Journal of Network and Computer Applications 218 (2023) 103716

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

BAC-CRL: Blockchain-Assisted Coded Caching Certificate Revocation List for Authentication in VANETs☆

Junwei Liang [a,b,1], Muhammad Sadiq [b,*], Geng Yang [b], Dongsheng Cheng [b]

[a] State Key Laboratory of Public Big Data, Guizhou University, Guizhou, 550025, China
[b] Shenzhen Institute of Information Technology, Shenzhen 518172, China

ARTICLE INFO

Keywords:
Vehicular Ad Hoc Networks
Certificate Revocation List
BAC-CRL
Blockchain
Multi-Layer Coded Caching

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) are considered the technological upheaval for improving road safety and traffic efficiency by enabling the wireless and network communications among vehicles and infrastructures. Certificate Revocation List (CRL)-based schemes are the most widely used recovery mechanism for VANETs to eliminate the negative effect of security and privacy attacks. However, providing an efficient and low-cost CRL-based scheme for certificate revocation is still a challenging issue since the communication resources must be capable of carrying various applications apart from the security and privacy purposes. Thus, in this paper, a Blockchain-Assisted Coded Caching Certificate Revocation List, called BAC-CRL, is proposed for authentication in VANETs. In BAC-CRL, the blockchain technology is leveraged to unite regional Trust Authorities (TAs) and Road-Side Units (RSUs) into a blockchain network, which uniformly manages and stores multipart CRLs for preventing resource-constrained On-Board Units (OBUs) to be overwhelmed. In addition, a multi-layer coded caching strategy is further designed in BAC-CRL to distribute the minimum bits of CRLs that satisfies the revocation requirements, aiming to reduce the communication overhead and shorten the processing time cost. Moreover, a novel anonymous authentication mechanism is developed utilizing the proposed BAC-CRL to achieve conditional privacy preservation with efficient certificate revocation. Security analysis and extensive simulations demonstrate that the proposed BAC-CRL is secure and computationally efficient for vehicle revocation.

1. Introduction

Both academia and industry consider that Vehicular Ad Hoc Networks (VANETs) is subverting the way of traffic travel today by enabling a wide range of safety-related applications, including prevention of collisions, real-time traffic condition monitoring, blind crossing, etc. (Nayeen Mahi et al., 2022). Many major car manufacturers and telecommunications have teamed up to equip their vehicles with Wireless Access Vehicular Environment (WAVE) from 2015, which indicates the imminent deployment of VANETs in one decade (Liang et al., 2019). WAVE protocols are designed based on IEEE 802.11p standard to provide the basic radio standard for Dedicated Short-Range Communication (DSRC) in VANETs. The vehicles in VANETs can use DSRC to communicate with each other, i.e., vehicle to vehicle (V2V), and with the infrastructure (e.g., roadside units (RSUs)), i.e., vehicle to infrastructure (V2I), to exchange safety information (Liang and Ma, 2020b). Although the deployment of VANETs can provide a variety of benefits, there are a lot of concerns on the issues of privacy and security, especially under the threat of cyber-attacks. Hence, the mechanism to tackle the privacy- and security-related issues in VANETs is necessary and critical (Liang and Ma, 2020a).

Due to the open-medium nature of VANETs, it is vulnerable to a number of attacks, e.g., bogus information attack, impersonation attack, message modification attack, etc. Authentication is considered to be the first line of defense against malicious behaviors in addition to preserving the identity privacy of vehicles in VANETs. Unfortunately, the safety of VANETs would be compromised if authenticated vehicles start to launch security attacks, such as poisoning attacks and inference attacks (Kumar et al., 2022). Thus, certificate revocation schemes should be proposed as the supplement for authentication to eliminate the negative effect of security and privacy attacks from authenticated

☆ This work was supported by the funding of the Foundation of State Key Laboratory of Public Big Data under No. PBD2022-14, Shenzhen Science and Technology Program, China (Grand No. RCBS20221008093252092), the Guangdong Provincial Research Platform and Project, China under No. 2022KQNCX233, and the Science and Technology Ph.D. Research Startup Project under No. Grant SZIIT2022KJ001.

* Corresponding author.
E-mail addresses: jwliang@sziit.edu.cn (J. Liang), sadiq@sziit.edu.cn (M. Sadiq), yangg@sziit.edu.cn (G. Yang), chengds@sziit.edu.cn (D. Cheng).
[1] IEEE Member.

https://doi.org/10.1016/j.jnca.2023.103716
Received 25 November 2022; Received in revised form 13 June 2023; Accepted 24 July 2023
Available online 28 July 2023

attackers. In recent years, many research efforts have been dedicated to design the revocation schemes for authentication in VANETs (Azam et al., 2021; Chuat et al., 2022; Dykstra et al., 2021; Tamper-Evident Technology, 2023; Jan et al., 2022; Chatzigiannis et al., 2021; Larisch et al., 2022; Zulfiqar et al., 2022; Saleem et al., 2022; Ozcelik and Skjellum, 2021; Ge et al., 2022), which mainly include three categories as follows (Azam et al., 2021):

(i) Short-Lived Certificate: A short-lived certificate is identical to a regular certificate, except that the validation period is a short span of time, such as a few days. In Chuat et al. (2022) and Dykstra et al. (2021), the authors propose a self-revoked digital receipt that has a field to show the valid time in order to control the life-cycle of the certificate. The imminent expiration of these certificates will result in their automatic failure and cessation of operation on client systems without the requirement of a revocation process. However, this mechanism is clearly not suited to mobile networks, especially VANETs. In VANETs, vehicles are mandated to frequently communicate with their nearest Trust Authority (TA) in order to renew their certificates, leading to increased communication overhead on shared links and a heightened computational burden on TAs. Moreover, malicious vehicles can still launch security attacks before their certificates reach the expiration time.

(ii) Tamper-Proof Device: Tamper-Proof Device (TPD) is a hardware device embedded in the On-Board Unit (OBU) of a vehicle, which stores all the sensitive information of the vehicle, e.g., secret keys, certificates, driver's real identity or pseudonyms, trajectory, etc. (Tamper-Evident Technology, 2023). Both the cryptography and authentication operations of the vehicle must interact with the TPD to access the security and privacy information. For TPD-based schemes, the revocation is achieved by unicasting a revocation message to the TPD to obsolete the secret keys or certificates (Jan et al., 2022; Chatzigiannis et al., 2021). The problem is that some skilled attackers can intercept the revocation messages sent from TAs or manipulate their TPDs against the revocation of their secret keys or certificates. In this case, other vehicles still trust the malicious vehicles since they have no knowledge about the revocation list.

