Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6679654, 14 pages
https://doi.org/10.1155/2021/6679654

Research Article

An Atomic Cross-Chain Swap-Based Management System in Vehicular Ad Hoc Networks

Chenkai Tan[1], **Shaoyi Bei**[1], **Zhengjun Jing**[1,2] and **Neal Xiong**[3]

[1]*School of Vehicle and Traffic Engineering, Jiangsu University of Technology, Changzhou 213001, China*
[2]*School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China*
[3]*Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA*

Correspondence should be addressed to Zhengjun Jing; zhengjun_jing@163.com

Received 20 October 2020; Revised 9 December 2020; Accepted 9 January 2021; Published 29 January 2021

Academic Editor: Hongju Cheng

Copyright © 2021 Chenkai Tan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The blockchain-based management system has been regarded as a novel way to improve the efficiency and safety of Vehicular Ad Hoc Networks (VANETs). A blockchain-based scheme's performance depends on blockchain nodes' computing power composed from the road-side unit (RSU). However, the throughput of blockchain-based application in VANETs is limited by the network bandwidth. A single blockchain cannot record large-scale VANETs' data. In this paper, we design an atomic cross-chain swap-based management system (ACSMS) to boost the scalability of blockchain-based application in VANETs. The blockchain-based public-key encryption with keyword search is further introduced to protect user privacy. The analysis shows that ACSMS achieves cross-chain swap without loss of CAV security privacy. The simulation results show that our method can realize multiple blockchain-based applications in VANETs.

1. Introduction

Nowadays, connected autonomous vehicles (CAVs) equip with the on-board unit (OBU) that enables vehicle ad hoc networks (VANETs [1]) to be formed. By C-V2X and DSRC communication, VANETs could improve traffic efficiency [2] and road safety [3]. With the VANET development [4], CAVs not only need safely and efficiently communicate to other CAVs [5–7] but also need to decide the trustworthiness of other CAVs' messages. There are many attacks and challenges [8–11] in VANETs, which may be solved by blockchain technology. For example, blockchain-based VANET schemes have been designed to provide security key transfers [12] and decentralized trust management [13].

Blockchain is a novel tamper-resistance and decentralization ledger-based storage method, which is useful in VANETs. However, the throughput of blockchain is limited by the blockchain consensus algorithms and network bandwidth. For example, the throughput on Ethereum [14] is 10-30 TPS. Simultaneously, the number of CAVs around the world will reach 2 billion in the next 10-20 years [15]. Therefore, Shrestha et al. [16] proposed that administrator divide multiple blockchains according to geographical location to enhance application scalability.

There are three categories of existing blockchain-based VANET schemes: the blockchain-based PEKS [17], the blockchain-based dynamic key management system [12] (BKM), and the blockchain-based trust management system [13] (BTM). First, the blockchain-based PEKS integrates blockchain technology and PEKS. The blockchain technology supports the data tamper resistance, and PEKS supports verifiable searching simultaneously. Second, BKM could prevent an unregistered user from accessing VANETs illegally. However, BKM uses pseudo-ID to achieve conditional privacy. Without revealing CAVs' identities, the user will lack the motivation to deliver information. Therefore, BTM was proposed to build trust in the communications of CAVs and incentive users to forward announcements.

Toward enhancing the flexibility of the blockchain-based scheme in VANETs, we integrate multiple blockchain-based

applications into the single blockchain. Then, we use the atomic cross-chain protocol to connect blockchain. The main contributions of this article are summarized as follows:

(i) In VANETs, network size can be geographically unbounded and very scalable, but nodes in VANETs have geographical relevancy. We partition country maps into regions, and each region maintains an independent blockchain, which will reduce blockchain scalability issues. We develop ACSMS to connect multiple blockchain systems while maintaining the property of trust and integrity built by individual blockchains. Therefore, the ACSMS can enhance interoperability between chains

(ii) We also introduce the blockchain-based PEKS [18] as the data-sharing scheme. In blockchain-based PEKS, CAVs use the public key to encrypt the data and the corresponding keywords. The ciphertext signature is provided in the blockchain, and the ciphertexts are stored in the TA server. Thus, the blockchain-based PEKS support verifies of ciphertexts

(iii) Furthermore, we attempt to integrate the blockchain-based PEKS, BKM, and BTM into ACSMS, such that CAVs could use multiple blockchain-based applications with a single blockchain account. Compared with other reference schemes, ACSMS achieves more functionality

The remainder of this paper is organized as follows. The related works are briefly introduced in Section 2. In Section 3, we give the model overview and details of ACSMS. Then, the security analysis of our scheme is presented in Section 4. In Section 5, we make a performance evaluation of ACSMS. In Section 6, we give the conclusions of the paper.

2. Related Work

2.1. Blockchain-Based PEKS. Since Boneth et al. [19] first posed PEKS, numerous schemes about PEKS have been put forward. There are four polynomial-time algorithms of PEKS: KeyGen, Trapdoor, PEKS, and Test. If the user sends a trapdoor to the server, the server can locate all ciphertexts containing keywords through PEKS. Significantly, blockchain-based PEKS for VANETs [17] allows users to detect any unauthorized manipulation. The blockchain-based PEKS schemes [17, 18, 20] avoid the centralized malicious attacks [21]. The oblivious protocol between users and blockchain servers can thwart off-line KGA (keyword guessing attacks).

2.2. BKM. Lei et al. [12] proposed a novel key management scheme based on blockchain to transfer key within the decentralized VANETs securely. Similarly, Ma et al. [22] developed an efficient BKM scheme that supports key updates and revocation. Furthermore, a lightweight mutual authentication protocol for BKM is proposed in [22]. Shrestha et al. [16] also employed blockchain technology to disseminate information in VANETs. Unlike Lei et al. and Ma et al., Shrestha et al. divided the map into blockchain regions according to the density of CAVs. However, Shrestha et al. did not use cross-chain technology to connect the different regions.

As shown in Figure 1, the basic structure of the blockchain [23] consists of a series of various blocks that has a block header and a block body. The block header is used to identify a particular block, and a block header consists of blockchain version, the previous block's hash (PreHash), MerkleRoot, Timestamp (TS), nonce, and difficulty target. The block body is used to record transactions, and a block body consists of transactions and a Merkle hash tree formed by the hash value of transactions. Each block has a difficulty target related to the previous block; a chain structure is formed. If a miner first finds the nonce and broadcast correct block into the network, this block's validity will be acknowledged by other blockchain nodes.

2.3. BTM. The decentralized BTM schemes are introduced to determine the trustworthiness of CAVs. Yang et al. [13] proposed a BTM to query neighbours' trust values and then verify the received messages. Similarly, Lu et al. [24] investigated an anonymous BTM to break the linkability between real identities and the public keys. Furthermore, Lu et al. used the TA and RSU to provide BTM. Zhao et al. [25] employed blockchain technology to associate resource allocation with the trust value of CAVs. In addition, Zhao et al. introduced a novel BTM based on PBFT&PoS consensus algorithm to support cross-chain swap. But PBFT&PoS-based BTM cannot support PoW-based scheme.

3. An Atomic Cross-Chain Swap-Based Management System

3.1. System Model. As illustrated in Figure 2, three types of entities in VANETs are CAVs, RSU, and TA.

The connected autonomous vehicles (*CAVs*): each CAV, as a sender *S*, is equipped with a hardware security module (HSM) and an OBU. The feature of CAVs includes dynamic mobility, wireless communication [26], and predictable trajectory [27] [22, 28]

The roadside unit (*RSU*): the RSUs provide V2I provide V2I services by communicating with CAVs. RSU as blockchain nodes $\{KS_1, KS_2, ..., KS_n\}$ creates new blocks to maintain the blockchain in the region

The trust authority (*TA*): as a storage server *C*, the main task of TA includes the identity review of CAV users, deployment of smart contracts, and the issuance of transaction data. The TA also sends the pseudo-IDs to the CAVs. After the initial registration, TA provides every CAVs with a unique blockchain account to realize ACSMS

With different transfer protocols, CAVs can reach V2X through OBU. The concept design of the OBU [29] is illustrated in Figure 3. The OBU provides with long-range wide-area network (LoRaWAN), narrow-band internet of things (NB-IoT), and global positioning system (GPS). By LoRaWAN and NB-IoT, CAVs communicate with the TA (V2C), RSU (V2I), and other CAVs (V2V).

