# A Blockchain-Based Accident Forensics System for Smart Connected Vehicles

***Abstract*—Internet of Vehicles (IoV) can greatly enhance transportation efficiency and safety. However, in the event of a traffic accident, the cause of the accident is more complex due to the mutual interaction and intelligent decision-making between vehicles and road infrastructure. The subject of accident responsibility is no longer limited to the driver only, the relevant stakeholders may tamper with the evidence to avoid responsibility, and the traditional centralized data depository system is often difficult to prove their own innocence. Therefore, we propose a blockchain-based accident forensics system underlying the V2X with Secure Credential Management System (SCMS). The system realizes a comprehensive restoration of the whole process of the accident by collecting all relevant V2X messages within the spatial and temporal scope of the accident, and protects the privacy of the identity of the forensic vehicles by utilizing the linkage value of the SCMS and the internal security pseudonym certificate mechanism. To reduce the on-chain data processing overhead, we propose a storage and query scheme for on-chain data in collaboration with the cloud, and design an efficient on-chain data structure based on the blockchain gateway. In addition, based on Bloom filters, we build a fast evidence retrieval mechanism from massive V2X messages that combines off-chain preprocessing and on-chain queries. Finally, we develop a prototype system for performance evaluation and verify the efficiency of the proposed method for incident message retrieval under large-scale full-volume V2X messages.**

***Index Terms*—blockchain, IoV, forensics, cloud-chain collaboration, Bloom filter**

## I. Introduction

The intelligent transportation system, exemplified by the Internet of Vehicles (IoV) and autonomous driving, has experienced significant growth in recent years, providing individuals with novel travel experiences [1]. However, while the use of these emerging technologies has improved transportation efficiency and safety, it has also created new challenges in determining liability for traffic accidents. The causes of these accidents are more complex than traditional traffic accidents and include poor decision-making by autonomous driving systems and flawed interactions between IoV entities. As a result, liability for accidents is no longer limited to the driver, but includes many parties such as automakers, IoV roadside facilities, V2X communications and self-driving software providers. Moreover, the current method of keeping vehicle data within the car itself or in the data center of the automobile manufacturer does not ensure the integrity of the evidence, as there is a risk that stakeholders will deliberately tamper with or delete the data in order to avoid liability. In addition, centralized storage methods cannot be self-evident.

Blockchain technology has unique advantages in terms of credibility. Due to its decentralized architecture and immutability, it ensures the integrity of stored evidence, making it easier to resolve traffic accidents involving multiple parties [2]. Efforts to utilize blockchain technology to solve the problem of attributing liability for the IoV accidents have already begun. Cebe et al proposed a blockchain-based forensic architecture for self-driving cars that collects, stores and analyzes data related to vehicle accidents for forensic purposes [3]. Given the challenges associated with collecting evidence and attributing blame in car accidents, a blockchain-based e-data forensics system was proposed within the IoV communication architecture [4]. The system utilizes the decentralized storage mechanism of blockchain to achieve remote storage of electronic evidence. It also accomplishes rapid retrieval of electronic evidence using smart contract technology. Dorri et al. explored the use of blockchain technology to improve automotive security [5], including potential applications such as insurance and wireless software upgrades for smart cars. However, blockchain scalability, privacy leakage related to vehicle identity, etc. becomes bottlenecks for the technology to become practical. In [6], a blockchain-based liability determination system for self-driving car accidents was proposed. Using this paradigm, one can tackle the issue of associated responsible entities avoiding accountability in accidents by analyzing potential responsible subjects.

Most of the above work proposed a technical framework for applying blockchain to vehicle data depository and accident forensics, but lacked in-depth technical implementation and application feasibility analysis. In particular, the scalability of large-scale vehicle data deposit, the integrity and efficiency of accident forensics, and the privacy preservation during accident forensics deserve in-depth investigation. Moreover, smart connected vehicles involve complex interactions among different entities, and the above work [3-6] can only collect some of the vehicle state data as evidence shortly before the collision. Therefore, it is not possible to fully reconstruct the entire accident process.

To advance the utility of blockchain in accident forensics, we propose techniques for full-volume depositing of V2X message and efficient querying of spatio-temporal correlation data of accidents. Specifically, in order to solve the scalability problem of full-volume high-frequency V2X message depositories, we adopt the chain-cloud collaboration technology-storing the V2X original messages on the distribution cloud, and storing the on-chain data with lightweight data encapsulation for V2Xoriginal messages. We design a compact on-chain data structure for V2X messages, including the time, linkage value, integrity proof, and distributed storage address of the

V2X message. Encapsulation of on-chain data can be achieved through a blockchain gateway embedded in the vehicle or RSU. It should be noted that we utilize the message linkage value of the Security Credential Management System (SCMS) [7] to achieve fast indexing of incident temporal and spatial correlation data. Another advantage of using message linkage values for message indexing is the reusability of the internal mechanism of the SCMS to realize the association of message link values to pre-linkage values, thus ensuring the privacy protection of vehicle identity data during the accident forensic process. To further improve the efficiency of vehicle-road-cloud-chain data retrieval, bloom filter is proposed for fast addressing of distributed cloud storage data.

