme can effectively achieve privacy preservation in the authentication process, including defense against identity-revealing attacks and locating tracking attacks.*

Proof. In our proposal, the user's real identity information is encrypted and stored by the RA. The RA has sufficient security levels, and the attacker cannot acquire the user's real identity by obtaining the RA's private key or breaking standard cryptographic primitives. Moreover, in the generation process of the PS_{trans}, timestamp fields facilitate no critical connection between $Cipher_{id}$ and the real identity. Therefore, the proposed scheme achieves privacy-preserving and anti-tracking capabilities. □

Theorem 3. *Our scheme can ensure the non-repudiation of published messages.*

Proof. When a selfish or malicious entity denies presented messages or facts, its E_{id} can be exposed via decrypting the EC_{trans}. Therefore, the RTA can track the malicious entity with the assistance of the E_{id}. Furthermore, the RTA forwards the E_{id} to the RA, and the RA can expose the real identity information of the malicious entity. Therefore, our scheme can effectively ensure the non-repudiation of published messages. □

6.2. Performance Evaluation

In this subsection, through theoretical analysis and simulation, we evaluate the performance of the proposed scheme. We deploy the Hyperledger Fabric V2.0.0 on Ubuntu V16.04 running on a machine with Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz and 4 GB RAM. Furthermore, we construct a consortium blockchain with four peer nodes belonging to two organizations, namely $org1$ and $org2$, three order nodes, and one client node. The operations involved in our scheme are executed via *chaincode*. Moreover, we implement the communication simulation using NS-2, and measure the related communication delay.

LevelDB is selected as the blockchain state database (statedb) to store the original transactions in the form of $key/value$. It is worth noting that the key utilized to retrieve the database is irregular and discontinuous, owing to the discreteness of the transaction pseudonym. However, the *LevelDB* has an excellent performance in reading/writing continuously, while it is poor for random keys [40]. Therefore, the scalability and efficiency of our scheme are influenced by the statedb performance. We address the above issues

and present the related simulation in the following section. In addition, we evaluate the authentication overhead and compare our scheme with related others.

6.2.1. Storage Cost and Retrieval Overhead

We implement a series of simulations to estimate the storage cost in the proposed scheme. As Figure 4 shows, original transactions are scaled from 10 to 12×10^4, and the storage cost shows an upward tendency with a non-linear trend. Specifically, when original transactions are in 10^5, the static storage cost is 42.3 M, and it shows an upward trend following the increase in the number of original transactions. Moreover, based on the retrieval properties of the blockchain state database, we further simulate the retrieval performance under various orders of storage magnitude, which is shown in Figure 4. Obviously, the time overhead of the retrieval operation is ever-mounting with the increase in storage. In particular, the retrieval overhead presents a clear upward trend from the original transaction in 60 k. Furthermore, when the storage is large enough, the time delay shows an irregular upward trend, and the retrieval efficiency is degraded [40].

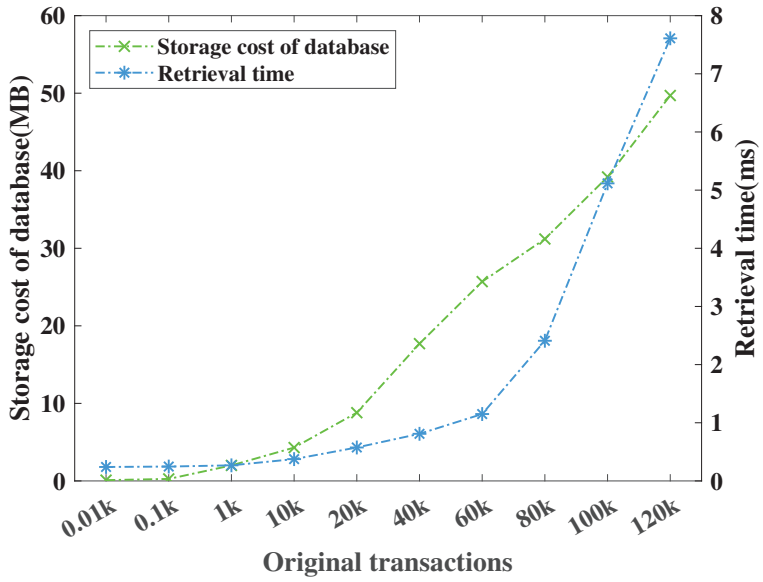

Figure 4. The storage cost and retrieval overhead.

Besides, as shown in Figure 5, we also compare the retrieval overhead when the number of vehicles varies. The single vehicle accomplishes the retrieval operation within 8 ms under the condition of storage in 12×10^4. Based on the simulation, the retrieval overhead is tolerable for the authentication process.

6.2.2. Analysis of Scalability

In our scheme, we construct a transaction update mechanism to maintain the storage of the blockchain state database at a stable level. The RTA needs to store the data in the blockchain state database with extra time overhead when the transaction confirmation process is executed. Without the transaction update mechanism, we observe that the amount of data stored in the database and the retrieval time overhead show a near-linear upward trend as the number of transaction confirmations grows. Under the simulation, the storage cost increases by 5.8 M when the number of transaction confirmations is scaled from 0 to 10,000. In addition, as the statedb has a data compression mechanism [41], the curve of storage cost has three descending processes, as shown in Figure 6a.