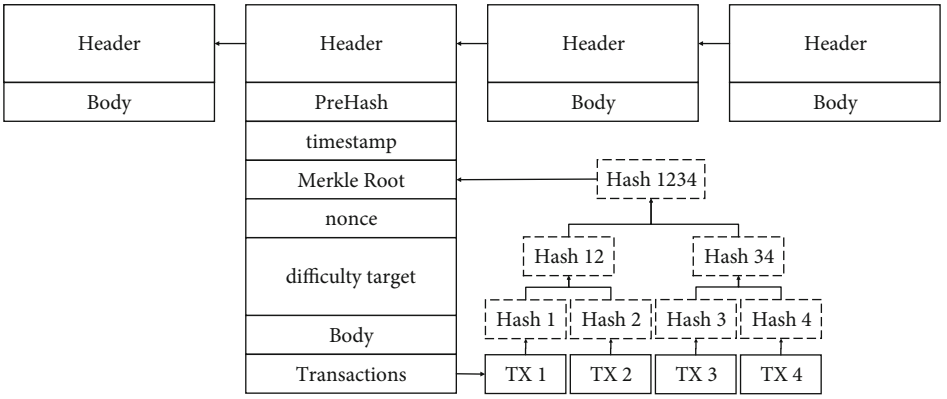

Figure 1: The basic blockchain structure.

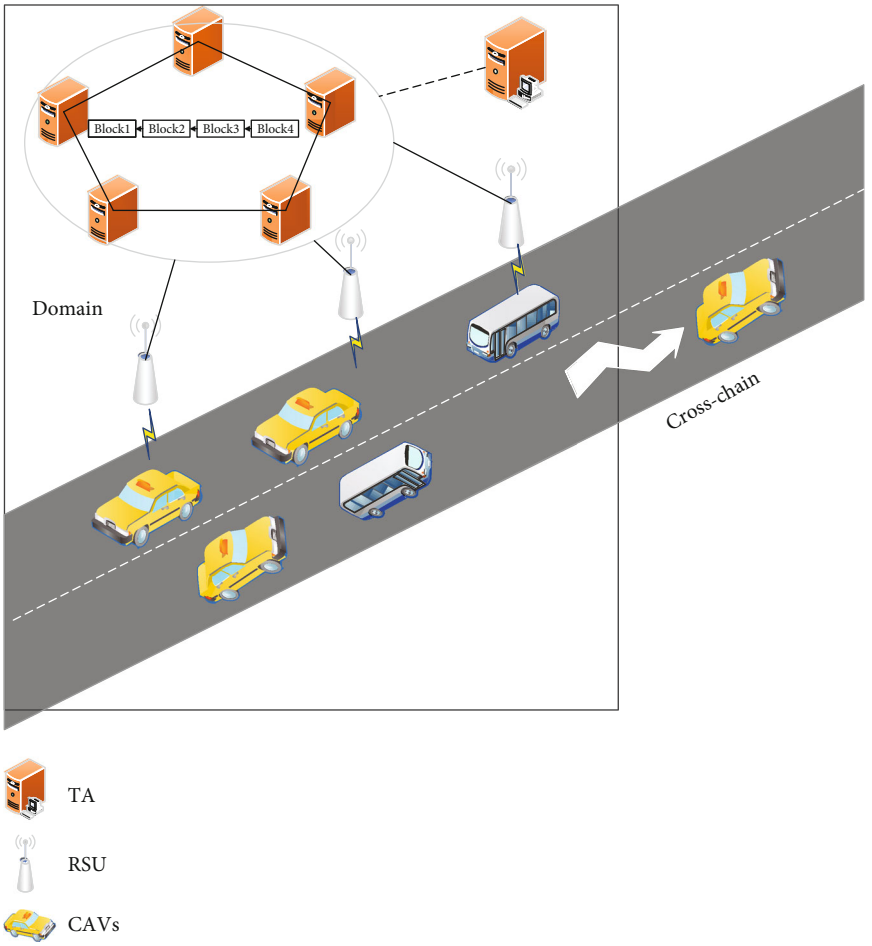

Figure 2: System model of ACSMS.

In the ACSMS, there are four primary functions: blockchain-based PEKS, BKM, BTM, and atomic-cross chain-swap. We will, respectively, present the four functions.

(a) *The blockchain-based public-key encryption with keyword search* (*PEKS*). CAVs outsource ciphertext and cypher-text signature to the TA. Then, the TA broadcasts the signature and the serial number of ciphertext into the blockchain. Finally, by TEST algorithm and the signature, the searcher can obtain correct ciphertext from the TA

(b) *The blockchain-based key management* (*BKM*). The TA registers CAVs and sends pseudo-ID to CAVs. By blockchain-based PEKS, CAVs' details (such as licence plate and driver's licence) are encrypted to protect privacy. By pseudo-ID, CAVs enable explicit mutual authentication with RSU through a handover

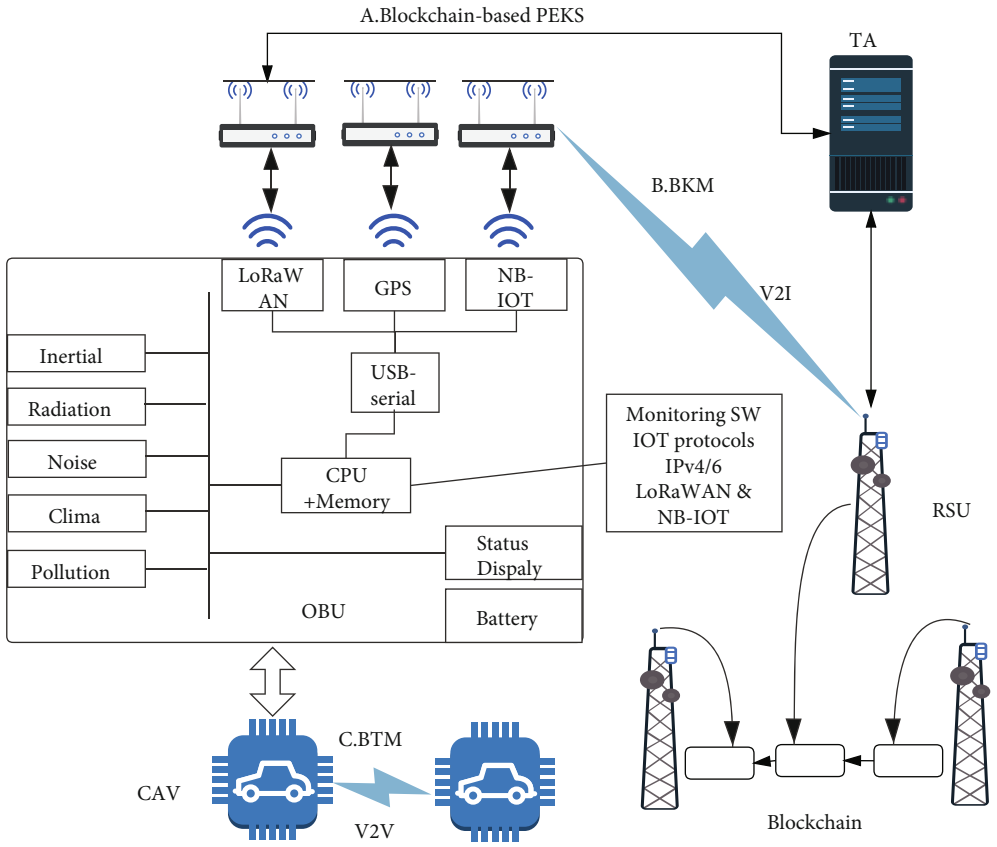

Figure 3: The structure of ACSMS.

authentication protocol [30]. Furthermore, blockchain technology provides decentralized authentication for this protocol

(c) *The blockchain-based trust management* (*BTM*). After the BKM, CAV, and RSU establish a shared symmetric key for the subsequent communication session. Then, RSU gets CAV's trust value from the blockchain. To share trust value with pseudo-ID, RSU generates an authentication code of CAVs' trust value. Other CAVs can then gain the credibility of CAVs' trust value without knowing the CAVs' real identity

(d) *The atomic-cross chain-swap*. We use an atomic-cross chain-swap algorithm [31, 32] that provides cross-chain interoperability among blockchain-based systems. Only serial number of ciphertext and CAVs' trust value are swapped across different blockchains

3.2. The Blockchain-Based PEKS. Based on the project of PEKS from bilinear pairings [17, 20, 33], we design a blockchain-based PEKS scheme for ACSMS. In the blockchain-based PEKS, the CAV (S) has a blockchain account b_{s_1} and send message (M) to the TA (C). Furthermore, the public key and private key of b_{s_1} are $(\mathrm{PK}_{b_s}, \mathrm{SK}_{b_s})$.

It should be emphasized that the PEKS construction in this work is very basic and can be extended to more complex PEKS through integrating with other technology. For example, by introducing cloud storage services [18], the CAV is able to outsource the data to the cloud storage services.

3.2.1. Setup Stage. With a security parameter λ, the system parameters $\{q, \hat{e}, \mathbb{G}_1, \mathbb{G}_2, H_1, H_2, H_3, H_4, \mu,\}$ are generated. Here, $(\mathbb{G}_1,+)$ and $(\mathbb{G}_2,\cdot)$ are two cyclic groups of prime order q. $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ be an admissible bilinear map, which satisfies bilinear, nondegenerate, and computable [34].