In summary, our proposed blockchain-based accident forensics system for smart connected vehicles underlying the V2X with SCMS has the following characteristics:

- **Comprehensiveness**: The ability to retrieve V2X messages from all vehicles within the spatial and temporal boundaries of the incident, allowing for a complete recovery of the entire incident.
- **Credibility**: Ensure data integrity and non-repudiation of the multi-party forensic process by leveraging the hard-to-tamper nature of blockchain data storage and multi-party data governance consensus mechanisms.
- **Confidentiality**: The encapsulation of V2X messages by the blockchain gateway and the use of the SCMS internal linkage value function ensure the privacy of the vehicle identity during on-chain data storage and multi-party forensics.
- **Rapidity**: Rapid retrieval of spatio-temporally correlated incident data from large-scale V2X messages is achieved through linkage values and bloom filters.
- **Lightweight**: Scalable storage and efficient indexing of large-scale, high-frequency, full-volume V2X messages through the design of a compact on-chain data structure in collaboration with a distributed cloud database.

The remaining parts of the paper are organized as follows. Section 2 describes the system architecture. Section 3 elaborates system working mechanism including V2X messages storage and accident evidence forensics. Section 4 takes experiment evaluation. Section 5 summarizes the article.

## II. System Architecture

Our blockchain-based accident forensics system for smart connected vehicles is constructed underlying V2X with SCMS as shown in Fig.1. The system architecture consists of four parts: the blockchain network layer, the blockchain application layer, the distributed cloud storage database, and the V2X with SCMS.

The blockchain network layer is a consortium blockchain using state-of-the-art efficient and scalable Byzantine-like consensus protocols [8]. Blockchain network comprises accident related parties, and consensus nodes are fixed by a few institutions (police department, traffic department, court, and insurance company) that possess high credibility and robust storage and computational capabilities. Meanwhile, other accident-related parties with limited resources (such as car owners, auto manufacturers, software service providers, maintenance shops, etc.) can act as light nodes and freely move in and out of blockchain network. They do not engage in consensus computation and solely save block header data synchronously, with little storage expenses. Nevertheless, by utilizing Simplified Pay Verification (SPV), they can also engage in post-accident forensics, thereby guaranteeing the equitable rights of all parties involved.

The blockchain application layer is designed to implement fundamental system functions(such as storing V2X messages and conducting accident evidence forensics. This layer consists of several components: the messages repository smart contract (MR contract), the evidence sources indexes smart contract (ESI contract), the evidence verification smart contract (EV contract), and the Bloom filter. MR contract is devised combined with off-chain distributed cloud storage to store V2X messages after structured encapsulation as evidence, where a specific on-chain data structure is devised to facilitate efficient evidence retrieval as well as reducing on-chain storage burden. Considering the huge amount of V2X messages, evidence retrieval may take a lot of time, so we establishe a fast retrieval mechanism by combining off-chain preprocessing and on-chain query based on Bloom filters for accident-related evidence.ESI contract aims to find all vehicles within the spatial and temporal scope of the accident while protecting forensic vehicles identity privacy through SCMS. EV contract is speicically built to automatically validate the integrity and origin of retrieved evidence combing data both on chain and on cloud without any human interference. Through the above smart contracts, they automate and collaborate on V2X message storage and incident forensics tasks.The details will be elaborated in the Section 3.

The distributed cloud storage database is responsible for storing the original full-volume V2X messages, which are used to alleviate the on-chain storage and computational load and to perform forensic tasks in conjunction with the blockchain. There have been related works that utilize scaled V2X messages for big data analysis [9], so archiving full-volume V2X messages has wide application value. How to realize efficient distributed storage of high-frequency large-scale V2X messages is out of the scope of this paper, and related work can be referred to [10].

In the SCMS, the registration authority (RA) is responsible for verifying the enrollment certificate of vehicle on-board unit (OBU) and pseudonym certificate authority (PCA) will generate pseudonym certificates for verified OBU. During the provision of pseudonym certificates, “linkage value” is calculated and embedded into pseudonym certificates to improve the efficiency of revoking pseudonym certificates. During a specific period, linkage values are collision resistant. In other words, linkage value and pseudonym certificate have a one-to-one correspondence relationship in a certain period. We take advantage of such relationship to use linkage value as IoV message origin vehicle identifier to help evidence retrieval. To prevent linking pseudonym certificates of a single device, there