(iii) CRL-Based Scheme: An implementation of a CRL-based scheme, called X.509 CRL, is proposed in Larisch et al. (2022), which has been further discussed in Zulfiqar et al. (2022) and Saleem et al. (2022). By using the X.509 CRL-based scheme, TAs will update and issue a copy of the latest CRL to all vehicles whenever rogue vehicles are reported and their malicious behaviors are confirmed. A certificate is considered invalid only if its digital identifier is contained in the CRL. The TAs are responsible to distribute the revoked certificate in order to timely update the CRLs among the networks. It has been observed that as the number of revoked vehicles increases, the size of X.509 CRL also grows accordingly, leading to a substantial increase in storage requirements for OBUs. Researchers in Ozcelik and Skjellum (2021) provide a lightweight CRL keychain using the hash chains to reduce the CRL size. Similarly, in Ge et al. (2022), an efficient certificate revocation scheme, i.e., compressed CRL, is designed. The receivers can calculate the revoked pseudonym certificates through the hash chains in a compressed CRL scheme. Unfortunately, the two schemes are exclusive for less mobile and small-scale scenarios as they can only revoke self-generated certificates but not the shuffled or exchanged pseudonym certificates. A promising CRL-based scheme is mentioned in Liang and Ma (2021) for VANETs, in which the blockchain concept is introduced for the distributed maintenance of CRL. Because the malicious vehicles in the CRL-based scheme are excluded from the communication group of RSUs, no CRL distribution is needed. However, RSUs would be incapable of timely verifying the authenticity of the messages from incoming vehicles, particularly as the traffic congestion occurs. This results in the newly arrived vehicles must queue for the group keys and are unable to communicate with other vehicles (Liang and Ma, 2021). More importantly, in sparse scenarios, vehicles can hardly interact with RSUs to ascertain whether neighboring vehicles are revoked or not.

Even though the existing CRL-based schemes have been deployed in VANETs, there are several problems remaining unsolved. (i) With the increasing number of revoked vehicles, CRLs will become too much massive to be stored and managed in OBUs. Moreover, CRL distribution and the updating procedures inevitably lead to considerably large processing time cost. (ii) Due to the highly dynamic nature, encounters are short-lived in VANETs, resulting in the authenticity of vehicles must be ascertained in time. In other words, CRL schemes should have ability to track down whether a vehicle is revoked both timely and accurately. (iii) RSU-assisted CRL schemes are unable to timely determine the authenticity of vehicles in sparse scenarios, and the computational resource of an RSU is easily exhausted for simultaneously conducting cryptography, key management and revocation. Even though the three challenges obviously limit the performance of CRL-based schemes, few efforts have been made to resolve or alleviate them currently.

To address the aforementioned challenges, the major contributions in this paper are as follows:

- A Blockchain-Assisted Coded Caching Revocation List, called BAC-CRL, is proposed for certificate revocation in VANETs. In BAC-CRL, TAs and RSUs in different regions of VANETs are clustered into a blockchain network to uniformly manage and store their CRLs. The proposed blockchain-based CRL is co-maintained by TAs and RSUs to prevent OBUs from overwhelmed with the massive and continuously growing revoked information of CRLs.
- For efficiently distributing CRLs to OBUs that satisfies the revocation requirements, a multi-layer coded caching strategy is further designed in the proposed BAC-CRL. The novel caching strategy can achieve a significant reduction in network load by creating and exploiting coded multicasting opportunities between OBUs with different demands. To the best of our knowledge, we are the first to design the multi-layer coded caching strategy for VANETs to reduce the communication overhead and shorten the processing time cost of CRL distribution.
- Furthermore, an innovative anonymous authentication method is developed that utilizes the proposed BAC-CRL to attain conditional privacy protection with efficient certificate revocation. The proposed BAC-CRL is applicable to the general pseudonym-based authentication mechanisms, considering it has no interference with any authentication procedure.

The remainder of this paper is organized as follows. In Section 2 and Section 3, the proposed CRL scheme, i.e., BAC-CRL, is presented with discussing the blockchain-based CRL and multi-layer coded caching strategy respectively. How an efficient authentication mechanism cooperates with the proposed BAC-CRL is provided in Section 4. Section 5 and Section 6 demonstrates the security analysis, experimental results and their discussions in detail. At last, conclusions are presented in Section 7.

2. Blockchain-based certificate revocation scheme

According to the surveys about authentication (Kumar et al., 2022; Wang et al., 2020), most of current pseudonym-based conditional privacy preserving mechanisms use CRL-based schemes to distribute the revocation information. To guarantee security and privacy, vehicles are supposed to use each pseudonym set for a short duration and frequently switch to a new pseudonym. The US-based Vehicular Communication System (VCS) standard SAE J2735 defines the pseudonym changing in 120 s or 1 km distance traveled, while the EU-based VCS standard ETSI TS 102.867 recommends changing pseudonyms every 5 min (Pre-Standardization Study on Pseudonym Change Management, 2023). Whenever a TA needs to revoke a malicious vehicle, it must add all

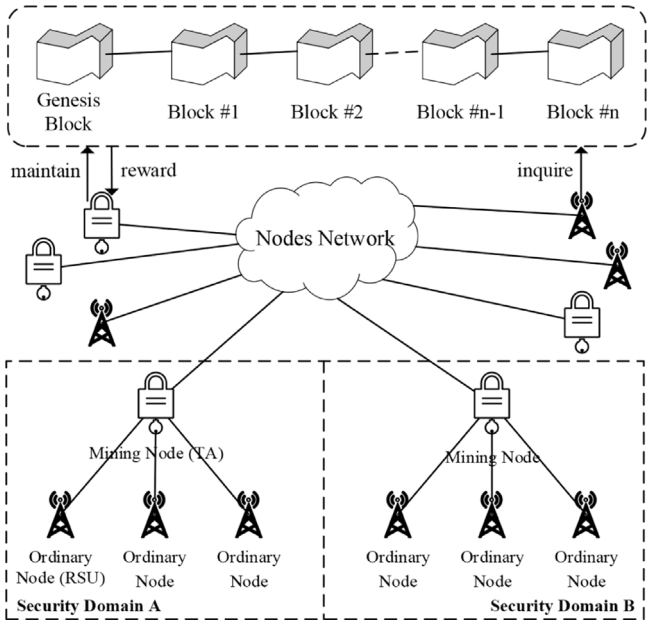

Fig. 1. Framework of blockchain-based CRL.