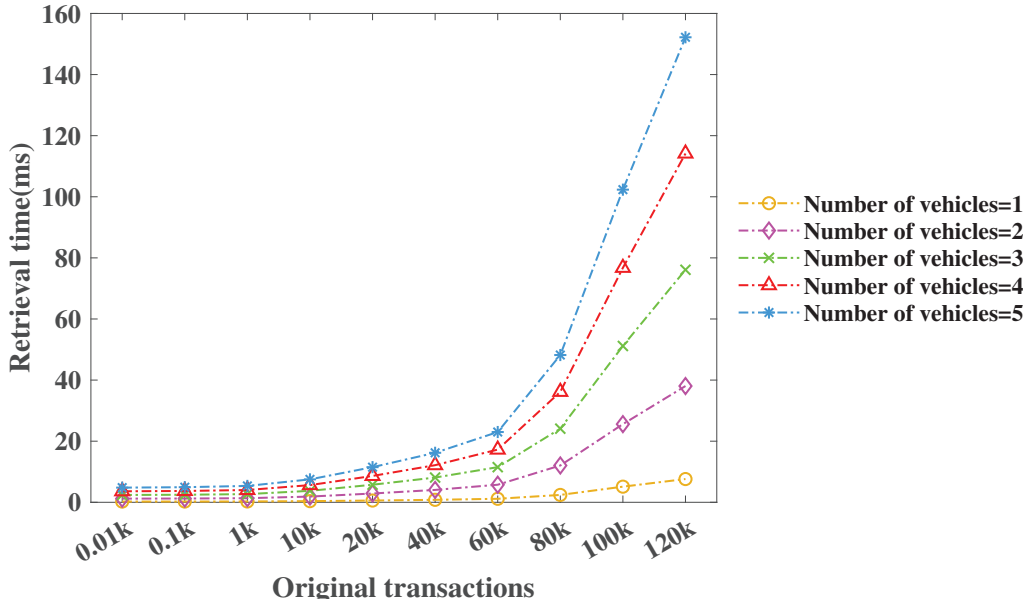

Figure 5. The retrieval overhead of multi-vehicles participating in authentication process.

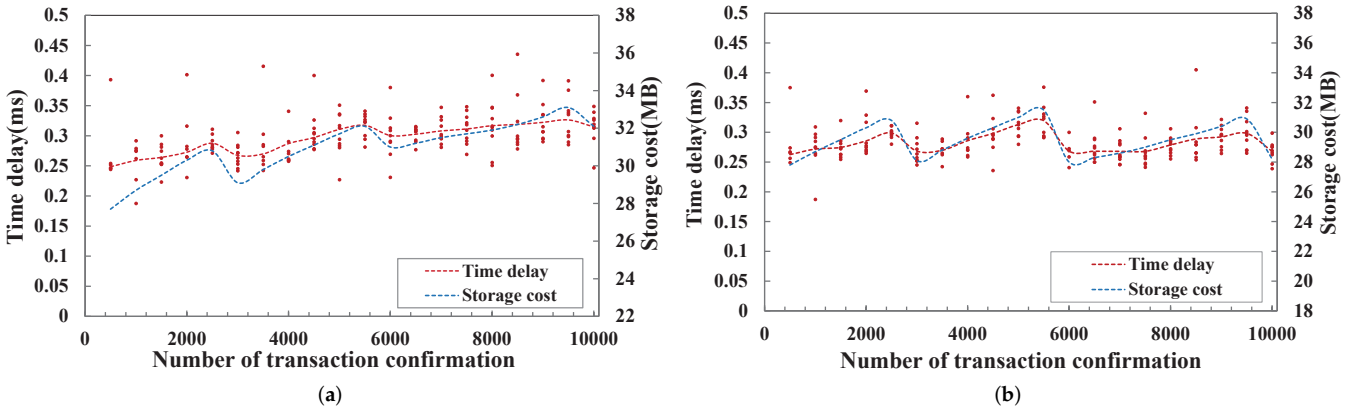

Figure 6. The storage cost and retrieval overhead of the blockchain state database. (**a**) presents the retrieval overhead and storage cost of statedb without the transaction update mechanism; (**b**) presents the retrieval overhead and storage cost of statedb with the transaction update mechanism.

Based on the transaction update mechanism, we observe the time overhead and storage status of statedb with the constructed system under existing facility conditions, as shown in Figure 6b. Contrary to Figure 6a, the storage of the statedb shows a dynamic and stable status as the number of transaction confirmations increases. Specifically, the retrieval operation is executed within 0.32 ms, while the storage cost is stable at around 32.7 M. Therefore, the storage cost would remain in a stable status under a certain vehicle order of magnitude. There would be no necessary connection between retrieval overhead and the number of transaction confirmations, and our scheme can efficiently achieve scalability.