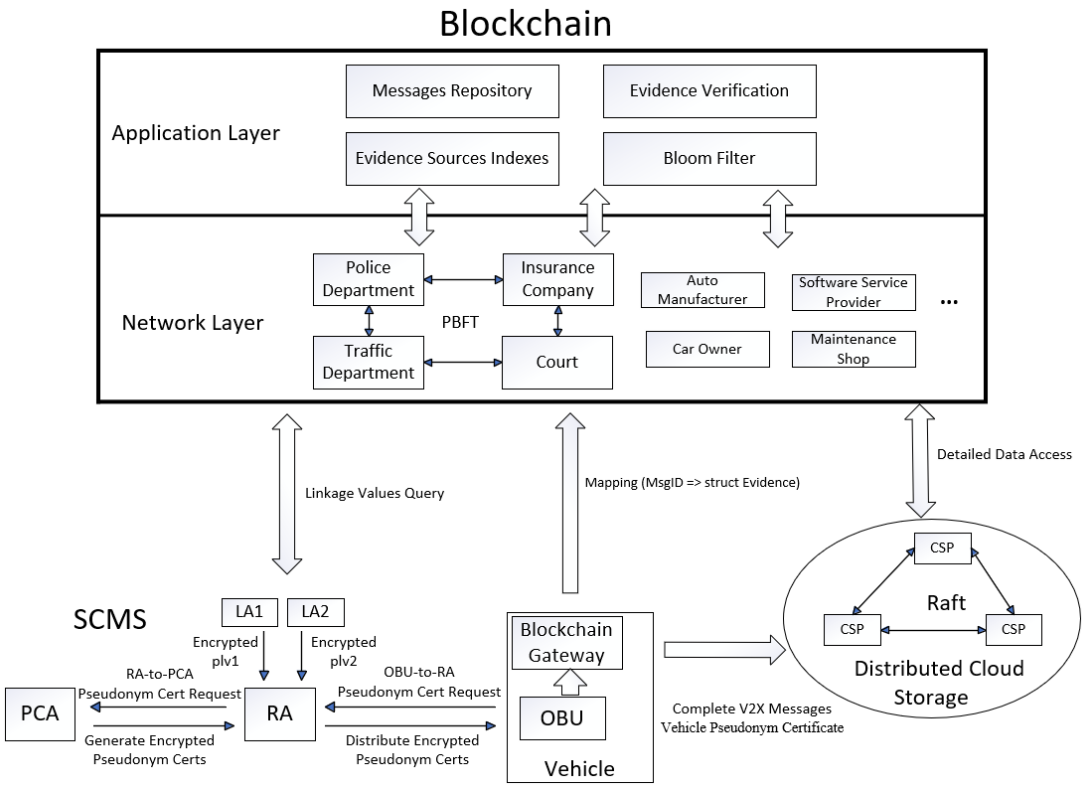


Fig. 1. System Architecture

are two mutually independent linkage authorities (LA1 and LA2). They respectively generate pre-linkage values, which are further performed an XOR operation by PCA to calculate the linkage value. Further information regarding the linkage value will be provided in the following section.

## III. Working Mechanism

Our proposed system's working mechanism can be divided into two parts: V2X message storage and accident evidence forensics. The operation of storing V2X messages is consistently carried out in conjunction with IoV communication. The accident evidence forensics process is initiated upon the occurrence of an accident. The details are elaborated as follows.

### A. V2X Messages Storage

In IoV, vehicles engage in frequent transmission and reception of communication messages with nearby vehicles or RSUs. The communication messages encompass vehicle status and coordination data, which can be utilized for the identification of accident causes. As shown in Fig.1, the blockchain gateway is embedded into vehicle, enabling real-time reading of communication messages from OBU. In order to minimize on-chain storage, it is not appropriate to store the full volume of V2X communication data directly on the chain. We propose to embed the blockchain gateway in the in-vehicle terminal, through which the V2X message is encapsulated and is organized into a predefined data structure. The original complete information is uploaded to a distributed cloud storage database simultaneously. The data stored in blockchain and cloud respectively is shown in Table 1.

The on-chain data is organized in mapped data types: keys are hashes and mapped values are structured data. The key is the hash value of identified information including message time, linkage value, and type as following:

$$MsgID = Hash(alignedTime||linkageValue \\ ||messageType) \quad (1)$$

Since the data transmitted via V2V and V2I, called Basic Safety Messages (BSM), is sampled at 10 Hz, we use a post-alignment time based on 0.1 seconds. In addition, millisecond times can be further captured in the mapped value structure data Evidence, if required. The mapped value structure data Evidence contains the millisecond time, the location point, the message hash, the digital signature and the cloud storage address of the message. Message hashing and digital signatures are used to check the integrity of the message and the non-repudiation of the source. The cloud storage address of the message is designed for the on-chain EV contract in order to retrieve the original complete information and the corresponding pseudonymized certificates from the cloud.