H is a cryptographic hash function, $H_1{:}\{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, $\mathbb{Z}_q^* = \{\rho \mid 1 \le \rho \le q\}$. $H_2{:}\mathbb{G}_2 \longrightarrow \{0,1\}^{\log q}$. $H_3{:}\{0,1\}^* \longrightarrow \{0,1\}^h$, where h is the output length of H_3. $H_4{:}\{0,1\}^* \longrightarrow \mathbb{G}_1$. $\mu = \hat{e}(P,P)$.

3.2.2. Blockchain Stage

(i) *KeyGen*. The CAV (S) picks a random secret key S $\mathrm{k}_s = x \in \mathbb{Z}_q^*$, and corresponding public key is $\mathrm{Pk}_s = X = xP \in \mathbb{G}_1$. With same parameters, the TA (C) generates the key as $(\mathrm{Pk}_c = Y = yP \in \mathbb{G}_1, \mathrm{Sk}_c = y \in \mathbb{Z}_q^*)$.

(ii) *Trapdoor*. Take as input Sk_s and keyword w, the CAV computes the Trapdoor [33] $T_w = (H_1(w) +$

$x)^{-1}P$. Then, the CAV encrypts M through a standard public-key encryption $E(\cdot)$ [35].

(iii) *PEKS*. Take as input Pk_s, Pk_c, and w, the CAV randomly selects $r_1, r_2 \in \mathbb{Z}_q^*$. The CAV computes $A = r_1 H_1(w)P + r_1 X$, $B = r_2 P$, $V = H_2(\hat{e}(r_1 P + r_2 A, Y))$, and output $\mathrm{PEKS}(w, \mathrm{PK}_s) = (A, B, V)$ [33].

With the encrypted M and keywords $\{w_1, w_2, \cdots, w_n\}$, a list of ciphertext is got by CAV, which is $\mathrm{Ct}_s = E(M)\| \mathrm{PEKS}(w_1, \mathrm{PK}_s)\|\cdots\|\mathrm{PEKS}(w_n, \mathrm{PK}_s)$, where $\|$ indicates message concatenation operation. Then, the CAV computes the signature of Ct_s as [34] $\mathrm{Sig} = P/(H_3(Ct_s) + x)$. Finally, the CAV sends ciphertext Ct_s and Sig to the TA.

To fast search Sig, the TA generates serial number Sn_i from b_{s_1} record Sn_{i-1}, and $\mathrm{Sn}_i = \mathrm{Sn}_{i-1} + 1$. The TA broadcasts a transaction $(\mathrm{Sig}, \mathrm{Sn}_i)$ into blockchain nodes (RSU) to check data integrity and authenticate user, which the receive account is b_{s_1}. Then, RSU KS_t will broadcast $(\mathrm{Sig}, \mathrm{Sn}_i)$ into the blockchain network, when a blockchain node KS_t(RSU) finds a correct nonce.

(iv) *TEST*. Take as input Pk_s, y, (A, B, V), and T_w, test $H_2(\hat{e}(yA, T_w + B)) \stackrel{?}{=} V$, where $\stackrel{?}{=}$ operation returns true if two values are equal and false if they are not equal. With keywords $\{w_1, w_2, \cdots, w_n\}$ and TEST, the TA obtains ciphertexts of each keyword. Then, check $\hat{e}(H_3(Ct_s)P + X, \mathrm{Sig}) \stackrel{?}{=} \hat{e}(P, P)$ through blockchain. The Sig proves that the ciphertexts belong to the blockchain account b_{s_1}, when $\stackrel{?}{=}$ operation returns true

By sending trapdoors and blockchain signature to the TA, the searcher can obtain ciphertexts as follows. The CAV (S) send $(\mathrm{Pk}_s, T_w, \mathrm{Sig}_{\mathrm{searcher}}, ts, \mathrm{PK}_{b_s})$ to the searcher, which $\mathrm{Sig}_{\mathrm{searcher}} = P/(H_3(\mathrm{Pk}_s\|T_w\|ts) + \mathrm{SK}_{b_s})$. With Pk_s, y, (A, B, V), and T_w, the TA can find ciphertexts from $(\mathrm{Pk}_s, T_w, \mathrm{Sig}_{\mathrm{searcher}}, ts, \mathrm{PK}_{b_s})$. If $\hat{e}(H_3(\mathrm{Ct}_s)P + X, \mathrm{Sig}) = \hat{e}(P, P)$, the ciphertexts belong to the blockchain account b_s. If $\hat{e}(H_3(\mathrm{Pk}_s\|T_w\|ts)P + X, \mathrm{Sig}_{\mathrm{searcher}}) = \hat{e}(P, P)$, the legal searcher is consented with the CAV (S).

The correctness of the scheme is proofed as follows:

$$
\begin{aligned}
&H_2(\hat{e}(yA, T_w + B))\\
&\quad = H_2\left(\hat{e}\left[yA, (H_1(w) + x)^{-1}P + r_2 P\right]\right)\\
&\quad = H_2\left(\hat{e}\left\{yr_1 P(H_1(w) + x), P\left[H_1(w) + x\right)^{-1} + r_2\right]\right\}\Big)\\
&\quad = H_2(\hat{e}\{r_1 Y, P[1 + r_2(H_1(w) + x)]\})\\
&\quad = H_2(\hat{e}\{Y, r_1 P + r_2[r_1 P H_1(w) + r_1 X]\})\\
&\quad = H_2(\hat{e}(Y, r_1 P + r_2 A)) = H_2(\hat{e}(r_1 P + r_2 A, Y)) = V,
\end{aligned}
$$

$$
\begin{aligned}
&\hat{e}(H_3(Ct_s)P + X, Sig)\\
&\quad = \hat{e}\left\{P[H_3(Ct_s) + x], \frac{1}{H_3(Ct_s) + x}P\right\}\\
&\quad = \hat{e}(P, P)^{(H_3(Ct_s)+x)(1/(H_3(\mathrm{Ct}_s)+x))} = \hat{e}(P, P).
\end{aligned}
\tag{1}
$$

3.3. Blockchain-Based Key Management in VANETs. The design goals of the BKM are security and efficiency user authentication. The VANET performs the BKM inside the blockchain region that can decentralized record user public key [12, 22]. However, BKM [12, 22] is hard to realize key revocation and hard to support cross-chain verify. In this paper, we take the PKI-based scheme [30] to realize key agreement, public update, and revocation. The ACSMS setups stage corresponding to the BKM system setup. The ACSMS blockchain stage includes the BKM user registration, CAV authentication, and the TA management CAVs' pseudo-ID. In contrast to [30], the blockchain is used to authentication users' pseudo-ID.

3.3.1. System Setup Phase. In our system, all nodes keep loose time synchronization. The system setup phase is as follows: First, the RSU has to install a blockchain-based application. Second, based on the density of CAVs in a country, maps are partitioned into different regions. Third, TA sends a unique blockchain account to every RSU. RSU then creates new blocks to maintain the blockchain in the region. Finally, administrators apply ACSMS to these blockchains.

3.3.2. Registration Phase. To ensure legitimacy within VANET, TA must register new CAVs [22, 36] through relevant CAV information S_{info}. S_{info} is formed by phone number, licence plate, driver's ID, etc. Different from the blockchain-based PEKS, the TA will sign the S_{info}. Take as input S_{info}, TA private key y, TA computes the signature of S_{info}, $\mathrm{Sig}_{S_{\mathrm{info}}} = P/(H_3(S_{\mathrm{info}}) + y)$. By the blockchain-based PEKS, CAVs get ciphertext ($\mathrm{Ct}_{\mathrm{info}}$) of S_{info}. Then, the CAV (S) sends the ciphertext $\mathrm{Ct}_{\mathrm{info}}$ to the TA (C), and the TA generates serial number Sn_i.

The blockchain transaction of $(\mathrm{Sig}_{S_{\mathrm{info}}}, \mathrm{Sn}_i)$ is processed as follows. The TA encapsulates $(\mathrm{Sig}_{S_{\mathrm{info}}}, \mathrm{Sn}_i)$ into JSON format. Next, TA sends the transaction to the RSU, and the RSU executes the mining process; the transaction $(\mathrm{Sig}_{S_{\mathrm{info}}}, \mathrm{Sn}_i)$ is added to the block. If a miner KS_t finds a nonce, KS_t broadcasts the block into the network. The RSU gets the transaction $(\mathrm{Sig}_{S_{\mathrm{info}}}, \mathrm{Sn}_i)$ through update the blockchain.

After the TA registers CAVs, the TA chooses a set of unlinkable pseudo-IDs $\mathrm{PID} = \{\mathrm{Pid}_1, \mathrm{Pid}_2, \cdots\}$ for each CAVs. For each $\mathrm{Pid}_j \in \mathrm{PID}$, TA computes the private key s $H_4(\mathrm{Pid}_j)$, s is a random number $\in \mathbb{Z}_q^*$. Then, TA sends all tuples $(\mathrm{Pid}_j, sH_4(\mathrm{Pid}_j))$ back to the CAV (S) using a secure transmission protocol. By changing its pseudo-ID, CAV achieves conditional location privacy and identity privacy in the VANETs.