6.2.3. Efficiency Analysis of License Coin Issuance

Generally, the RTA constructs one transaction with a single output to issue license coins for each vehicle. However, it would cause unacceptable storage costs and communication overhead when more vehicles need to issue license coins. Based on the UTXO model, the RTA can create the transaction containing multi-outputs to accomplish the license coins' issuance when there are multiple issuance objects. Therefore, we simulate the issuance process of license coins when transactions have different numbers of $Output$, as shown in Figure 7. Under the simulation, the storage cost of the transaction with the multi-outputs does not alter significantly. On the contrary, if the RTA constructs one transaction with a single output to issue license coins for each

object, the storage cost of the transaction with a single output increases linearly with the number of objects. In addition, the communication overhead increases with the number of outputs in the transaction. The communication overhead is 57.21 ms and 300.23 ms when there are 10 and 100 objects receiving license coins, respectively.

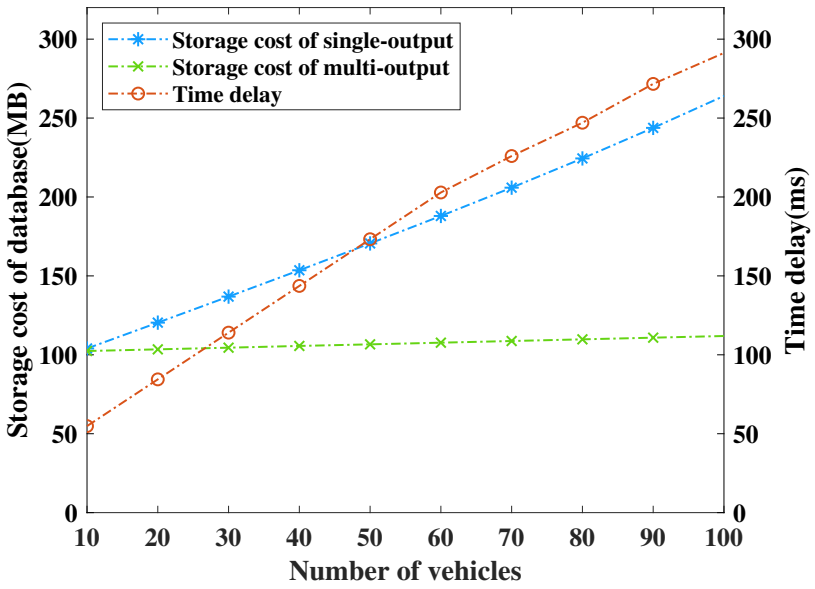

Figure 7. The storage cost and communication overhead of license coin issuance.

6.2.4. Authentication Overhead

We evaluate the authentication overhead of our scheme for both the transaction confirmation process and authentication of users.

The transaction confirmation process mainly contains two steps, public key matching and signature verification. We calculate the time overhead, which is T_{mat}^{Au} and T_{ver}^{Au}, respectively. The calculation details are as follows:

$$T_{mat}^{con} = T_h + T_{mat} \tag{11}$$

$$T_{ver}^{con} = T_{ret} + T_{ver} \tag{12}$$

The T_{ver} is the time overhead to implement the signature verification, which is stable at around 0.62 ms. T_{ret} is the retrieval overhead, which is around 0.32 ms. Moreover, T_h is the time overhead for executing one hash operation. T_{mat} is the time overhead for comparing the hash value of the public key and *Lockinghash* stored in the original transaction output. Moreover, the RTA may have a number of verification missions simultaneously. Therefore, we compare the time overhead $T_{mat}^{Au} + T_{ver}^{Au}$ when there are multiple input loads in the transaction, as Figure 8 shows. The time overhead is 31 ms when there are 50 transaction verification requests (Input Num=1). In general, the time overhead shows an upward linear trend as the number of transaction verification requests increases.

The authentication process for users mainly includes the transaction generation process, communication process, and transaction confirmation process. We give details about the time overhead for the authentication process for users. Here, Equation (13) is the time overhead of a single authentication process without an aggregation mechanism. On the contrary, Equation (14) is the time overhead of a single authentication process with an aggregation mechanism. It is worth noting that the authentication transaction contains only one input (Input Num = 1) with the assistance of the license coin aggregation mechanism. Therefore, the authentication in our scheme can be accomplished more efficiently compared with multi-inputs. Under the simulation, considering the communication overhead, the sin-

gle authentication is implemented within 29.142 ms. Figure 9 shows the authentication overhead considering the communication delay.