The specific on-chain data structure is devised to enable swift retrieval of evidence from a vast amount of IoV messages and support the following on-chain automatic evidence verification. We only need calculate accident related potential

Table 1. Data Stored Separately In Blockchain And Cloud

| **Data stored in blockchain** | **Data stored in cloud** |
|---|---|
| Mapping (MsgID ⇒ struct Evidence) | Complete IoV message |
| MsgID = Hash(alignedTime ‖ linkageValue ‖ messageType) | Vehicle pseudonym certificate |
| Struct Evidence = {millisecondTime, positionPoint, messageHash, digitalSignature, cloudAddress} | |

keys off chain and input them into MR contract to query corresponding evidence on chain. Compared with method of on-chain conditional retrieval based on accident time and position, it need not traverse all IoV messages to achieve all accident-related messages and does not consume on-chain computation resources for on-chain querying need not consensus, thus more efficient. To further reduce the evidence retrieval time, combining off-chain preprocessing and on-chain querying, we propose a querying approach based on Bloom filters, a probabilistic data type that provides faster querying efficiency compared to mapped data types due to its compact data structure and the way it handles queries [9]. The Bloom filter uses multiple hash functions to map an element to multiple locations in an array of bits. To check if an element exists, simply verify that the bits are set at those locations. Since the number of hash functions is fixed and small, the process is very fast and the time required is independent of the key size, which remains constant. In contrast, mapped data types use hash tables whose lookups may involve comparing the keys of the query to the keys actually stored, which can be slower, especially if those keys are large or if hash collision happens that needs to be handled through techniques such as chaining or open addressing. In our scheme, the blockchain gateway adds the key MsgID to the Bloom filter while uploading the mapping data to the blockchain. Bloom filter is set in blockchain node to help swift evidence retrieval when receiving evidence retrieval request from police. When retrieving evidence, the bloom filter can extremely quickly exclude some keys that definitely do not exist, which helps reduce on-chain mapping querying times.

Algorithm 1 gives the complete process of structured encapsulation and key-value indexing for v2x message on the blockchain gateway. Bloom filter is initialized through setting bit array size and adopted hash functions (line 2-3 in Algorithm 1). The gateway stores complete IoV messages to cloud and structures messages before uploading into chain (line 7-11 in Algorithm 1). Besides, it adds the key MsgID into Bloom filer (line 12-14 in Algorithm 1).

### *B. Accident Evidence Forensics*

This subsection provides a comprehensive explanation of the accident evidence forensics method, as depicted in Fig. 2. In general, the process of forensics can be split into accident report (step 1-2 in Fig. 2), evidence retrieval (step 3-7 in Fig. 2), and evidence verification (step 8-9 in Fig. 2).

For a complete reconstruction of the accident, the design of accident evidence forensics should fulfill integrity of accident data, Scalability of evidence retrieval, Consistency and non-repudiation, and Privacy-preservation.

*1) Accident Evidence Retrieval:* When the accident happens, collision vehicle detects collision event by sensor and then calls EIS contract to report accident. Specifically, collision vehicle puts a new key accident time  location string that can uniquely identify an accident to the key value pair mapping(accident time  location string => linkage values array), and adds its single linkage value to corresponding linkage values array (step 1a in Fig. 2). If more than one vehicle collides in an accident, the accident identifier accident time  location string will be recorded only once and the other collision vehicles will merely upload their linkage values to corresponding linkage values array.

In order to obtain complete traffic situational awareness data of the accident process, we propose a spatio-temporal correlation accident evidence retrieval algorithm as shown in Algorithm 2. First, collision vehicle broadcasts evidence sources gathering request that contains the mapping's key accident time  location string to surrounding vehicles (step 1b in Fig. 2). Once receiving the request, each surrounding vehicle within the spatial and temporal scope of the accident will upload its single linkage value to array (step 1c in Fig. 2). Considering that the on-chain data is open to all participants of consortium blockchain, if the vehicle directly uploads all linkage values to the blockchain, malicious attacker may launch a link attack to trace the vehicle. Therefore, the vehicle only uploads its single linkage value and the other linkage values can be accessed later by police through the internal mechanism of the SCMS, thus fully protecting the privacy of the vehicle.

Once police receives a new accident report event notification (step 2a in Fig. 2), it will then further call ESI contract (step 2b in Fig. 2) to obtain the whole linkage values array based on the mapping key accident time  location string (step 2c in Fig. 2) and further enquiry all other linkage values through SCMS (line 4-9 in Algorithm 2). To better understand the enquiry of accident related linkage values (obtain all other contemporary linkage values from single linkage value without compromising forensic vehicles forward anonymous communication), detailed linkage values generation process is explained as follows.

LA1 and LA2, respectively denoted by $la\_id_1, la\_id_2$, separately picks a random 128-bit string called the initial linkage seeds $ls_1(0)$ and $ls_2(0)$, which can be further recursed by unidirectional hashing algorithm SHA256 to generate more $u$-byte-long linkage seeds for different time period, denoted by $i$. The recursive formula is:

$$ls_x(i+1) \leftarrow \mathrm{H}_u\left(ls_x(i)||la\_id_x\right), x \in \{1,2\} \qquad (2)$$

For a given linkage seed, linkage authority takes it as AES

**Algorithm 1** Structured Encapsulation and Key-Value Indexing for V2X Message on the Blockchain Gateway

**Setup:**
Initialize bitArray of size $N$
Initialize $k$ hash functions
**Input:** IoV Message
**Output:** storageResult
**Start:**
$cloudAddress \leftarrow StoreCloud(IoVMessage)$
$alignedTime \leftarrow millisecondTime\,after\,alignment\,0.1s$
$MsgID \leftarrow Hash(alignedTime \parallel linkageValue \parallel messageType)$
$structEvidence \leftarrow \{millisecondTime, positionPoint, messageHash, digitalSignature, cloudAddress\}$
$mapping[MsgID] \leftarrow struct\,Evidence$
for each HashFunction in $k$
    $index \leftarrow HashFunction(MsgID) \mod N$
    $bitArray[index] \leftarrow 1$
**End**