Transaction Header
Hashed Transaction Header
Transaction Index
Source TA
Destination TA
Digital Signature
Payload: (Revoked Identifiers) $Cipertext = ENC\{Information\}_{PK_{TA}}$

Fig. 2. Transaction format of blockchain-based CRL.

Block Header	
Field	Description
Block Index	Index Value of Block
Previous Block Hash	Hash of Previous Block
Merkle Tree Root	Hash of Merkle Tree Root
Timestamp	Creation Time of Block
Consensus Receipt	Result of Consensus Algoirithm
Block Payload (Transactions)	
Transaction #1, Transaction #2, … , Transaction #n	

Fig. 3. Block format of blockchain-based CRL.

the pseudonym sets to CRLs. As the number of revoked vehicles grows, CRLs will consequentially become tedious. If placing a tedious CRL inside the OBU of a vehicle, it depletes the precious storage resources of the OBU and slows down the time of retrieving revoked vehicles.

For addressing the CRLs management and storage problems mentioned above, we propose a blockchain-based CRL in BAC-CRL, in which the employed blockchain is responsible for the trust, security, transparency, and traceability of the CRLs data sharing across the TAs and RSUs in the separate regions of VANETs. In the following, the framework of the blockchain-based CRL is introduced first. Then, the correlative consensus protocols of the blockchain are discussed with implementing the mining and synchronization processes to validate collective records and achieve the agreement of blockchain transactions.

2.1. Framework of proposed blockchain-based CRL

In the proposed BAC-CRL, the blockchain-based CRL is employed to provide a platform for the internal networking system of all authorized users in VANETs, as shown in Fig. 1. Two portions are included in the blockchain-based CRL. The entire CRL is stored in the blockchain or the upper portion in Fig. 1, while two types of nodes, i.e., ordinary nodes and mining nodes, in the nodes network or the lower portion interact with each other to persistently maintain the consistency, accord, and satisfaction of the blockchain.

(1) Ordinary Node: An ordinary node is an RSU, which is a fixed infrastructure deployed on the roadside. It acts as the bridge between TAs and vehicles by connecting TAs with securing wire links and vehicles over a shared wireless channel. Instead of participating the maintenance of blockchain, the ordinary nodes can only access the blockchain to find out whether a newly-join vehicle is revoked or not, so as to prevent the malicious attacks from previously revoked vehicles.

(2) Mining Node: The mining nodes are comprised of the TAs in different regions, which are responsible for maintaining the entire VANETs. As we know, the whole road transportation system is divided into several geographic regions and each region has a TA. When a vehicle moves from one geographic region to another geographic region, the vehicle is authenticated by the TA of the new region. Worldwide, the certificate authority business is fragmented with national or regional TAs dominating their home market, and a number of multinational companies serve as TAs, such as WebTrust in North America and ETSI in Europe (Certificate Authority, 2023). The mining nodes are fully trusted and infeasible for any opponent to compromise. Unlike the ordinary node, the mining nodes compete with others to ascertain the real identities of malicious vehicles and put their revoked identifiers into the blockchain-based CRL to prevent further damage. After the competition (or mining process), the mining node that firstly mined a block will receive some digital coins as reward, like Bitcoin (Liang and Maode, 2020). Using the digital coins as a medium of exchange, the security information or sensitive data can be traded among the TAs to improve their security level.

The second portion, i.e., the blockchain of the blockchain-based CRL, consists of a series of connected blocks to record the information about validation, connection, and the revoked identifiers of malicious vehicles. The specific structure of the blockchain is provided in Figs. 2 and 3.

Fig. 2 shows the format of each transaction in the block of the blockchain-based CRL, which contains a transaction header and payload. In the transaction header, the index of this transaction gives the position where the transaction locates. The source TA and destination TA are similar to Bitcoin input and output for indicating the addresses of sender and receiver (Sai et al., 2021). Because the transactions are broadcasted among all mining nodes, the digital signature must be used to maintain the authentication, integrity and non-repudiation. In the proposed CRL scheme, the Elliptic Curve Cryptosystem (ECC)-based Cryptographic scheme (ECC, 2023) is employed to issue a digital signature for each transaction. In the payload, the encrypted revoked identifiers of vehicles are listed to eliminate negative effect from malicious vehicles. Each revoked identifier is a long serial number with the size in 6 bytes.

The block is designed for containing all transactions and will be appended into the blockchain after the mining process. The format of a block of the blockchain-based CRL is shown in Fig. 3. The first row shows the block index, which is the sequence number of the whole blockchain. The previous block hash links this block to the last block.

Algorithm 1 Mining process of a TA (PoW)

Thread 1: Block Packaging
repeat
  hear Msgs from RSUs and vehicles
  if New Msg contains malicious report **then**
    verify and trace revoked identifiers
    encapsulate related information in a transaction
    broadcast the transaction to other TAs
  end if
until New block is mined
Thread 2: Power of Work
calculate previous block hash and initiate nonce
while New block and nonce are not found **do**
  try to find the matching nonce by brute force
  if New block is mined **then**
    do Synchronization process
  else if nonce is found **then**
    create a new block and send to other TAs
    append the block to blockchain after consensus
  end if
end while

Any two consecutive blocks are connected as a chain structure to eventually form the blockchain. The Merkle tree root (Merkle Tree, 2023) is deployed for securing the transactions integrity. All transactions in this block are joined into the Merkle tree root, so that any alteration on any transaction would cause a different value of Merkle root value. Similar to the Bitcoin, a timestamp is employed to prevent time tempering. The consensus receipt field includes the digital materials for verifying the consensus algorithm, where the targeted difficulty and nonce are illustrated if the Proof of Work (PoW) algorithm is used for block mining (Li et al., 2017). The payload contains the aforementioned transactions that the block creator randomly allocated.