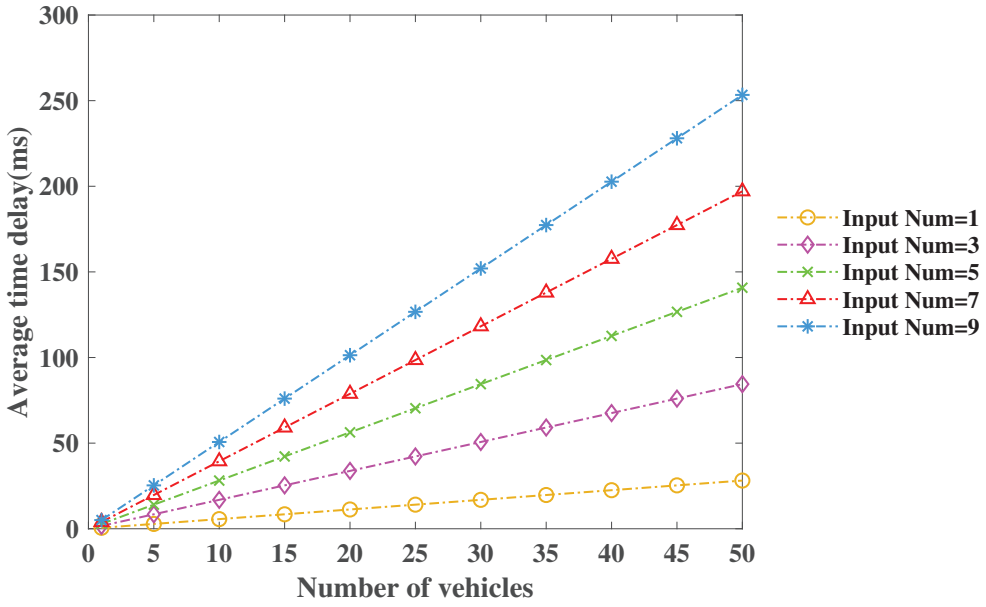

Figure 8. The average time overhead to implement the transaction confirmation process.

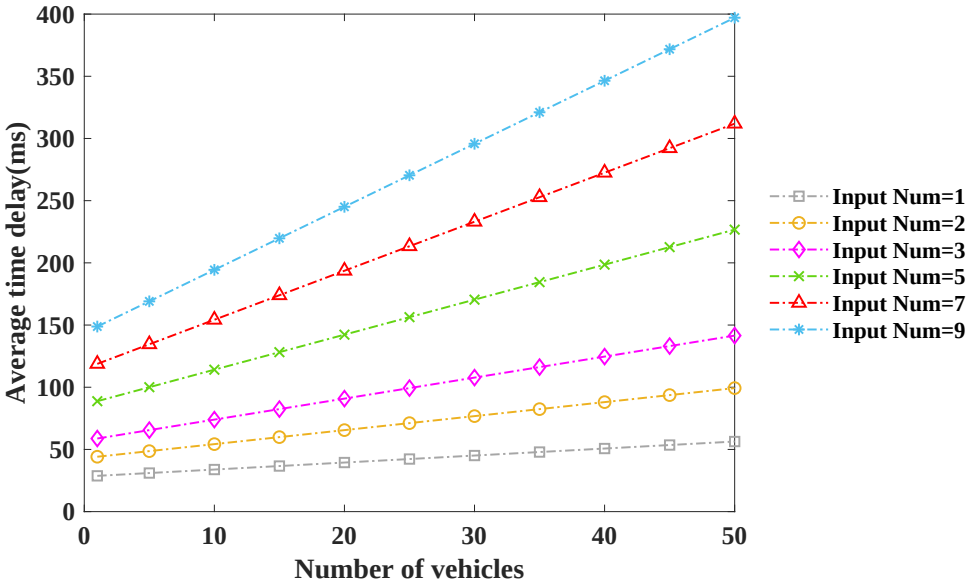

Figure 9. The time delay of authentication when the transaction has different $Input$ quantities.

$$T_{N-agg}^{Au} = T_{gen} + n(T_{mat}^{Au} + T_{ver}^{Au}) + \xi T_c \tag{13}$$

$$T_{agg}^{Au} = T_{gen} + (T_{mat}^{Au} + T_{ver}^{Au}) + T_c \tag{14}$$

It is worth noting that the communication overhead increases as the data package size increases. Thus, the proportional coefficient ξ is utilized to represent the increments. Moreover, T_c represents the time overhead of communication; T_{gen} represents the time overhead of transaction generation.

6.2.5. Computation Cost Analysis and Comparison

In this subsection, we evaluate the computation overhead of our proposal. We calculate the execution time of several basic cryptographic operations with the JPBC li-

brary [42]. Then, we calculate the details of the authentication overhead for several related schemes, presented in Table 4. Finally, we compare our scheme in batch authentication overhead with other related schemes. The details are as follows:

- T_p is the time required to perform a bilinear pairing. $T_p \approx 39.872$ ms.
- T_h is the time required to implement the hash function. $T_h \approx 0.002$ ms.
- T_m is the time required to implement one elliptic curve point multiplication operation. $T_m \approx 0.601$ ms.
- T_{ep1} is time required to implement one exponentiation in G_1. $T_{ep1} \approx 20.311$ ms.
- T_{ep2} is time required to implement one exponentiation in G_2. $T_{ep2} \approx 16.928$ ms.
- T_a is the time required to implement one elliptic curve point addition operation. $T_a \approx 0.051$ ms.

Table 4. Comparison for the computational costs.