3.3.3. Authentication Phase. The authentication based on bilinear pairing [30] could ensure the security of ACSMS. We use an handover authentication protocol to ensure efficiency seamless handover over multiple access points [30].

As Figure 4 shows, when an RSU (KS_t) is within the CAV direct communication range, the protocol run of BKM is as follows:

(1) The CAV picks an unused pseudo-ID $(\mathrm{Pid}_i, sH_4(\mathrm{Pid}_i))$

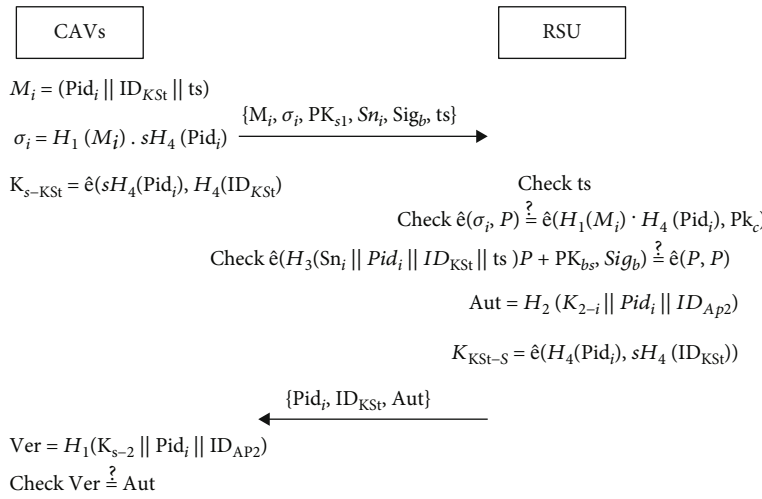

FIGURE 4: The protocol run of BKM.

(2) With message $M_i = (\mathrm{Pid}_i \| \mathrm{ID}_{\mathrm{KS}_t} \| \mathrm{ts})$ and $sH_4(\mathrm{Pid}_i)$, the CAV computes the signature $\sigma_i = H_1(\mathrm{M}_i) \cdot sH_4(\mathrm{Pid}_i)$ [30], where $\mathrm{ID}_{\mathrm{KS}_t}$ is RSU' (KS_t) identity, ts is a timestamp, and ‖ indicates the message concatenation operation

(3) The CAV computes the signature of $(\mathrm{Sig}_{S_{\mathrm{info}}}, \mathrm{Sn}_i)$ as $\mathrm{Sig}_b = P/H_3(\mathrm{Sn}_i \| \mathrm{Pid}_i \| \mathrm{ID}_{\mathrm{KS}_t} \| \mathrm{ts}) + \mathrm{SK}_{b_s}$, where SK_{b_s} is the private key of b_{s_1}

(4) The CAV sends the access request message $\{\mathrm{M}_i, \sigma_i, \mathrm{PK}_{b_s}, \mathrm{Sn}_i, \mathrm{Sig}_b, \mathrm{ts}\}$ to RSU (KS_t) through unicast, where PK_{b_s} is the public key of b_{s_1}

(5) The CAV computes the shared symmetric key with RSU (KS_t) as $K_{\mathrm{S-KS}_t} = \hat{e}(sH_4(\mathrm{Pid}_i), H_4(\mathrm{ID}_{\mathrm{KS}_t}))$ [30]

Upon receipt of $\{\mathrm{M}_i, \sigma_i, \mathrm{PK}_{b_s}, \mathrm{Sn}_i, \mathrm{Sig}_b\}$, the RSU KS_t proceeds as follows:

(1) RSU first checks ts to prevent replay attack. Then, RSU checks signature σ_i as $\hat{e}(\sigma_i, P) = \hat{e}(H_1(\mathrm{M}_i) \cdot sH_4(\mathrm{Pid}_i), P) = ^? \hat{e}(H_1(\mathrm{M}_i) \cdot H_4(\mathrm{Pid}_i), \mathrm{Pk}_c)$ [30]. Then, RSU needs to retrieve the corresponding $\{b_{s_1}, \mathrm{Sn}_i\}$ from the blockchain to ensure user legitimacy within pseudo-ID. Finally, RSU checks $\hat{e}(H_3(\mathrm{Sn}_i \| \mathrm{Pid}_i \| \mathrm{ID}_{\mathrm{KS}_t} \| \mathrm{ts})P + \mathrm{PK}_{b_s}, \mathrm{Sig}_b) = ^? \hat{e}(P, P)$. If both formula is correct, the CAV (S) is a legal vehicle, which was signed by the TA

(2) RSU further computes the shared symmetric key K_{KS_t-S}: $K_{\mathrm{KS}_t-S} = \hat{e}(H_4(\mathrm{Pid}_i), sH_4(\mathrm{ID}_{\mathrm{KS}_t})) = \hat{e}(sH_4(\mathrm{Pid}_i), H_4(\mathrm{ID}_{\mathrm{KS}_t})) = K_{\mathrm{S-KS}_t}$ [30]

(3) Then, KS_t generates an authentication code $\mathrm{Aut} = H_1(K_{\mathrm{KS}_t-S} \| \mathrm{Pid}_i \| \mathrm{ID}_{\mathrm{KS}_t})$ [30] and sends $\{\mathrm{Pid}_i, \mathrm{ID}_{\mathrm{KS}_t}, \mathrm{Aut}\}$ to the CAV (S)

Upon receipt of $\{\mathrm{Pid}_i, \mathrm{ID}_{\mathrm{KS}_t}, \mathrm{Aut}\}$, the CAV computes the verification code $\mathrm{Ver} = H_1(K_{\mathrm{KS}_t-S} \| \mathrm{Pid}_i \| \mathrm{ID}_{\mathrm{KS}_t}) = ^? \mathrm{Aut}$ [30]. If Ver matches Aut, KS_t is a legitimate RSU. The shared key between the CAV and RSU is K_{KS_t-S}.

By this protocol, CAV realizes authentication with RSU. The blockchain technology provides decentralized key authentication for the protocol. Between RSU and CAVs, a shared symmetric key K_{KS_t-S} is also established for the subsequent communication session.

If the administrator wants to find S_{info} of $\{\mathrm{M}_i, \sigma_i, \mathrm{PK}_{s_1}, \mathrm{Sn}_i, \mathrm{Sig}_b\}$, the TA first obtains T_w and Pk_s from the CAV. The TA takes as input Pk_s, y, and T_w and tests if $H_2(\hat{e}(yA, T_w + B)) = V$. With keywords $\{w_1, w_2, \cdots, w_n\}$ and TEST, the TA obtains ciphertext $\mathrm{Ct}_{\mathrm{info}}$ of each keyword. Then, the TA sends ciphertext $\mathrm{Ct}_{\mathrm{info}}$ to the CAV and CAV decryption $\mathrm{Ct}_{\mathrm{info}}$ as S_{info}'. After this, the CAV sends S_{info}' to the TA. Then, TA checks if $\hat{e}(H_3(S_{\mathrm{info}}')P + Y, \mathrm{Sig}_{S_{\mathrm{info}}}) = \hat{e}(P, P)$, where $\mathrm{Sig}_{S_{\mathrm{info}}}$ is obtained from the account b_{s_1}. If so, the correct S_{info} belongs to account b_{s_1}.

3.4. Blockchain-Based Trust Management in VANETs. The BKM uses pseudo-ID to provide conditional privacy. However, VANET is difficult to forward reliable message with an anonymous environment. Besides, the user with pseudo-ID will lack the motivation to forward announcements [37]. The BTM [13, 24, 25] was proposed to resolve these issues, which can recognize and eliminate of misbehaving vehicles. The process of the BTM is as follows. First, the CAV validates the legitimacy of received messages from neighboring CAVs. Second, based on the validated result, the rating of messages is generated by CAVs. Third, with the ratings uploaded from CAVs, the RSU gets the offsets of involved CAVs. Finally, the RSU adds the offset into the blockchain, where the offset is reflected by the CAV's blockchain account balance.

To calculate trust value, Zhao et al. [25] classify CAVs into three roles: privilege vehicles (S_1), enterprise vehicles

(S_2), and individual vehicles (S_3). Each role has different initial credibility. To share trust value with pseudo-ID, KS_t generates an authentication code of CAV's trust value OC_P as $Sig_{OC_p} = P/(H_3(Pid_i \| OC_P \| ts_1) + SK_{KS_t})$. SK_{KS_t} is the private key of the RSU (KS_t). Upon receipt of {Message, Pid$_i$, ts$_1$, Sig_{OC_p}, OC_P}, other CAV checks $\hat{e}(H_3(Pid_i \| OC_P \| ts_1)P + PK_{KS_t}, Sig_{OC_p}) = ^{?} \hat{e}(P, P)$, where PK_{KS_t} is RSU's (KS_t) public key.