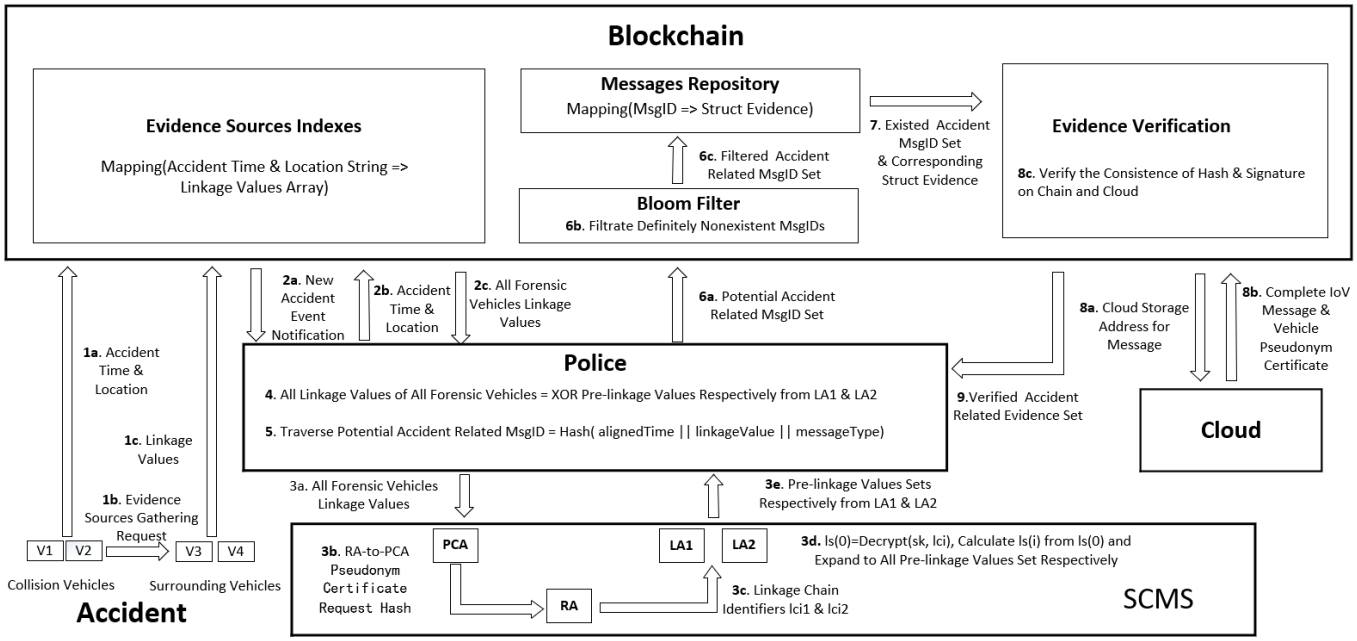


Fig. 2. The Accident Related IoV Messages Retrieval and Verification Diagrams

key in the Davies-Meyer mode to calculate corresponding $j_{\max}$ pre-linkage values, each is a $v$-byte-long bit-string. The calculation formula is: For a given linkage seed, linkage authority takes it as AES key in the Davies-Meyer mode to calculate corresponding $j_{\max}$ pre-linkage values, each is a $v$-byte-long bit-string. The calculation formula is:

$$plv_x(i,j) \leftarrow \left[E\left(l_x(i), (la\_id_x \| j)\right) \oplus (la\_id_x \| j)\right]_v, \quad x \in \{1,2\}, 0 \le \mathrm{j} \le j_{\max} \tag{3}$$

Then PCA receives $plv_x(i,j)$ and calculates linkage value:

$$lv(i,j) \leftarrow plv_1(i,j) \oplus plv_2(i,j) \tag{4}$$

To search the correct linkage seed when revoking pseudonym certificates, Linkage Chain Identifier (LCI) is designed to select a specific linkage chain used by a certain vehicle. LCI is calculated by encrypting the initial linkage seed with linkage authority's public key:

$$lci_x \leftarrow E\left(pk_x, ls_x(0)\right), x \in \{1,2\} \tag{5}$$

The proposed accident related linkage values enquiry is similar to SCMS certificate revocation. However, we do not demand the vehicle linkage seed and only obtain pre-linkage values respectively from LA1 and LA2 through SCMS internal secure mechanism. If the vehicle linkage seed is achieved from SCMS, the unlink ability of vehicle certificates is compromised for all the forward linkage values can be easily

calculated from linkage seed. The proposed accident related linkage values enquiry is detailed as follows.