2.2. Mining and synchronization processes

In the blockchain-based CRL, PoW (Li et al., 2017) algorithm is employed as the mining technique. It requires the mining nodes to solve a hard-mathematical puzzle that is changed frequently and has been agreed by all the nodes. Once a node solves the puzzle, the block is appended into the blockchain. Other mining nodes validate the latest block to make sure that the submitter or winner is not falsifying. If the block is verified by other mining nodes, the winner will be rewarded. The agreement here is based on a majority consensus. Thus, it is difficult to fake unless the attackers can compromise more than 50 percent of the mining nodes.

The summarized procedures of the mining process are shown in Algorithm 1, which runs two threads in parallel to package a new block and solve a PoW problem. According to the algorithm, thread 1 is responsible for block packaging. If a TA hears a malicious report from a message, the suspect behaviors of the report are verified, and the corresponding revoked identifiers will be tracked down. After that, the related information is encapsulated into a transaction before broadcasting to other TAs. Once a new block is mined or received by the TA, the next round mining process or the thread 2 begins. At the beginning, the previous block hash and nonce are calculated and initiated respectively. Then, the brute force approach is adopted to persistently search the matching nonce in the hash function until receiving a new block or the matching nonce is found that satisfices a given targeted difficulty. At the end of this thread, all corresponding parameters are packaged into the new block, which will be appended into the blockchain after the consensus of all participants.

Before executing the mining process, all individual mining nodes have to carry out the synchronization process (or consensus process) to resolve the conflict within its blockchain and make it consistency with other mining nodes. In the synchronization process, two consensus rules, which every legitimate node has to obey for resolving the blockchain forks (Liang and Ma, 2021), are (i) the longest chain rule, and (ii) the error-free rule. The former ensures that all nodes will recognize "the chain with most efforts" as the formal chain, which is typically the longest of the forks. On the other hand, the latter guarantees the trustworthiness of the blockchain. The revocation results must be precise and accurate considering the irreversible characteristics of blockchain. For even a minor label-revision, the consensus admission from all nodes and recalculating a number of hash values are prerequisite, which is extremely expensive and time-consuming.

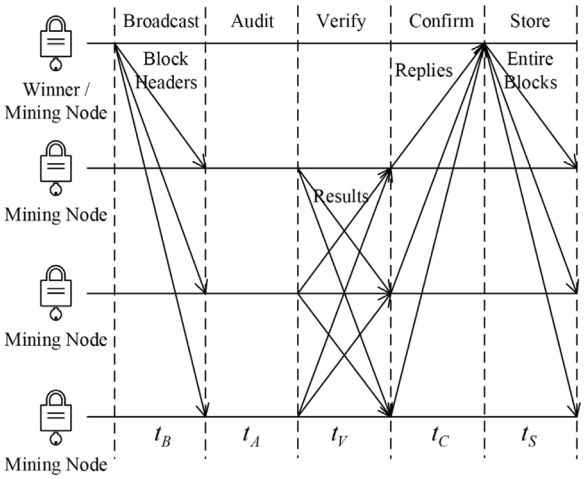

Fig. 4. Synchronization process of blockchain-based CRL.

The synchronization process occurs among the winner of the mining process and other mining nodes as shown in Fig. 4. Once the winner creates a block, it broadcasts the header of the block, which includes the timestamp, previous block hash, matching nonce, etc., to other mining nodes for verification and audit. In the process of audit, these mining nodes audit the block's header while using the longest chain and error-free rules to resolve the blockchain forks. After that, they broadcast their audit results to each other for mutual supervision. After receiving the audit results, each mining node compares its result with others and sends a reply back to the winner node. The reply consists of the node's audit result, comparison result, the records of received audit results and signatures. Next, the winner analyzes the received replies. If more than 50% mining nodes agree on the block, the winner will send the entire block to its peer nodes for storage. At last, the block is appended into the blockchain, and the winner is awarded by some digital coins. If the majority refuses the block, the winner will reexamine the related parameters in header.

According to Fig. 4, the delay of the synchronization process t_{syn} can be calculated as Eq. (1):

$$t_{syn} = t_B + t_A + t_V + t_C + t_S \tag{1}$$

where t_B, t_A, t_V, t_C, and t_S are the delays for the broadcast, audit, verification, confirm and store respectively. The delay of the synchronization primarily comes from t_A and t_V, as others are the propagation delays that depend on network bandwidth, cable materials, etc. Because the hash value is easy to be validated given the previous block hash and provided nonce, t_A and t_V will not be much longer than other delays.

3. Multi-layer coded caching strategy

Beforehand, the foundation of BAC-CRL (i.e., the blockchain-based CRL) has been presented and discussed, which is a co-maintained

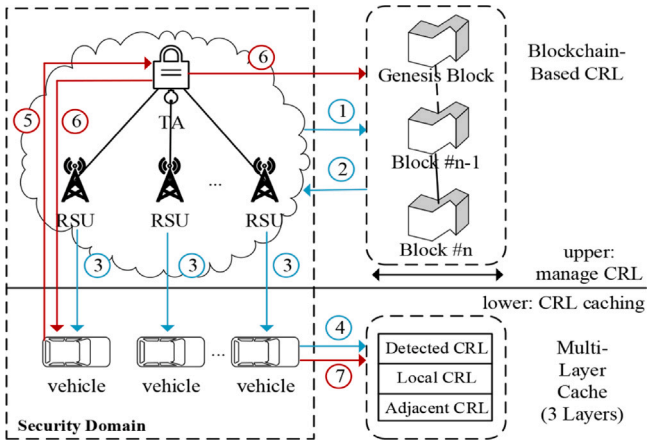

Fig. 5. Multi-layer coded caching strategy: including (a) placement phase (i.e., steps 1-4 with blue arrows), followed by (b) delivery phase (i.e., steps 5-7 with red arrows).

database jointly possessed by the TAs and RSUs in VANETs. Obviously, it is time-consuming and resource-wasteful for the TAs and RSUs to lookup all CRLs and distribute the authenticity of a vehicle every time receiving a require message. Due to encounters are short-lived in VANETs, the authenticity of vehicles must be ascertained timely and accurately. RSU-assisted CRL scheme (Azam et al., 2021) seems to be a promising method by not distributing any group key to revoked vehicles, in which revoked vehicles are excluded from the communication group and no distribution is needed. Unfortunately, the RSU-assisted CRL scheme is unable to timely determine the authenticity of vehicles in sparse scenarios, and the computational resource of an RSU is easily exhausted for simultaneously conducting cryptography, key management and revocation.