A CAV receive M kinds of road-relevant events from different CAVs in a certain time. There are J CAVs in the reference set. The CAVs separate received messages into M groups as $\{g_1, g_2, \cdots, g_M\}$. g_i is one road-relevant event, which contains messages $\{W_i^1, W_i^2, \cdots, W_i^J\}$ that send by CAVs in the reference set J. According to [25], the tuple of one specific message (W_k^j [25]) is defined as

$$W_k^j = \left(d_k^j, \text{time}_k^j, OC_P\right), P \in (S_1, S_2, S_3). \tag{2}$$

The CAV receiver obtains the road-relevant event k from the neighboring CAV j. d_k^j is the distance between the location of the event k and CAV j; time_k^j is the message received time of CAV j; OC_P is the trust value of CAV j. According to [25], the credibility of specific messages (c_k^j) is calculated as:

$$c_k^j = \eta e^{\alpha d_k^j} + (1 - \eta) m_k^j + OC_p. \tag{3}$$

η is the balance coefficient between distance and time; α is a parameter to control the communication boundary; m_k^j is used to prevent replay attack. Next, t_i is the trust value of one road-relevant event g_i, which contains $\{c_1^i, c_2^i, \cdots, c_J^i\}$. t_i is calculated as (3) using Bayesian Inference [38].

$$t_i = P(e \mid g_i) = \frac{P(e) \cdot \prod_{k=1}^{J} P\left(c_i^k \mid e\right)}{P(e) \cdot \prod_{k=1}^{j} P\left(c_i^k \mid e\right) + P(\bar{e}) \cdot \prod_{k=1}^{J} P\left(c_i^k \mid \bar{e}\right)}. \tag{4}$$

$P(e)$ is the prior probability of event e; $\bar{e}$ is the complementary event of e. By comparing with threshold (Thr), the CAV j judges the event g_i. Then, CAV j generates ratings rate_i^j(i.e., +1 or -1) and sends to RSU. By weighted aggregation (4), RSU gets the offset of the trust value $\in [0, 1]$. According to [13], the sensitivity of ratings offset_i is controlled by $F(\cdot)$.

$$\text{offset}_i = \frac{F(m) \cdot m + F(n) \cdot n}{m + n}, \tag{5}$$

where m is the positive(+1) rating and n is the negative(-1) rating;

Compared offset_i with a threshold (Thr_1), if $\text{offset}_i >= \text{Thr}_1$, the offset of one CAV j is $\text{offset}_i^j = \text{rate}_i^j$. Otherwise, the offset of one CAV j is $\text{offset}_i^j = -\text{rate}_i^j$. After the event g_i, the new trust value of CAV j is $OC_p = OC_p + \text{offset}_i^j$. Finally, if a miner finds offset blocks that meet difficulty target, RSUs can share the new trust value of CAV j.

3.5. Atomic Swap-Based Cross-Chain Scheme. For now, the researcher proposed four categories cross-chain technologies [39], which include atomic cross-chain swap, multicentre witness, side chain (Ethereum 2.0), and distributed private key control. Atomic cross-chain swap [31, 32, 40, 41] is a standard method to connect multiple blockchains. Without trusted third parties, the atomic swap can realize fair cross-chain exchange.

The ACSMS can exchange the CAV's data for the blockchain-based PEKS, the BKM, and the BTM. Take as input last signature (Sig, Sn_i, data), old blockchain account (PK_{b_s1}, SK_{b_s1}), and new blockchain account (PK_{b_s2}, SK_{b_s2}). S computes the cross-chain transaction $\text{Asset}_R = \{\text{Sig}(Sn_{i+1} \| ts \| OC_p), Sn_{i+1}, ts, PK_{b_s1} \longrightarrow PK_{b_s2}, OC_p, \text{swap}\}$. The signature $\text{Sig}(Sn_{i+1} \| ts \| OC_p) = P/(H_3(Sn_{i+1} \| ts \| OC_p) + SK_{b_s1})$. After the ACSMS process, if searcher wants to find past information, then searcher uses index Si and PK_{b_s1} to find past signature (Sig), which can check data integrity and authenticate the user. The cross-chain model is illustrated in Figure 5.

The atomic swaps problem in VANETs is as follows. Assume a CAV has an account (PK_{b_s1}, PR_{b_s1}) and holds an asset $Z = (\text{Asset}_S, OC_p)$ in blockchain A. If CAV accesses blockchain B domain, the CAV creates a new blockchain account (PK_{b_s2}, PR_{b_s2}) in blockchain B. By the smart contract [32], the CAV can deal with other CAV or RSU or TA to realized cross-chain swap. Compared with other CAVs and RSU, it is more securely and efficient that CAVs deal with the TA as escrow agent [31, 32, 41]. The escrow agent to occur for the ACSMS is as follows.

4. Security Analysis

4.1. Security of Blockchain-Based PEKS. The security of blockchain-based PEKS consists of three aspects: CAVs' data security preservation [42], conditional identity privacy, and storage correctness guarantee.

(i) *CAVs' data security preservation*. In the ACSMS, with a standard public key encryption $E(\cdot)$, message M with keywords, the TA gets ciphertext Ct_s. Ct_s is encrypted with (SK_s, PK_s, PK_c), any adversary is infeasible to compute SK_s due to the hardness assumption of bilinear Diffie-Hellman problem [43]. Without SK_s, the TA can not decryption Ct_s

(ii) *Conditional identity privacy*. Many ciphertexts are stored in the server. It is difficult for any adversary to find specific ciphertext without trapdoor T_w. However, the ciphertexts have identity privacy during storage; any searcher can use trapdoor T_w to find the ciphertext. The searcher has no right to download ciphertext without the signature of blockchain account b_{s_1}. Therefore, blockchain-based PEKS can resist the keyword guessing attacks (KGA).

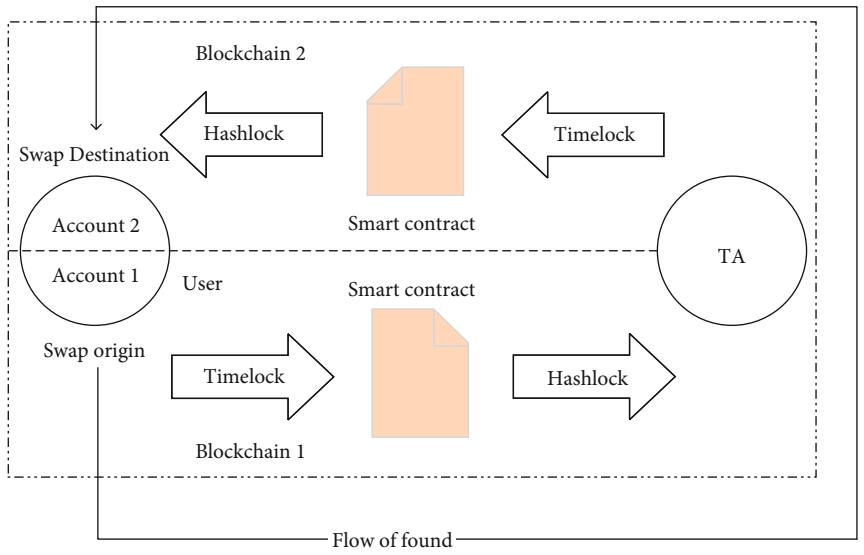

Figure 5: Atomic swap-based cross-chain scheme.

```
Atomic swap for ACSMS {
   depositA, depositB: bool; hash1, hash2 : int256;
//the deposit function allows user transfer of the asset to the smart contract.
   uint startA, startB; //protocol starting time
   initialize {deposited A : = False; deposited B := False; hash1=hash2=0; }
   depositA{if sender = account_A and asset=Z:
      Then depositA : = True and hash = h1 and startA = time()}
//h1 is hash-lock; startA + time_set is time-lock.
   depositB{if sender = TA and asset=Z:
      Then depositB : = True and hash = h2 and startB = time()}
   Finalise{if depositA and depositB and now < startA + time_set :
      Then {depositA : = False; depositB:= False
      send(account_B,Z), send(TA,Z)}}
   cancelA{if (depositA and sha-256(x2) = h2) or now > startA + time_set
      then {depositA : = False; Send(account_A,Z)}}
   cancelB{if (depositB and sha-256(x1) = h1) or now > startB + time_set
      then {depositB : = False; Send(TA,Z)}}
}
```

Algorithm 1

(iii) *Storage correctness guarantee*. The signature of the ciphertext provides storage correctness guarantee, which is obtained from the blockchain

4.2. Security of BKM. A distributed BKM is built through a key management protocol [30] and blockchain technology. ACSMS can achieve subscription validation, server authentication, CAV untraceability, and anonymity. Furthermore, the ACSMS can achieve distributed and tamper-proofing through blockchain technology. In addition, the BKM can against typical attacks towards the VANETs as follows: internal attack, public key tampering, and DoS Attack.