First, PCA looks up the RA-to-PCA pseudonym certificate request hash corresponding to the inputted single linkage value from its local database and sends the hash to RA. According to the hash, RA identifies the linkage chain identifiers $lci1$ & lci2 of the vehicle from its local database and sends them respectively to linkage authorities LA1 & LA2. LA1 & LA2 take following operations to get pre-linkage values in time $t$ corresponding to accident time. Herein, each linkage authority uses its secret key to decrypt the initial linkage seed:

$$ls_x(0) \leftarrow \text{Decrpt}\left(sk_x, lci_x\right), x \in \{1, 2\} \tag{6}$$

The linkage seed of time $t$ can be achieved with the recursive formula (2). Based on linkage seed of time $t$, LA1 & LA2 can calculate all pre-linkage values using formula (3) and return them to police respectively. After receiving all pre-linkage values, police calculates linkage values by XORing the pre-linkage values. The linkage values enquiry process is exploiting on the linkage value and pseudonym of SCMS, thus maximizing the privacy of the forensic vehicles.

After gaining all accident related linkage values, the police begin to retrieve all V2X messages within the temporal range of the incident based on the time of the incident and all linked values associated with it. Our retrieval mechanism does not need to traverse all the stored V2X messages, nor does it require on-chain conditional queries to find all the messages related to the accident, thus reducing expensive on-chain computations.

Rather, we calculate all potential accident related mapping keys (described before in formula (1)) off chain, and input them to blockchain node (line 10-13 in Algorithm 2). The node then filters partial received keys that definitely do not exist on chain through its Bloom filter (line 14-17 in Algorithm 2). Bloom filter enquiry is more efficient than mapping enquiry, so it helps save evidence retrieval time. Finally, the node enquiries the $MR$ contract to see if filtered keys really exist on chain (line 18-20 in Algorithm 2).

The accident related key is devised not only for searching all IoV messages within the spatial and temporal scope of the accident, but also for enhancing the scalability of evidence retrieval considering the continually growing scale of stored messages. As described in formula (1), the key is the combination of message aligned time, linkage value and message type (a vehicle could send different types messages at a certain time), which can uniquely identify a certain piece of IoV message. As long as accident time and related linkage values are obtained, all potential accident related messages' corresponding keys can be determined, thus finding all accident related messages through traversing and querying these keys. The number of keys to be traversed depends on forensics requirement such as the forensics time range and forensics message type, and has little thing to do with the number of on-chain stored IoV messages, largely reducing the scope of retrieval. The raw IoV message time is millisecond, so for a certain message type from one vehicle in a second, we need inquiry 1,000 times if using millisecond in key, which is actually redundant, for the vehicle message sending frequency is about one time per 0.1 second. Thus, message time in key is aligned based on 0.1 second, which reduces the keys to be calculated and queried while not missing any related messages compared with the former.

In this way, all vehicles within the spatial and temporal scope of the accident are found without revealing their real identities and compromising forward communication anonymity.

**Algorithm 2** Spatio-Temporal Correlation Accident Evidence Retrieval

1: **Input:** accident time & location, timeOffset, messageType
2: **Output:** accidentEvidence[]
3: **Start:**
4: lvs[] ← mapping[accident time & location]
5: for each $lv$ in lvs[]:
6: plvSet1, plvSet2 ← getFromSCS(lv)
7: for each $plv1, plv2$ in plvSet1, plvSet2:
8: $lv \leftarrow plv1 \oplus plv2$
9: add $lv$ to lvSet
10: $(startTime, endTime)$←(accident time - timeOffset, accident time + timeOffset)
11: for each alignedTime based on 0.1s in $(startTime, endTime)$:
12: for each $lv$ in lvSet:
13: MsgID ← Hash( alignedTime ‖ lv ‖ messageType )
14: for each HashFunction in $k$:
15: index ← HashFunction(MsgID) mod $N$
16: if bitArray[index] == 0:
17: break and go back calculate next MsgID
18: query mapping[MsgID] on chain
19: if exist:
20: add MsgID to accidentEvidence[]
21: return accidentEvidence[]
22: **End**

*2) Evidence Verificationl:* After the evidence retrieval algorithm that successively takes off-chain pretreatment, Bloom filtration, and on-chain enquiry, accident-related IoV messages can be located. According to the key MsgID, mapping structure data Evidence can be obtained, which contains the message's millisecond time, position point, message hash, digital signature, and cloud storage address for the message. The above data is then used directly by the on-chain EV contract to automatically validate message integrity and source without human interference. The evidence verification mechanism is shown in Algorithm 3.

Through the call between contracts, EV contract receives structure data Evidence from MR contract and performs evidence verification based on it. First, EV contract fetches complete IoV message and vehicle pseudonym certificate from cloud according to cloud storage address for message. Then, it calculates the hash of complete IoV message and extract pub-

**Algorithm 3** Evidence Verification

```
Input: msgHash, signature, cloudAddr
Start:
cmpltMsg, pseudoCert ← GetFromCloud(cloudAddr)
pk ← ExtractPublicKey(pseudoCert)
calcHash← CalculateHash(cmpltMsg)
if calcHash == msgHash && VerifySignature(msgHash,
signature, pk) == true
    return valid
else
    return invalid
End
```

lic key from vehicle pseudonym certificate. Subsequently, it compares the calculated hash with on-chain message hash and checks the validity of signature with public key, respectively to ensure the message integrity and source non-repudiation. At last, EV contract returns the validation result based on whether both are verified successfully.