Scheme	For One Entity Authentication	For n Entities Authentication
IBCPPA	$3T_p + 2T_{ep1} + 2T_h$	$(2+n)T_p + (2n)T_{ep1} + 2nT_h$
BPPA	$2T_m + T_a + 25T_h$	$2nT_m + nT_a + 25nT_h$
EAAP	$2T_p + 4T_{ep1} + T_{ep2}$	$(1+n)T_p + 4T_{ep1} + nT_{ep2}$
CPAV	$2T_p + 2T_{ep1} + T_h$	$(1+n)T_p + 2nT_{ep1} + nT_h$
EBAS (proposed)	$T_{mat} + T_h + T_{ver} + T_{ret}$	$nT_{mat} + nT_h + nT_{ver} + nT_{ret}$

For one single authentication, our scheme requires one hash operation, one key match operation, one signature verification operation, and one retrieval operation. According to the simulation, the average execution time of hash (T_h), retrieval (T_{ret}), and verification (T_{ver}) is 0.002 ms, 0.32 ms, and 0.62 ms, respectively. Thus, the total computation cost of EBAS is $T_{mat} + T_h + T_{ver} + T_{ret} \approx 0.942$ ms. It should be noted that the execution time of the match operation is negligible under the simulation.

In addition, we analyze the calculation details of the compared schemes, IBCPPA [43], BPPA [21], EAAP [44], and CPAV [45]. For example, the EAAP requires two bilinear pairing operations, four exponentiations in G_1 operations, and one exponentiation in G_2 hash operation. Thus, the total computational cost is $2T_p + 4T_{ep1} + T_{ep2} \approx 177.916$ ms. For batch authentication, the EAAP needs $(1+n)$ bilinear pairing operations, four exponentiations in G_1 operations, and n exponentiations in G_2 hash operations. Thus, the total computational cost is $(1+n)T_p + 4T_{ep1} + nT_{ep2}$.

From Table 4, it is seen that the computational cost of a single authentication in IBCPPA, EAAP, CPAV, BPPA, and the proposed EBAS scheme is 160.242 ms, 177.916 ms, 120.368 ms, 1.303 ms, and 0.942 ms, respectively. Thus, the proposed scheme is characterized by lower computational overhead than the compared schemes.

In addition, as shown in Figure 10, we compare our proposal in batch authentication with four existing schemes, IBCPPA, EAAP, CPAV, and BPPA. When there are 50 authentication requests, the time delay is 4104.644 ms, 3204.848 ms, 4064.672 ms, 65.15 ms, and 56.52 ms, respectively, for IBCPPA, EAAP, CPAV, BPPA, and the proposed scheme. Therefore, our scheme outperforms the other four schemes by at least $(65.15 - 56.52)/65.15 \approx 13.25\%$. Therefore, the proposed scheme is more efficient than the compared schemes when the traffic load increases.

6.2.6. Authentication Analysis and Comparison

We analyze the communication overhead of our scheme, compared to those of the IBCPPA, BPPA, EAAP, and CPAV schemes. The communication overhead refers to sending the information from a vehicle to an RTA. Table 5 lists the total communication overhead of all schemes in terms of sending out a single message and n messages.

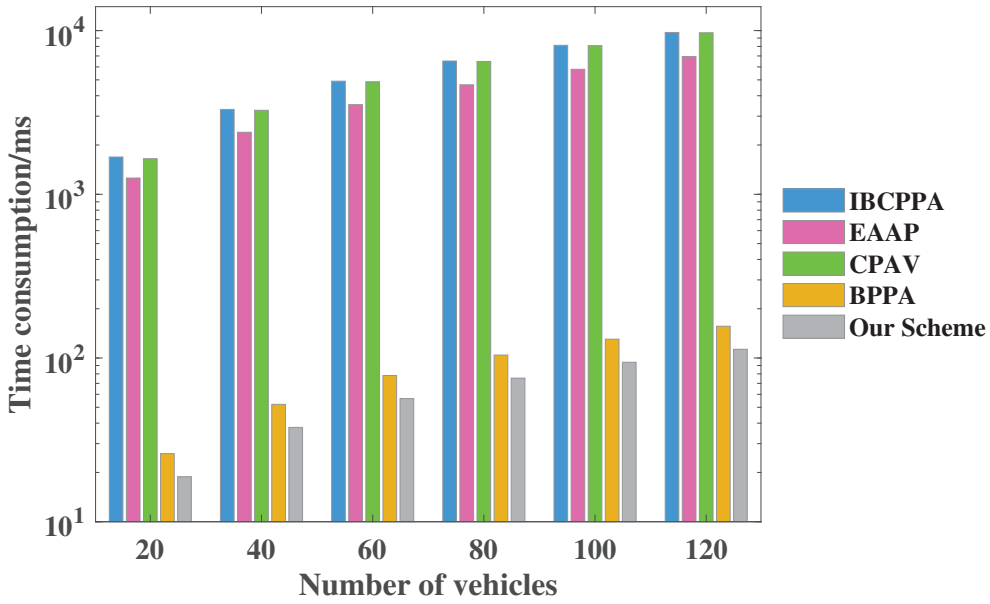

Figure 10. Comparison of computational cost between different schemes.

Table 5. Comparison for the communication overhead.