(i) *Internal attack*. The adversary eavesdrops on the communication data between the CAVs and RSU [22]. However, without private key $sH_4(\mathrm{Pid}_i)$ or s $H_4(\mathrm{ID}_{\mathrm{KS}_i})$, adversary cannot get shared symmetric key $K_{S-\mathrm{KS}_i}$ between the CAVs and RSU

(ii) *Public key tampering*. If adversary creates fake pseudo-ID Pid_i without a legal blockchain account, the RSU can verify that $\hat{e}(H_3(\mathrm{Sn}_i \| \mathrm{Pid}_i \| \mathrm{ID}_{\mathrm{KS}_i} \| \mathrm{ts})P + \mathrm{PK}_{b_s}, \mathrm{Sig}_b) \neq \hat{e}(P, P)$. In the ACSMS, legal blockchain account is registered by the TA; adversary cannot counterfeit the registration transaction into the blockchain

(iii) *DoS Attack*. The ACSMS can resist DoS Attack by timestamp ts and hardware security module. If adversary sends a lot of intercepted messages to VANETs, timestamp ensures the freshness of the messages

4.3. Security of BTM. The security of BTM consists of three aspects: conditional identity privacy, prevention of replay attack, and prevention of message spoofing attack.

(i) *Conditional identity privacy*. CAVs broadcast message $\{\text{Message}, \text{Pid}_i, \text{ts}_1, \text{Sig}_{\text{OC}_p}, \text{OC}_P\}$ without blockchain account information; the Pid_i is unlinkable pseudo-IDs. Thus, other vehicles are difficult to analyze the user information with the BTM. But, the RSU can find the blockchain account b_{s_1} through the BKM, and the TA can find relevant CAV information S_{info} by the blockchain-based PEKS

(ii) *Prevention of replay attack*. By time_k^j and m_k^j, if the messages are send by the same CAVs at different times, the credibility of specific messages c_k^j will set to zero. Therefore, equations (2) and (3) will reject the same information with a different timestamp

(iii) *Prevention of message spoofing attack*. Adversary may broadcast fake messages to neighbors. In the ACSMS, we take the Bayesian Inference-based rating generation scheme. Therefore, the receiver will analyze messages from different reference CAVs about the same event

4.4. Security of Atomic Swap. The ACSMS achieves hash locking and smart contract to create conditional payment. The atomicity of atomic swap is guaranteed by the cross-chain protocol. At the same time, the security of the atomic cross-chain swap is ensured by blockchain-based PEKS, BKM, and BTM. The security of atomic swap consists of three aspects: CAVs' data security preservation, conditional identity privacy, and storage correctness guarantee. The atomic swap also against typical attacks towards the VANETs is as follows: internal attack, public key tampering, and DoS Attack. The ACSMS achieves security and anonymity by the blockchain-based PEKS, BKM, and BTM. Furthermore, in ACSMS, only the TA and RSU can create new transaction that can resist targeted attacks [44].

4.5. Comparison of Related Works. There are many blockchain-based application systems that have been developed in VANETs [12, 13, 16, 17, 22, 24]. These researches can protect CAV privacy and ensure security certificate. As shown in Table 1, compared with these schemes, ACSMS integrates the function of the above methods into the ACSMS. Besides, the ACSMS can provide cross-chain process, which boost interoperability between multiple blockchain. Furthermore, compared with [25], the ACSMS can support cross-chain stage security in VANETs.

Table 1 shows the comparison of related work. A safe and reliable decentralized data storage system was built in [17, 18, 20]. The scheme [12, 16, 22, 24] applied blockchain technology into VANETs; the management of CAVs' key is more efficient and secure. Furthermore, by BTM [13, 24, 25], the CAV could forward reliable announcements without relevant CAVs' information, and the users with pseudo-ID are encouraged to forward announcements. The ACSMS scheme uses PEKS and blockchain technology to integrate above the functions to support multiple blockchain-based applications with a single blockchain account. In addition, we use atomic swap smart contract to achieve cross-chain interoperability among blockchain-based systems.

5. Performance Evaluation

The performance evaluation of ACSMS was carried out using simulations. Our results are generated using pypbc (the encryption algorithm) and truffle (blockchain for ACSMS). ACSMS is simulated by our laptop with 2.6Ghz Core i7 CPU and 16G RAM and display card Radeon Pro 555X 4 GB. We also simulated the ACSMS in the elastic compute service with Ubuntu 18.02, 4v 3.1GHz/3.5Ghz CPU and 16G RAM.

The simulation of the trust management system in VANET [45] shows that the rate of malicious vehicles. A variable percentage is generated to simulate a different percentage of malicious vehicles like in [45]. The performance of trust management is evaluated through the trust value of CAVs OC_P. However, our BTM scheme only affects the initial OC_P, and the RSU uploads trust value after receiving multiple pieces of evidence. Therefore, blockchain will not change the efficiency of trust management [32, 45, 46].

5.1. Blockchain-Based PEKS. To evaluate blockchain-based PEKS and other related woks, we use a cryptographic function called Type-A [17]. We set $|\mathbb{Z}_q^*| = 160$ bits and $|\mathbb{G}_1| = |\mathbb{G}_2| = 1024$ bits by setting parameters qbits = 1024 and rbits = 160, where $|x|$ is the length of x. We take ZSS (Zhang-Safavi-Naini-Susilo) short signature scheme from Bilinear Pairing [34]. Besides, pypbc requires the hash of the signature scheme and key in the same group. So, we use 160 bits $\text{Sig} = P/(H_1(\text{Ct}_s) + x)$ instead of $\text{Sig} = P/(H_3(\text{Ct}_s) + x)$.

Figure 6 shows the computation cost comparisons for the FTDS [17] and the BSPP [20]. The computation costs of encrypting ciphertext and decrypt ciphertext are constant, which not show in the picture. Each scheme uses 100 keywords to encrypt 100 files. The FTDS [17] is slower than the ACSMS and the BSPP in KeyGen and PEKS, because the FTDS uses key-aggregate encryption, which is a decentralized PEKS scheme. In BSPP [20], the user's medical data is encrypted and stored in the server of the hospital. The corresponding index of user data is stored in the blockchain. It can be seen that the calculation delay of BSPP is close to the ACSMS, because the calculation of KeyGen, Trapdoor, PEKS, and Test all depends on the PEKS algorithm.

The blockchain storage time of the three schemes is similar. Because all three plans store indexes or proofs, the small amount of data will not affect the blockchain's storage speed. The block generation rate will limit blockchain-based on workload consensus. By adjusting the difficulty target, administrators can change the block generation rate of the blockchain. However, too fast block generation rate will lead to the blockchain bifurcating. Therefore, the block generation rate of the typical blockchain is maintained at a certain level. Consequently, we can consider that the three schemes have the same computing delay on the blockchain storage.

Table 1: Comparison of related works.

Feature	[12]	[16]	[22]	[13]	[25]	[24]	[18, 20]	[17]	ACSMS
PEKS	×	×	×	×	×	×	√	√	√
BKM	√	√	√	×	×	√	×	×	√
BTM	×	×	×	√	√	√	×	×	√
PoW	√	√	√	√	×	√	√	×	√
Smart contract	×	×	√	×	√	×	×	×	√
No TA	√	√	×	√	×	×	×	√	×
Key agreement	√	√	√	×	√	×	×	×	√
Cross-chain	×	×	×	×	√	×	×	×	√

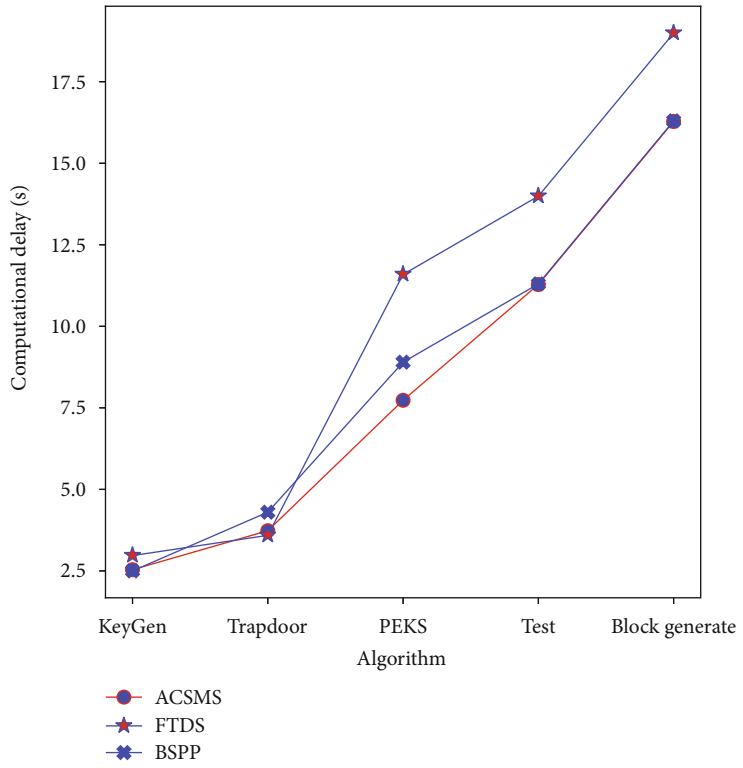

Figure 6: Computation cost comparison for algorithms.