## IV. Experiment Evaluation

This section presents a series of experiments conducted to assess the efficiency benefits of our proposed evidence retrieval mechanism in handling IoV large amounts of data. Additionally, we evaluate the performance of our system functions, such as response latency and concurrency, to determine if they meet the requirements of high-speed mobility and real-time large-scale data transmission processing in IoV scenarios.

Utilizing the FISCO BCOS blockchain platform, we construct a consortium blockchain network consisting of four nodes and implement corresponding smart contracts. Every node is furnished with an Intel Xeon Processor with 16 cores, 32GB of DDR4 RAM, 1TB of SSD storage, and a 1Gbps Ethernet network connection. The Bloom filter is instantiated utilizing the Guava package in Java and smart contract is implemented in solidity. We define the communicating data in JSON format with AES128 encrypting and BASE64 encoding.

As detailed in Table 2, we compare the time consumption of three different retrieval algorithms to prove the efficiency of our proposed retrieval algorithm: on-chain querying, off-chain pretreatment + on-chain querying, off-chain pretreatment + bloom filtration + on-chain querying. We conduct experiments in two distinct test dataset conditions: ideal condition and actual condition. In ideal condition, every IoV message has a uniform size. We evaluate the time consumed by the three algorithms as the data storage capacity increases. Comparatively, the size of each IoV message is different under real-world conditions, and we measure the time consumed by the three algorithms as the data tuple size increases.

The experiment results for the two conditions are shown in Fig.3 and Fig.4, respectively. It is clear that performing simple on-chain queries consumes more time than collaborative on-chain and off-chain queries. This is due to the large number of on-chain computations required to traverse all stored IoV information. On the contrary, the second algorithm greatly reduces the number of keys to be searched by using off-chain preprocessing, thus reducing the time required for on-chain queries. Unlike the second algorithm, our proposed algorithm utilizes an efficient Bloom filter to weed out computational keys that are known not to exist. This will further drastically reduce the on-chain query time, ultimately saving about 30% of the retrieval time.

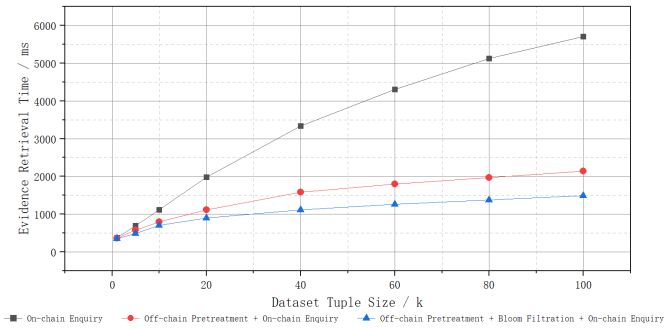


Fig. 3. Evidence Retrieval Time in Ideal Condition

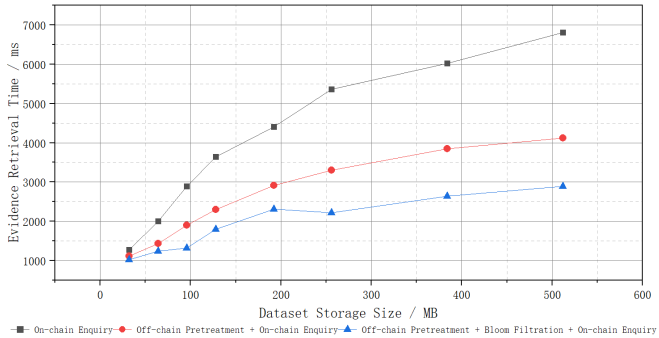


Fig. 4. Evidence Retrieval Time in Actual Condition

As described above, our system has four basic functions for single IoV message: message storage, message querying, sources gathering and evidence verification. To evaluate system basic functions performance, we take multiple experiments (1-20 times) to test their average response time (the time from smart contract receiving the request to processing completion) and transactions per second (TPS), which is the maximum number of concurrent requests that the system can handle stably. JMeter is used as load generator to simulate concurrent requests from the vehicle networking system to smart contracts. Gradually increase the number of concurrent requests, starting from 1,000/second to 10,000/second. Monitor the response time of smart contracts and the status of the blockchain in real time. Record the response time of smart contract processing requests and the system TPS under different concurrency levels.