Scheme	For One Single Message	For n Messages
IBCPPA	833 bytes	833 n bytes
BPPA	552 bytes	552 n bytes
EAAP	220 bytes	220 n bytes
CPAV	800 bytes	800 n bytes
EBAS (proposed)	161 bytes	161 n bytes

Generally, the communication overhead is caused by the identity information, certificate, pseudo identity, and timestamp, etc. The total packet sizes of IBCPPA and BPPA are 833 bytes and 552 bytes, respectively. The packet size of the EAAP scheme is 220 bytes, which contains a 20-byte signature, a 20-byte public key, and a 180-byte certificate. The total packet size of CPAV is 800 bytes, which consists of a 544-byte certificate and 256 bytes for other authentication parameters. The transaction involved in our scheme requires 161 bytes. Based on the parameters in Tables 5 and 6, we use NS-2 to simulate the communication delay and compare the proposed scheme with other schemes in terms of communication overhead, as shown in Figure 11. From Table 5 and Figure 11, we can observe that our scheme has advantages over other schemes in terms of communication overhead.

Table 6. Parameters for communication simulation.

Parameter	Value
Simulation area	2500 × 16 m^2
No. of traffic lane	4
No. of RSUs	4
No. of RTAs	1
Maximum no. of vehicles	25
Simulation time	100 s
Communication protocol	802.11 p
Channel bandwidth	6 Mbps
Transmission range of OBU	300 m
Transmission range of RTA	1000 m
Minimum inter-vehicle distance	40 m
Route protocol	AODV

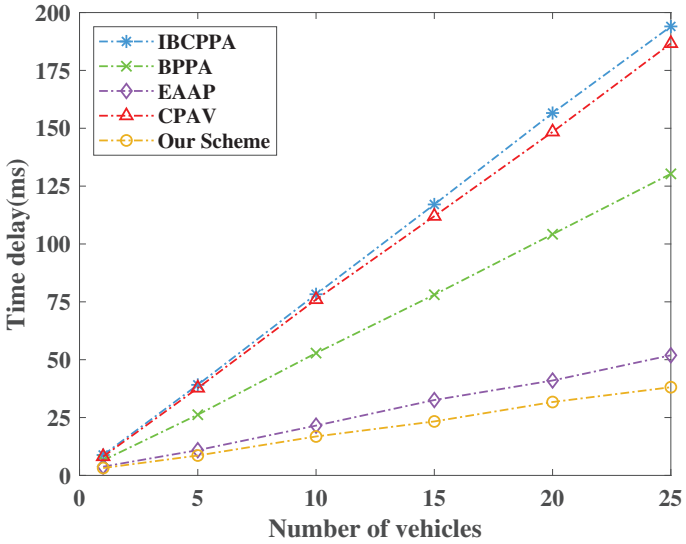

Figure 11. Comparison of communication delay between different schemes.

Finally, considering the communication delay and computational costs, we compare the proposed scheme with others in terms of authentication overhead. From Figure 12, we can observe that our scheme outperforms others in terms of authentication overhead.

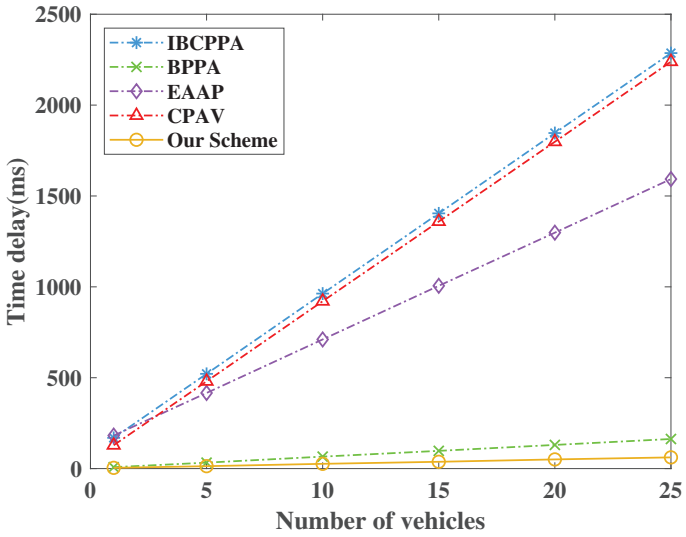

Figure 12. Comparison of authentication overhead between different schemes.

7. Conclusions

In this article, we propose an efficient blockchain-based authentication scheme for secure communication in a vehicular ad hoc network (EBAS). In our scheme, the entities can accomplish authentication with the assistance of transactions constructed based on the UTXO model. The verifier checks the validity of the single input contained in the uploaded transaction to verify the legitimacy of the message sender's identity. In terms of privacy preservation, based on the asymmetric key encryption technique, the transaction pseudonym is generated to assist users to participate in the authentication process anonymously. Moreover, our scheme guarantees the scalability of EBAS by proposing a transaction update mechanism, which can keep data storage and retrieval efficiency at a stable level rather than undergoing near-linear growth. Based on the security analysis, our scheme is more comprehensive in terms of privacy preservation and resisting common attacks in the VANET. Regarding the authentication scheme, the simulations show that the average computational cost of the proposed scheme is around 0.942 ms, which

outperforms the existing schemes. Furthermore, we implement a simulation experiment to evaluate the communication delay and authentication overhead. Thus, compared with other existing schemes, our proposal has advantages in communication delay and authentication overhead. The future work will focus on adding an incentive module, which is a significant step toward wider applications.