5.2. BKM. Then, we study the processing time cost for the BKM. Our BKM scheme based on PairHand [30], the user cryptographic operation of [30] is 1ECSM+1Pairing. Based on PairHand, we use blockchain to realize distributed key authentication. Therefore, the user cryptographic operation of the BKM in ACSMS is 1ECSM+2Pairing+1hash. In the PairHand, the RSU will transmit messages to TA after authentication, which can find the real identity of CAVs. However, if CAV transmission legal pseudo-IDs to the adversary, the RSU cannot distinguish the adversary and the normal CAVs in the authentication. Although our scheme costs more processing time than the PairHand, our schemes can provide decentralized key authentication than [30]. To evaluate the performance of our scheme, we compare the conventional handover schemes [12, 30].

We can obtain that the exponential increase of blockchain key transfer time in decentralized BKM [12] as shows in Figure 7. In [12], the keys of new joining members are delivered by the blockchain. Our BKM only uses the blockchain record S_{info} to authenticate users, which increases linearly. Although, in the experiment, our key transmission time authenticated by this scheme is relatively large. But, RSU only needs to find the user's signature in the blockchain. There is no need to store data into the blockchain. In this way, our scheme can better support other blockchain-based applications.

5.3. Atomic Cross-Chain Swap. To evaluate the effect of ACSMS, we realize the atomic swap protocol in python. Two PoW-based blockchains are built in python. Administrators can set the difficulty target of each blockchain to adjust blockchain throughput. The total number of CAVs is 1000, and $Z = (\text{Asset}_R, \text{OC}_p)$ is calculated by CAVs. Furthermore, there are average packet latency and blockchain

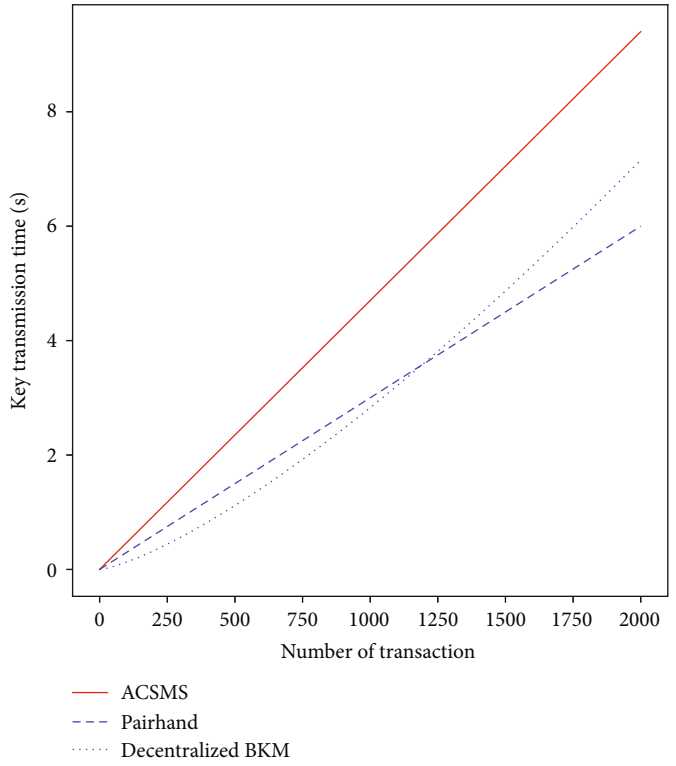

Figure 7: Processing time comparison different schemes.

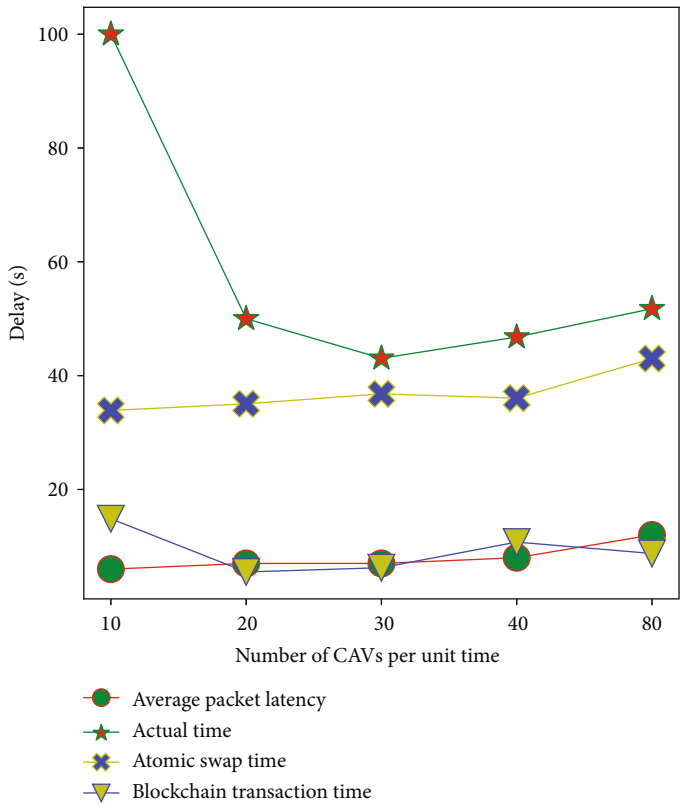

Figure 8: Average delay over different number of CAVs.

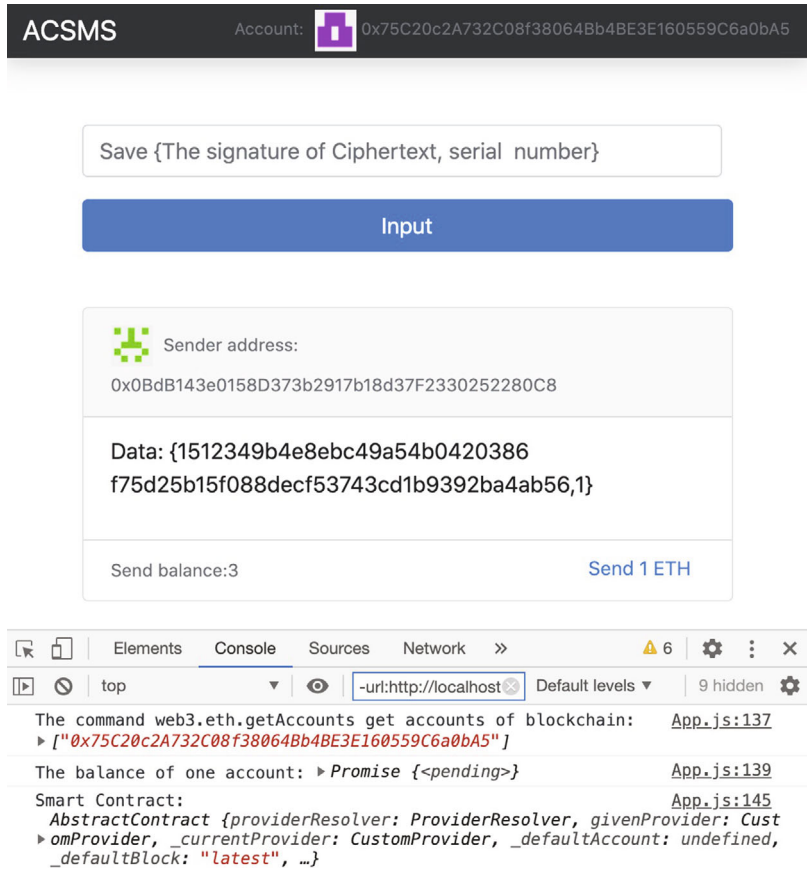

Figure 9: The application of ACSMS.

transaction time in the blockchain-based VANET application [22]. As Figure 8 shows, the delay of atomic cross-chain swap changes with the number of CAVs per unit time.

5.4. Engineering Applications. On Ethereum [14], the user can write a smart contract to realize a distributed program. In order to evaluate our proposed ACSMS, we have built a web application of ACSMS, as illustrated in Figure 9.