As shown in Figures 5 and 6, the response times of the above four types of contracts are all within the minute level, and the throughput is greater than 3,000 TPS, which meets the real-time performance requirements. The evidence verification

Table 2. Three Retrieval Algorithms Comparing Experiment

| Retrieval Algorithm | Algorithm Detail | Dataset Size (Tuple) | Dataset Size (MB) |
|---|---|---|---|
| On-chain Querying | Traverse on-chain IoV messages and then select the messages whose time and location fields fall within spatial and temporal scope of accident. | 100000 | 512 |
| Off-chain Pretreatment + On-chain Querying | Calculate the potential keys that may exist within spatial and temporal scope of accident off chain, and then query on chain to see if these keys actually exist. | | |
| Off-chain Pretreatment + Bloom Filtration + On-chain Querying | Calculate the potential keys that may exist within spatial and temporal scope of accident off chain, then use bloom filter to eliminate the keys that definitely do not exist, and finally query on chain to verify if these keys actually exist. | | |

smart contract has the longest response time, which is due to the fact that the evidence verification process involves extracting data from the cloud and performing cryptographic operations such as hashing and digital signature verification. Therefore, this process takes more time to complete.

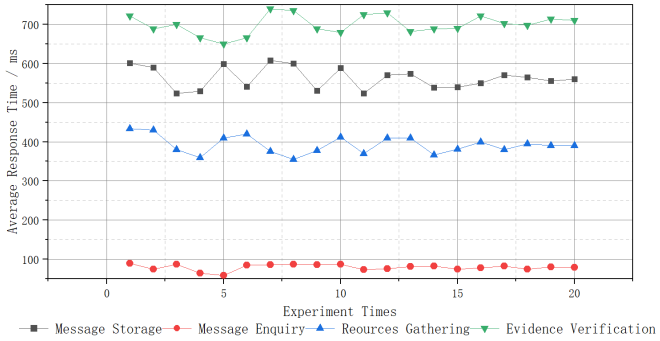


Fig. 5. Average Response Time of Smart Contracts Functions

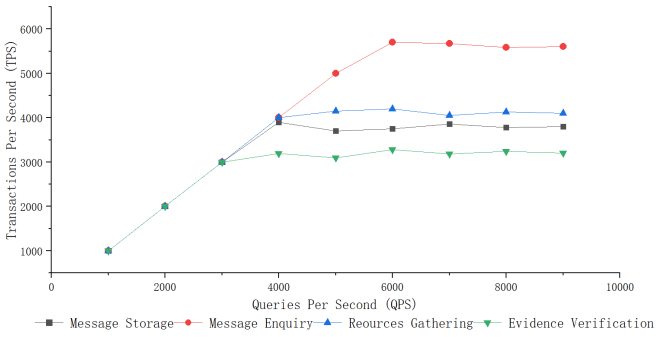


Fig. 6. Transactions Per Second (TPS) of Smart Contracts Functions

## V. Summary

We propose a blockchain-based accident forensics system for smart connected vehicles under the IoV system architecture with SCMS.The system collects V2X communication and sensing information of all vehicles within the spatial and temporal range of the accident to fully restore the whole process of the accident. It also utilizes the link value of the SCMS scheme and the internal security pseudonym certificate mechanism to protect the identity privacy of the forensic vehicles. To reduce the burden of on-chain data processing, we propose a synergistic approach between on-chain data and cloud storage, and design an on-chain data storage structure for V2X communication data. To further improve the efficiency of accident data querying, we propose the technique of combining off-chain preprocessing and on-chain querying, and design a fast evidence retrieval method based on Bloom Filter. The experimental results verify that our proposed accident forensics system for smart connected vehicles meets the application requirements in terms of real-time and scalability.

## References


[1] Mallozzi P, Pelliccione P, Knauss A, et al. Autonomous vehicles: state of the art, future trends, and challenges[J]. Automotive systems and software engineering: State of the art and future trends, 2019: 347-367.

[2] Zheng Z, Xie S, Dai H N, et al. Blockchain challenges and opportunities: A survey[J]. International journal of web and grid services, 2018, 14(4): 352-375.

[3] Cebe M, Erdin E, Akkaya K, Aksu H, Uluagac S. Block4Forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles. IEEE Commun Mag. 2018;56(10):50-57.

[4] Weiwei C, Li C, Xiang G. E-forensics model for internet of vehicles based on blockchain. J Comput Appl.2021;41(7):1989-1995.

[5] Dorri A, Steger M, Kanhere SS, Jurdak R. BlockChain: a distributed solution to automotive security and privacy. arXiv preprint arXiv:1704.00073; 2017.

[6] Oham C, Kanhere SS, Jurdak R, Jha S. A blockchain based liability attribution framework for autonomous vehicles; 2018.

[7] Brecht B, Therriault D, Weimerskirch A, et al. A security credential management system for V2X communications[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(12): 3850-3871.

[8] Zhang G, Pan F, Mao Y, et al. Reaching consensus in the byzantine empire: A comprehensive review of bft consensus algorithms[J]. ACM Computing Surveys, 2022.3871.

[9] Arooj A, Farooq M S, Akram A, et al. Big data processing and analysis in internet of vehicles: architecture, taxonomy, and open research challenges[J]. Archives of Computational Methods in Engineering, 2022, 29(2): 793-829.

[10] Cai H, Xu B, Jiang L, et al. IoT-based big data storage systems in cloud computing: perspectives and challenges[J]. IEEE Internet of Things Journal, 2016, 4(1): 75-87

[11] Bloom B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426.