Author Contributions: Conceptualization, X.F. and K.C.; methodology, X.F. and K.C.; software, K.C.; validation, X.F., K.C., H.J. and Z.L.; formal analysis, X.F. and K.C.; investigation, X.F. and K.C.; data curation, K.C. and Z.L.; writing—original draft preparation, X.F. and K.C.; writing—review and editing, X.F. and K.C.; supervision, X.F.; project administration, X.F.; funding acquisition, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 61902157.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the anonymous referees for their valuable comments and constructive suggestions.

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

References

1. Tan, H.; Xuan, S.; Chung, I. Hcda: Efficient pairing-free homographic key management for dynamic cross-domain authentication in vanets. *Symmetry* **2020**, *12*, 1003. [CrossRef]
2. Al-shareeda, M.A.; Anbar, M.; Manickam, S.; Hasbullah, I.H. An efficient identity-based conditional privacy-preserving authentication scheme for secure communication in a vehicular ad hoc network. *Symmetry* **2020**, *12*, 1687. [CrossRef]
3. Ashraf, M.; Bilal, H.; Khan, I.A.; Ahmad, F. Vanet challenges of availability and scalability. *VFAST Trans. Softw. Eng.* **2016**, *4*, 46–53. [CrossRef]
4. Dötzer, F. Privacy issues in vehicular ad hoc networks. In *International Workshop on Privacy Enhancing Technologies*; Springer: Berlin/Heidelberg, Germany, 2005.
5. Hartenstein, H.; Laberteaux, K.P. A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.* **2008**, *46*, 164–171. [CrossRef]
6. Parno, B.; Perrig, A. Challenges in securing vehicular networks. In *Workshop on Hot Topics in Networks (HotNets-IV)*; MD, USA, 2005; pp. 1–6. Available online: https://www.semanticscholar.org/paper/Challenges-in-Securing-Vehicular-Networks-Parno-Perrig/d49b53b33590a4aafe5f5779c41ae40f50af0d6a (accessed on 6 June 2022).
7. Wang, Y.; Ding, Y.; Wu, Q.; Wei, Y.; Qin, B.; Wang, H. Privacy-preserving cloud-based road condition monitoring with source authentication in vanets. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1779–1790. [CrossRef]
8. Motlagh, N.H.; Taleb, T.; Arouk, O. Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. *IEEE Internet Things J.* **2016**, *3*, 899–922. [CrossRef]
9. Horng, S.-J.; Tzeng, S.-F.; Huang, P.-H.; Wang, X.; Li, T.; Khan, M.K. An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. *Inf. Sci.* **2015**, *317*, 48–66. [CrossRef]
10. Luo, B.; Li, X.; Weng, J.; Guo, J.; Ma, J. Blockchain enabled trust-based location privacy protection scheme in vanet. *IEEE Trans. Veh. Technol.* **2019**, *69*, 2034–2048. [CrossRef]
11. Yao, Y.; Chang, X.; Mišić, J.; Mišić, V.B.; Li, L. Bla: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services. *IEEE Internet Things J.* **2019**, *6*, 3775–3784. [CrossRef]
12. Chen, L.; Ng, S.-L.; Wang, G. Threshold anonymous announcement in vanets. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 605–615. [CrossRef]
13. Liu, Y.; Ling, J.; Wu, Q.; Qin, B. Scalable privacy-enhanced traffic monitoring in vehicular ad hoc networks. *Soft Comput.* **2016**, *20*, 3335–3346. [CrossRef]
14. Goumidi, H.; Harous, S.; Aliouat, Z.; Gueroui, A.M. Lightweight secure authentication and key distribution scheme for vehicular cloud computing. *Symmetry* **2021**, *13*, 484. [CrossRef]
15. Adams, C.; Lloyd, S. *Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations*; Sams Publishing: Carmel, IN, USA, 1999.
16. Wu, L.; Fan, J.; Xie, Y.; Wang, J.; Liu, Q. Efficient location-based conditional privacy-preserving authentication scheme for vehicle ad hoc networks. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147717700899. [CrossRef]