The application of ACSMS was built as follows:

(1) We installed truffle, Ganache, and node.js v9.10.0. Next, we configured babel (a JavaScript compiler), bootstrap (HTML, CSS, and JS library), identicon (a visual representation of a hash value), chai (a BDD/TDD assertion library), mocha (a feature-rich JavaScript test framework), and web3 (web3.js interact with Ethereum node using HTTP, IPC, or WebSocket). Then, we use those to create a package.json file for the project using npm install to install these libraries

(2) We developed a smart contract in atom text editor and write test cases for use with TDD (test-driven development, chai, and mocha). Then, metamask was used to access ethrum enabled distributed applications

(3) As shown in Figure 9, the application of ACSMS saves $(\mathrm{Sig}, \mathrm{Sn}_i)$ into the blockchain network, which blockchain-based PEKS was realized

(4) With metamask and web3 command web3.eth.getAccounts(·), user blockchain account was got for BKM. RSU can retrieve the corresponding $\{b_{s_1}, \mathrm{Sn}_i\}$ from the blockchain to ensure user legitimacy within pseudo-ID

(5) With metamask and web3 command web3.eth.getBalance (Account), the user account balance was got for BTM. To share trust value with pseudo-ID, RSU generates an authentication code of CAV's trust value OC_P, where OC_P is reflected by the CAV's blockchain in account balance. Furthermore, after BTM, RSU could send the balance to the user account through the button

6. Conclusion

In this paper, we proposed the ACSMS to boost the scalability of blockchain-based applications in VANETs. Besides, the ACSMS integrates the blockchain-based schemes into a single blockchain through the bilinear pairing. First, we designed a blockchain-based PKES to check data integrity and authenticate the user. Second, we designed blockchain-

based key management. The BKM uses data of blockchain-based PKES to provide decentralized key authentication. Third, based on the BKM, CAVs get proof of trust value from the RSU. Therefore, the CAVs can share reasonable trust value in blockchain-based trust management. Finally, the atomic swap-based cross-chain scheme can exchange user trust value and data index across the chain. The searcher quickly finds the data of blockchain-based PKES in the different blockchain. However, the simulation results demonstrate that the performance of ACSMS is no improvement than the previous scheme. Nevertheless, security analysis shows that our schemes can achieve the security requirements of VANETs.

In the future, we will take PEKS for cloud storage services. The sender in ACSMS can outsource the data to the cloud storage services. On the other, we may use PoS-based blockchain instead of PoW-based blockchain as long as security is ensured.

Data Availability

The data are available at https://github.com/tanchenkai/Paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. Xu, X. Li, L. Jiao et al., "Bagkd: a batch authentication and group key distribution protocol for vanets," *IEEE Communications Magazine*, vol. 58, no. 7, pp. 35–41, 2020.

[2] J. Liu, J. Li, L. Zhang et al., "Secure intelligent traffic light control using fog computing," *Future Generation Computer Systems*, vol. 78, pp. 817–824, 2018.

[3] S. I. Fadilah, A. R. M. Shariff, and M. N. M. Hilmi, "Crash avoidance based periodic safety message dissemination protocol for vehicular ad hoc network," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Kuala Lumpur, Malaysia, April-May 2019.

[4] K. Sjoberg, P. Andres, T. Buburuzan, and A. Brakemeier, "Cooperative intelligent transport systems in Europe: current deployment status and outlook," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 89–97, 2017.

[5] A. Shahzad, M. Lee, Y. K. Lee et al., "Real time MODBUS transmissions and cryptography security designs and enhancements of protocol sensitive information," *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.

[6] L. Shu, Y. Zhang, Z. Yu, L. T. Yang, M. Hauswirth, and N. Xiong, "Context-aware cross-layer optimized video streaming in wireless multimedia sensor networks," *The Journal of Supercomputing*, vol. 54, no. 1, pp. 94–121, 2010.

[7] W. Wu, N. Xiong, and C. Wu, "Improved clustering algorithm based on energy consumption in wireless sensor networks," *IET Networks*, vol. 6, no. 3, pp. 47–53, 2017.

[8] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Vanet security challenges and solutions: a survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017.

[9] K. Huang, Q. Zhang, C. Zhou, N. Xiong, and Y. Qin, "An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2704–2713, 2017.

[10] A. Vaibhav, D. Shukla, S. Das, S. Sahana, and P. Johri, "Security challenges, authentication, application and trust models for vehicular ad hoc network-a survey," *International Journal of Wireless and Microwave Technologies*, vol. 7, no. 3, pp. 36–48, 2017.

[11] Q. Zhang, C. Zhou, N. Xiong, Y. Qin, X. Li, and S. Huang, "Multimodel-based incident prediction and risk assessment in dynamic cybersecurity protection for industrial control systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 10, pp. 1429–1444, 2015.

[12] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.

[13] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019.

[14] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[15] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 263–284, 2016.

[16] R. Shrestha, R. Bajracharya, and S. Y. Nam, "Blockchain-based message dissemination in VANET," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, Kathmandu, Nepal, 2018.

[17] J. Sun, H. Xiong, S. Zhang, X. Liu, J. Yuan, and H. Deng, "A secure flexible and tampering-resistant data sharing system for vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12938–12950, 2020.

[18] Y. Zhang, C. Xu, J. Ni, H. Li, and X. S. Shen, "Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage," *IEEE Transactions on Cloud Computing*, pp. 1–14, 2020.

[19] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *International conference on the theory and applications of cryptographic techniques*, Springer, Berlin, Heidelberg, May 2004.

[20] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, no. 8, p. 140, 2018.

[21] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 58241–58254, 2019.

[22] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5836–5849, 2020.

[23] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, Manubot, 2019.

[24] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "Bars: a blockchain-based anonymous reputation system for trust management in VANETs," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science*

And Engineering (Trust-Com/BigDataSE), New York, NY, USA, August 2018.

[25] N. Zhao, H. Wu, and X. Zhao, "Consortium blockchain-based secure software defined vehicular network," *Mobile Networks and Applications*, vol. 25, no. 1, pp. 314–327, 2020.

[26] W. Guo, N. Xiong, A. V. Vasilakos, G. Chen, and H. Cheng, "Multi-source temporal data aggregation in wireless sensor networks," *Wireless Personal Communications*, vol. 56, no. 3, pp. 359–370, 2011.

[27] Y. Yang, N. Xiong, N. Y. Chong, and X. Défago, "A decentralized and adaptive flocking algorithm for autonomous mobile robots," in *2008 The 3rd International Conference on Grid and Pervasive Computing-Workshops*, Kunming, China, 2008.

[28] C. Lin, Y. X. He, and N. Xiong, "An energy-efficient dynamic power management in wireless sensor networks," in *2006 Fifth International Symposium on Parallel and distributed computing*, Timisoara, Romania, July 2006.

[29] J. Santa, L. Bernal-Escobedo, and R. Sanchez-Iborra, "On-board unit to connect personal mobility vehicles to the IoT," *Procedia Computer Science*, vol. 175, pp. 173–180, 2020.

[30] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 48–53, 2012.

[31] R. van der Meyden, "On the specification and verification of atomic swap smart contracts (extended abstract)," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Seoul, Korea (South), May 2019.

[32] M. H. Miraz and D. C. Donald, "Atomic cross-chain swaps: development, trajectory and potential of non-monetary digital token swap facilities," *Annals of Emerging Technologies in Computing*, vol. 3, no. 1, pp. 42–50, 2019.

[33] C. Gu and Y. Zhu, "New efficient searchable encryption schemes from bilinear pairings," *IJ Network Security*, vol. 10, no. 1, pp. 25–31, 2010.

[34] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *International Workshop on Public Key Cryptography*, pp. 277–290, Springer, 2004.

[35] Z. Jing, C. Gu, C. Ge, and P. Shi, "Cryptanalysis of a public key cryptosystem based on data complexity under quantum environment," *Mobile Networks and Applications*, pp. 1–7, 2020.

[36] L. Li, G. Xu, L. Jiao et al., "A secure random key distribution scheme against node replication attacks in industrial wireless sensor systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2091–2101, 2020.

[37] L. Li, J. Liu, L. Cheng et al., "Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[38] M. Raya, P. Papadimitratos, V. D. Gligor, and J. P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, Phoenix, AZ, USA, April 2008.

[39] D. Yang, C. Long, H. Xu, and S. Peng, "A review on scalability of blockchain," in *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology (ICBCT'20), Association for Computing Machinery*, pp. 1–6, New York, NY, USA, 2020.

[40] M. Herlihy, "Atomic cross-chain swaps," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pp. 245–254, ACM, 2018.

[41] T. Nolan, *Alt Chains and Atomic Transfers*, Bitcoin Forum, 2013.

[42] Z. Jing, C. Gu, Y. Li et al., "Security analysis of indistinguishable obfuscation for internet of medical things applications," *Computer Communications*, vol. 161, pp. 202–211, 2020.

[43] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology — CRYPTO*, pp. 213–229, Springer, 2001.

[44] S. Bartolucci, F. Caccioli, and P. Vivo, "A percolation model for the emergence of the bitcoin lightning network," *Scientific reports*, vol. 10, no. 1, 2020.

[45] F. Gómez Mármol and G. Martínez Pérez, "Trip, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 934–941, 2012.

[46] Y. Xiao, Y. Liu, and T. Li, "Edge computing and blockchain for quick fake news detection in IoV," *Sensors*, vol. 20, no. 16, p. 4360, 2020.