To address the issues above, the multi-layer coded caching is further designed for BAC-CRL in this section. It serves as the supplement to mitigate the inherent limitations of blockchain (i.e., scalability and processing speed), which can efficiently distribute CRLs that satisfies revocation requirements and achieve a significant reduction in network load by exploiting coded multicasting opportunities between vehicles with different demands. In the following, we detailly discuss the proposed caching strategy in problem setting, placement & delivery phases, and theoretical results respectively.

3.1. Problem setting

The multi-layer coded caching strategy is shown in Fig. 5. According to the figure, two portions are involved in a security domain of VANETs. In the upper portion, a TA and several RSUs are connected with each other through wired and secure links. The TA and RSUs interact with each other for the consensus and maintenance of the blockchain-based CRL, and both of them can access the blockchain to retrieve the latest CRLs for vehicle revocation. The lower portion is consisted of a number of vehicles that can perform wireless communication with the regional TA and RSUs in the security domain through a shared, wireless and error-free channel. Three caches are hosted in every vehicle for local, adjacent and detected CRLs. Specifically, the caches for local and adjacent CRLs are adopted to timely determine the authenticity of revoked vehicles that are in the local region and from other geographically adjacent regions, and the cache for detected CRL is implemented to blacklist the newly-discovered rogue vehicles while their revoked certificates are waiting for being appended into the blockchain-based CRL.

As shown in Fig. 5, the multi-layer coded caching strategy continuously operates in the two phases, i.e., the placement phase in $t-1$ and the delivery phase in t, represented by blue and red arrows respectively. At the beginning of the placement phase, the blockchain-based CRL is pre-synchronized by the TA and RSUs so that the latest CRL can be accessed to eliminate the negative effect from malicious vehicles (i.e., Step 1 in blue). During the placement phase, whenever an RSU perceive a revoked vehicle in its communication range, the infrastructure adds or updates the revoked identifier into its local CRL with an expire time (i.e., Step 2 in blue). By the end of the phase, the RSU will collect the local CRLs from the geographically adjacent RSUs into its adjacent CRL and broadcast the local and adjacent CRLs to all the vehicles in its communication range (i.e., Step 3 in blue). Once receiving the two CRLs, each vehicle updates its cache memory in local and adjacent CRL layers correspondingly (i.e., Step 4 in blue). In the delivery phase, if a vehicle detects malicious messages from a legitimate vehicle, it reports the malicious messages to the TA for tracing the real identifying information, and then store the revoked identifier into its detected CRL layer with an expire period (i.e., Steps 5–6 in red). The expiration period of items in detected CRL layer is usually set as the update period of the blockchain-based CRL. Once receiving the malicious messages, the TA will validate the messages and broadcast them to other peers for consensus before appending the related information of the newly revoked vehicle into the blockchain (i.e., Step 7 in red).

Assume that there are K_t vehicles $(\{v_i\}, i = 1, \ldots, K_t)$ in time slot $t \in T$ (notice: the time axis is divided into equal time slots). Each vehicle v_i can request a revoked identifier of an encountered vehicle, i.e., $r_t(v_i)$, from a time-varying set N_t of the blockchain-based CRLs with cardinality $N_t > K_t$. The K_t vehicles' requests, collectively denoted by the vector $\boldsymbol{r}_t$, are chosen uniformly at random without replacement from N_t. The XOR operation $\oplus$ can be executed by the TA to efficiently deliver the revoked identifiers to $\{v_i\}$. Each revoked identifier has size F bits, and each vehicle is equipped with a cache memory of size $M_t F$ bits. Given these conditions, we can undertake the analysis of the proposed strategy in time slot t. Recall that each vehicle already has $M_t F/N_t$ of its bits cached locally by the operation of the placement phase in time slot $t-1$. In case where $N_t > K_t$, there are a maximum of K_t different revoked identifiers requested. Since this needs to be done for all N_t revoked identifiers, the normalized rate in the delivery phase is $(1-\frac{M_t}{N_t})K_t$. Furthermore, consider a particular bit in one of revoked identifiers, by symmetry this bit has probability $q = \frac{M_t}{N_t} \in (0, 1]$ of being in the cache of any fixed vehicle. Thus, the expected number of bits of the revoked identifier that are cached at a fixed subset of k out the K_t user is given by $Fq^k(1-q)^{K_t-k}$. Summing over all the derived values yields the normalized peak load $R_t(M_t, N_t, K_t)F$ in time slot t, which can be referred as the peak rate as Eq. (2):

$$R_t(M_t, N_t, K_t) = K_t(1-\frac{M_t}{N_t}) \cdot \frac{N_t}{M_t K_t}(1-(1-\frac{M_t}{N_t})^{K_t}) \tag{2}$$

The goal of the multi-layer coded caching strategy is to minimize the long-term average rate of the caching system while satisfying the memory and reconstruction constraints, which can be modeled by Eq. (3) as follows:

$$\bar{R}^* = \begin{cases} min & \bar{R} \\ s.t. & \bar{R} \triangleq \lim\limits_{T\to\infty} sup \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}(R_t) \end{cases} \tag{3}$$

where R_t, $\bar{R}$ and $\bar{R}^*$ are the peak rate in time slot t, the long-term average rate of the caching strategy and the minimum rate respectively. It should be noted that the rate $\bar{R}$ is the long-term average load $\bar{R}F$ normalized by the identifier size F over the shared channel.

3.2. Placement and delivery phases

3.2.1. Procedures of placement

To minimize the peak load $R_t(M_t, N_t, K_t)F$ over the shared channel during a later delivery phase, the cache memory of each v_i is prefetched in a placement phase or in last time slot $t-1$. The placement phase of an RSU is provided in Algorithm 2. Once a newly arrived v_i is perceived by the RSU, its certificate, i.e., $Cert_{v_i}$, will be extracted from Msg_{v_i}.