17. Zhang, C.; Lu, R.; Lin, X.; Ho, P.-H.; Shen, X. An efficient identity-based batch verification scheme for vehicular sensor networks. In Proceedings of the IEEE INFOCOM 2008—The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 246–250.
18. Salim, A.; Tripathi, S.; Tiwari, R.K. Applying Geo-Encryption and Attribute Based Encryption to Implement Secure Access Control in the Cloud. SSRN 3459330. 2019. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3459330 (accessed on 6 June 2022).
19. Zukarnain, Z.A.; Muneer, A.; Aziz, M.K.A. Authentication securing methods for mobile identity: Issues, solutions and challenges. *Symmetry* **2022**, *14*, 821. [CrossRef]
20. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decent. Bus. Rev.* **2008**, 21260. Available online: https://www.microstrategy.com/en/bitcoin/documents/bitcoin-a-peer-to-peer-electronic-cash-system (accessed on 6 June 2022).
21. Lu, Z.; Wang, Q.; Qu, G.; Zhang, H.; Liu, Z. A blockchain-based privacy-preserving authentication scheme for vanets. *IEEE Trans. Very Large Scale Integr. Syst.* **2019**, *27*, 2792–2801. [CrossRef]
22. Lei, A.; Ogah, C.; Asuquo, P.; Cruickshank, H.; Sun, Z. A secure key management scheme for heterogeneous secure vehicular communication systems. *ZTE Commun.* **2019**, *14*, 21–31.
23. Arora, A.; Yadav, S.K. Block chain based security mechanism for internet of vehicles (iov). In Proceedings of 3rd the International Conference on Internet of Things and Connected Technologies (ICIoTCT), Jaipur, India, 26–27 March 2018; pp. 26–27.
24. Wu, Q.; Domingo-Ferrer, J.; Gonzalez-Nicolas, U. Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications. *IEEE Trans. Veh. Technol.* **2010**, *59*, 559–573.
25. Lin, X. Lsr: Mitigating zero-day sybil vulnerability in privacy-preserving vehicular peer-to-peer networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 237–246. [CrossRef]
26. Miller, C.; Valasek, C. Remote Exploitation of an Unaltered Passenger Vehicle. Black Hat USA. 2015. Available online: https://dl.packetstormsecurity.net/papers/attack/Remote-Car-Hacking.pdf (accessed on 6 June 2022).
27. Engoulou, R.G.; Bellaieche, M.; Pierre, S.; Quintero, A. Vanet security surveys. *Comput. Commun.* **2014**, *44*, 1–13. [CrossRef]
28. Petit, J.; Shladover, S.E. Potential cyberattacks on automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 546–556. [CrossRef]
29. Jaballah, W.B.; Conti, M.; Mosbah, M.; Palazzi, C.E. Fast and secure multihop broadcast solutions for intervehicular communication. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 433–450. [CrossRef]
30. Zhang, L.; Wu, Q.; Qin, B.; Domingo-Ferrer, J. Appa: Aggregate privacy-preserving authentication in vehicular ad hoc networks. In Proceedings of the Information Security, 14th International Conference, ISC 2011, Xi'an, China, 26–29 October 2011.
31. Sucasas, V.; Mantas, G.; Saghezchi, F.B.; Radwan, A.; Rodriguez, J. An autonomous privacy-preserving authentication scheme for intelligent transportation systems. *Comput. Secur.* **2016**, *60*, 93–205. [CrossRef]
32. Jiang, Y.; Shi, M.; Shen, X.; Lin, C. Bat: A robust signature scheme for vehicular networks using binary authentication tree. *IEEE Trans. Wirel. Commun.* **2008**, *8*, 1974–1983. [CrossRef]
33. Ying, B.; Nayak, A. Anonymous and lightweight authentication for secure vehicular networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10626–10636. [CrossRef]
34. Simmons, G.J. Symmetric and asymmetric encryption. *ACM Comput. Surv.* **1997**, *11*, 305–330. [CrossRef]
35. Hasan, A.; Sabah, S.; Haque, R.U.; Daria, A.; Rasool, A.; Jiang, Q. Towards convergence of iot and blockchain for secure supply chain transaction. *Symmetry* **2022**, *14*, 64. [CrossRef]
36. Liu, Y.; Wang, L.; Chen, H.-H. Message authentication using proxy vehicles in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2014**, *64*, 3697–3710. [CrossRef]
37. Lu, Z.; Wang, Q.; Qu, G.; Liu, Z. Bars: A blockchain-based anonymous reputation system for trust management in vanets. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 98–103.
38. Chuang, Y.-H.; Lei, C.-L.; Shiu, H., Jr. How to design a secure anonymous authentication and key agreement protocol for multi-server environments and prove its security. *Symmetry* **2021**, *13*, 1629. [CrossRef]
39. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ecdsa). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
40. Liu, Y.; Guo, W.; Fan, C.-I.; Chang, L.; Cheng, C. A practical privacy-preserving data aggregation (3pda) scheme for smart grid. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1767–1774. [CrossRef]
41. Gorenflo, C.; Lee, S.; Golab, L.; Keshav, S. Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second. *Int. J. Netw. Manag.* **2020**, *30*, e2099. [CrossRef]
42. Caro, A.D.; Iovino, V. jpbc: Java pairing based cryptography. In Proceedings of the 2011 IEEE Symposium on Computers and Communications (ISCC), Kerkyra, Greece, 28 June 2011–1 July 2011; pp. 850–855.
43. Shao, J.; Lin, X.; Lu, R.; Zuo, C. A threshold anonymous authentication protocol for vanets. *IEEE Trans. Veh. Technol.* **2015**, *65*, 1711–1720. [CrossRef]
44. Azees, M.; Vijayakumar, P.; Deboarh, L.J. Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2467–2476. [CrossRef]
45. Vijayakumar, P.; Chang, V.; Deborah, L.J.; Balusamy, B.; Shynu, P. Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks. *Future Gener. Comput. Syst.* **2018**, *78*, 943–955. [CrossRef]