LBSI Format	
Petrol Station :	50 m
Speed Breaker:	35 m
Traffic Signal :	40 m
School Zone :	105 m
Accident Zone:	120 m
Local CRL :	$\{L_i\}_l^*$
Adjacent CRL :	$\{L_j\}_a^*$

Fig. 6. Format of location-based safety information (LBSI).

Then, the RSU tries to retrieve the revoked identifier, i.e., L_{v_i}, from the blockchain-based CRL represented by D^* as $L_{v_i} = Retrieval(Cert_{v_i}, D^*)$, which will be presented in next section. If $L_{v_i} \neq \emptyset$, it means v_i has been revoked so that its L_{v_i} must be added to the RSU's local CRL by $\{L_i\}_l^* = \{L_i\}_l \cup L_{v_i}$. By the end of the placement phase, the local CRL of the RSU, i.e., $\{L_i\}_l^*$, will be sent to the geographically adjacent RSUs, while updating its adjacent CRL, i.e., $\{L_j\}_a^*$, after receiving the local CRLs from its adjacent RSUs, i.e., $\{L_i^{(rsu)}\}^*$. At last, the RSU will broadcast the location-based safety information (LBSI) with $\{L_i\}_l^*$ and $\{L_j\}_a^*$ to all authenticated $\{v_i\}$ when they enter its communication range. The LBSI, as shown in Fig. 6, is employed to provide the knowledge to vehicles about the obstacles within its coverage region (Azees et al., 2017).

Algorithm 2 Placement phase of RSU in $t-1$

Input: $\{Msg_{v_i}\}$ % Messages of Newly-join v_i
Output: $\{L_i\}_l^*$, $\{L_j\}_a^*$ % Latest Local & Adjacent CRL
for each Msg_{v_i} in $\{Msg_{v_i}\}$ **do**
  extract $Cert_{v_i}$ from Msg_{v_i}
  $L_{v_i} = Retrieval(Cert_{v_i}, D^*)$
  if $L_{v_i} \neq \emptyset$ **then**
    $\{L_i\}_l^* = \{L_i\}_l \cup L_{v_i}$
  end if
end for
send $\{L_i\}_l^*$ to adjacent RSUs
update as $\{L_j\}_a^* = \{L_j\}_a + \{L_i^{(rsu)}\}^*$
broadcast LBSI with $\{L_i\}_l^*$ and $\{L_j\}_a^*$

It should be noted that we cache the revoked identifiers of the malicious vehicles not only in the local region of an RSU but also in geographically adjacent areas. For the reason, let us consider a situation when a vehicle is leaving from the region A to the Region B, as shown in Fig. 7. Once the newly arrived vehicle enters the new region (Region B), the RSU can perceive its existence after hearing its messages. Then, the RSU will check whether the vehicle is revoked or not by retrieving the revoked identifier from the blockchain-based CRL, and then broadcast the results to all the vehicles in the Region B. However, in the period of authenticity confirmation, a malicious vehicle is still allowed to communicate with other vehicles even though it has been revoked in the past, as its revoked results have not yet been cached to the vehicles in the region B. Hence, the geographically adjacent CRLs, in which the revoked identifiers of incoming vehicles are already cached, must be collected to prevent the attacks launched by the revoked vehicles.

3.2.2. Procedures of delivery

Although all the revoked vehicles have been cached in the local and adjacent CRLs in the placement phase, it is indispensable to cache the newly discovered rogue vehicles that just switch to malicious from legitimate for some selfish intention or just because of sensor malfunction. In worse case, the newly discovered attackers can launch a number of attacks to paralyze VANETs before its identifying information has

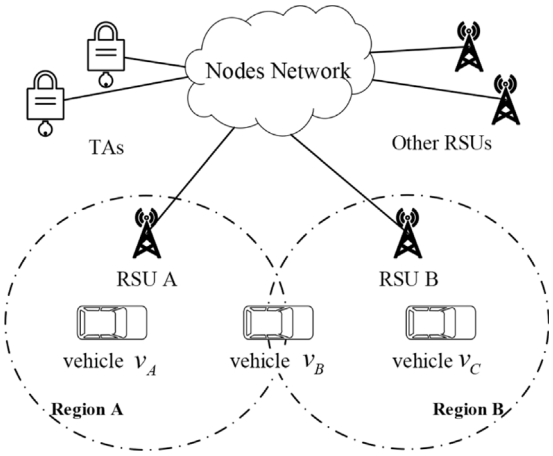

Fig. 7. Situation of a vehicle crossing two regions.

been revoked. Due to the limitation of blockchain technology, the revoked identifiers in a block can only be confirmed after 2 or 3 blocks mined for resolving the conflicts, such as the chain forks in blockchain. Thus, an online delivery phase is crucial to cache the revoked identifiers of newly discovered vehicles into the detected CRL.

Different from the placement phase, which is offline process and executed in $t-1$, the pseudo-code of delivery phase of an OBU in t is provided in Algorithm 3. Firstly, $Cert_{v_i}$ of v_i is extracted from Msg_{v_i}. Then, if v_i is revoked as $Retrieval(Cert_{v_i}, \{L_i\}_l^* \cup \{L_j\}_a^*) \neq \emptyset$, all the messages from this vehicle will be discarded. Otherwise, the OBU will perform the verification functions to ensure the integrity of Msg_{v_i} and the authenticity of v_i, which will be discussed in the next section. Suppose the result of verification is negative, i.e., $Verify(Msg_{v_i}) =$ "$malicous$", the OBU reports the certificate of v_i, i.e., $Cert_{v_i}$, to the nearest TA and wait for the confirmed revoked identifier, i.e., L_{v_i}, from the TA. At last, the confirmed L_{v_i} will be cached into the detected CRL as $\{L_i\}_d^* = \{L_j\}_d \cup L_{v_i}$.

Algorithm 3 Delivery phase of OBU in t

Input: $\{Msg_{v_i}\}$, L_{v_i} % Message & Identifier of v_i
Output: $\{Cert_{v_i}\}$ % Certificate Generated by v_i
for each Msg_{v_i} in $\{Msg_{v_i}\}$ **do**
  extract $Cert_{v_i}$ from Msg_{v_i}
  if $Retrieval(Cert_{v_i}, \{L_i\}_l^* \cup \{L_j\}_a^*)$ **then**
    discard all Msg_{v_i} from v_i
  else
    $\varepsilon = Verify(Msg_{v_i})$
    if $\varepsilon =$ "$malicious$" **then**
      report $Cert_{v_i}$ to nearest TA
      receive confirmed L_{v_i} from TA
      $\{L_k\}_d^* = \{L_k\}_d \cup L_{v_i}$
    end if
  end if
end for

Example. Consider an example of the caching problem with $N_t = 3$ revoked identifiers in demand, i.e., $\{L_\emptyset, L_k, L_{k+1}\}$, three vehicles $(\{v_i\}, i = A, B, C)$ in time slot t, and each vehicle hosts a detected cache of size $M_t F$. In the placement phase, v_B and v_C cached L_k and L_{k+1} respectively, as shown in Fig. 8. In the delivery phase, suppose that:

revoked identifier in demand: $\mathbf{r}_t = \{L_\emptyset, L_{k+1}, L_k\}$
dataload in coded caching: $L_\emptyset, L_k \oplus L_{k+1}$ (size $2F$ sent)
dataload in uncoded caching: $L_\emptyset, L_{k+1}, L_k$ (size $3F$ sent)

In the multi-layer coded caching strategy, it can be observed that v_B and v_C stored L_k and L_{k+1} in their caches. From the output $L_{k+1} \oplus L_k$ of the shared channel, v_B and v_C can easily recover their required identifiers L_{k+1} and L_k. In other words, by using the contents of their caches and the outputs of the share channel, the coded caching strategy can reduce the peak load $R_t(M_t, N_t, K_t)F$ from $3F$ to $2F$ by applying element-wise XOR operation and improve the average peak rate $\bar{R}$ while satisfying all requirement and reconstruction constraints.

3.3. Theoretical results

For better understanding, we firstly provide the simplest system with $N_t = 3$ revoked identifiers in demand and $K_t = 2$ vehicles in time slot t. If a vehicle maintains a list of identifiers with $M_t = \rho N_t$, let $\rho = 1/3$ so that each vehicle caches one third of the bits as $M_t = \frac{1}{3} \times N_t = 1$ revoked identifier. We assume that the two vehicles initially cache the revoked identifier $\{L_k\}$ and $\{L_{k+1}\}$ respectively.

- $t = 1$: The set of revoked identifiers of the TA is $N_1 = \{L_\emptyset, L_k, L_{k+1}\}$. Assume a vehicle requests $r_1(v_1) = \{L_\emptyset, L_{k+1}\}$ and another one requests $r_1(v_2) = \{L_\emptyset, L_k\}$. In the delivery process, the TA sends $L_\emptyset$ and $L_{k+1} \oplus L_k$ to the two vehicles. Assuming that each of these identifiers has close to expected size, the result of $R_1(M_1, N_1, K_1)$ can be calculated by Eq. (2) as:

$$R_1(M_1, N_1, K_1) = 2(1 - \frac{1}{3}) \cdot \frac{3}{2}(1 - (1 - \frac{1}{3})^2) = \frac{10}{9}$$

- $t = 2$: When $t = 2$, v_1 and v_2 cache the revoked identifiers $\{L_{k+1}\}$ and $\{L_k\}$, and the set of the revoked identifiers in demand is changed to $N_2 = \{L_k, L_{k+1}\}$. Other conditions remain the same. The TA will broadcast $L_k \oplus L_{k+1}$ to the two vehicles once hearing the requests $r_2(v_1) = \{L_k\}$ and $r_2(v_2) = \{L_{k+1}\}$. Thus, the result of $R_2(M_2, N_2, K_2)$ are as follows:

$$R_2(M_2, N_2, K_2) = 2(1 - \frac{1}{2}) \cdot \frac{2}{2}(1 - (1 - \frac{1}{2})^2) = \frac{3}{4}$$

- $t = 3$: If $\{L_{k+1}, L_k\}$ is cached into the two vehicles v_1 and v_2, and the set of the revoked identifiers remains the same as $N_3 = \{L_{k+1}, L_k\}$, it is easily to derive that $R_3(M_3, N_3, K_3) = 0$.

The performance of a system with and without caching strategies are shown in Fig. 9. In time slot t, the system contains $N_t = 1000$ revoked identifiers in demand, $K_t = 30$ vehicles, and arrival probability $\rho = 0.1$. According to the figure, the proposed multi-layer coded caching strategy provides significant gain over the uncoded caching strategy regardless of the cache memory size. Comparing uncoded and coded caching, it is obvious that the performances of the two strategies are similar only when M_t is very close to the number of N_t. For all other values, the coded caching strategy significantly outperforms uncoded caching one, which means that the extra gain of the multi-layer coded caching strategy benefits from the coded delivery except for the large value of M_t.

4. Authentication with proposed BAC-CRL

In this paper, we do not confine a special kind of authentication mechanism for the proposed BAC-CRL. In other words, BAC-CRL is applicable to the general pseudonym-based authentication mechanisms, considering there is no interference to any authentication process during the operation of BAC-CRL. In this section, we employ an efficient anonymous authentication mechanism, namely EAAP (Azees et al., 2017), to provide anonymous authentication for VANETs, in which the bilinear pairing technique (Bilinear Pairing Cryptography, 2023) is considered as the basic concept. Although EAAP can efficiently trace the vehicles that abuse VANETs, there is not any certificate revocation scheme to revoke the privacy of misbehaving vehicles. Hence, the proposed BAC-CRL is implemented with EAAP to realize conditional privacy preservation in a computationally efficient way.

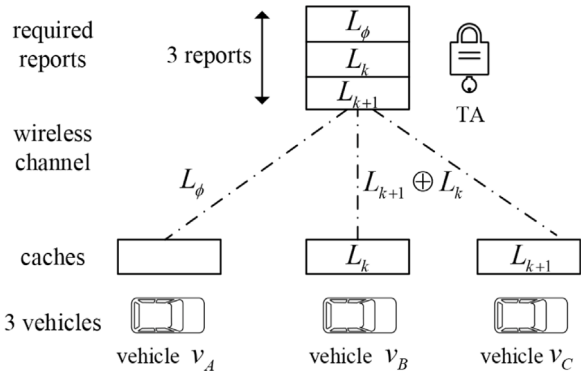

Fig. 8. Example of multi-layer coded caching strategy.

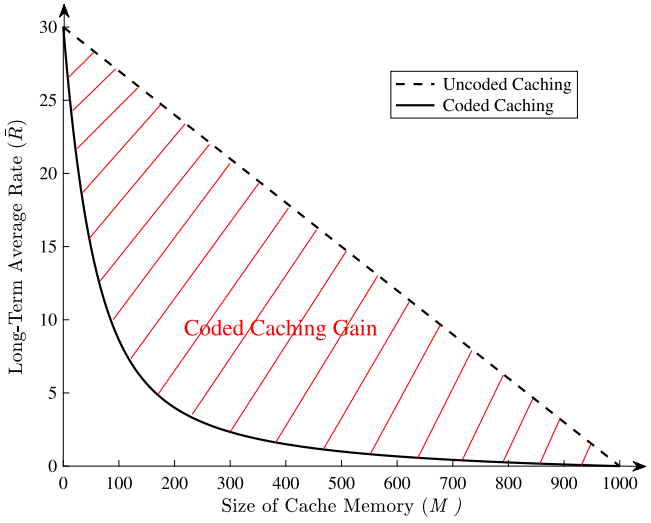

Fig. 9. Performance of system of caching strategies (Pedarsani et al., 2015).

4.1. System initialization

Based on the bilinear parameters (G_1, G_2, G_T, e, q), a TA can generate the system parameters. The TA first selects two random numbers $a, b \in Z_q^*$ as the master secret keys and computes $A_1 = g_1^a$ and $B_1 = g_1^b$. The TA also selects a secure cryptographic hash function $H : \{0,1\}^* \rightarrow Z_q^*$. Finally, the TA publishes the system parameters as $param = (q, e, g_1, g_2, G_1, G_2, G_T, A_1, B_1, H)$.

4.2. Registration and key generation

Before getting involved in the VANETs communication, registration and key generation are essential for both vehicles and RSUs to become a legitimate user. In the registration process, each user of vehicles or RSUs, i.e., u_i, is required to submit the required information to a TA, such as name, address, mail, contact number, etc. Once u_i has completed the registration process, the TA starts to generate the necessary keys for u_i. In the key generation process, the TA first generate the real identity, i.e., RID_{u_i}, for u_i. After that, the TA generates dummy identities, i.e., $\boldsymbol{DID}_{u_i} = \{DID_{u_{i\_j}}\}$, for u_i. To generate $DID_{u_{i\_j}}$, the TA chooses a revoked identifier L_j ($L_j \in Z_q^*$) and computes $DID_{u_{i\_j}} = g_1^{L_j + a} \ mod \ q$. Next, the TA randomly selects an orientation number

$n_i \in Z_q^*$ and computes a locator $U_i = g_1^{\frac{1}{n_i+a+b}}$. Finally, the TA stores $(RID_{u_i}, \boldsymbol{DID}_{u_i}, U_i^b)$ in its tracking list database and return the authorization key $AK = (\boldsymbol{DID}_{u_i}, U_i, \boldsymbol{E}_{u_i} = \{E_{u_{i\_j}}\})$, where $E_{u_{i\_j}} = g_1^{-L_j} \ mod \ q$, to u_i through a smart card directly in the offline mode to avoid various kind of online attacks. Once u_i get the AK from the TA through a smart card, it is stored in a secure environment such as TPD by the user. After the completion of successful registration, u_i can communicate with other vehicles or infrastructures by using the registered credentials.

4.3. Anonymous certificate generation

When u_i is involved in VANETs, it uses the AK and conducts the following steps to generate the required anonymous certificates.

(1) u_i first selects some random numbers $d_1, d_2, \ldots, d_l \in Z_q^*$, $l \leq q$, as the temporary short time private keys and computes the corresponding temporary short time public key $Y_k = g_2^{d_k}$ for $k = 1, 2, \ldots, l$.

(2) For each temporary short time public key Y_k, u_i computes the required parameters for the anonymous short time certificate $Cert_k$ as follows:

First, u_i randomly selects $\mu, k_1, k_2 \in Z_q^*$ and computes γ_1, γ_2 and λ, λ_1, λ_2 as Eqs. (4)–(8):

$$\gamma_1 = B_1^{\mu} \tag{4}$$

$$\gamma_2 = U_i \cdot A_1^{\mu} \tag{5}$$

$$\lambda = (\mu + d_k) \ mod \ q \tag{6}$$

$$\lambda_1 = \gamma_1^{\mu+k_1} \tag{7}$$

$$\lambda_2 = \frac{\gamma_1^{\mu+k_1}}{\gamma_2^{\mu+k_2}} \tag{8}$$

After computing the values γ_1, γ_2 and λ, λ_1, λ_2, u_i is able to compute the challenger $c = H(DID_{u_{i\_j}} \| A_1 \| B_1 \| E_{u_{i\_j}} \| \gamma_1 \| \gamma_2 \| Y_k \| \lambda_1 \| \lambda_2)$, and δ_1, δ_2 as Eqs. (9)–(10):

$$\delta_1 = (d_k - k_1) \ mod \ q \tag{9}$$

$$\delta_2 = (d_k - k_2) \ mod \ q \tag{10}$$

(3) Finally, u_i can generate the anonymous certificate as $Cert_k = \{Y_k \| E_{u_{i\_j}} \| DID_{u_{i\_j}} \| \gamma_1 \| \gamma_2 \| c \| \lambda \| \delta_1 \| \delta_2\}$. When u_i enters into a new geographic region, u_i can be authenticated by the TA in the new region by using the public value of the TA in the registered region. For example, when u_i meets a new TA in a new region, it sends an anonymous message $Msg = (M \| Sig \| Y_k \ \| \ Cert_k)$ to the new TA, where $Cert_k = \{Y_k \| E_{u_{i\_j}} \| DID_{u_{i\_j}} \| \gamma_1 \| \gamma_2 \| c \| \lambda \| \delta_1 \| \delta_2\}$ is the anonymous certificate. From the certificate, the new TA takes two parameters, i.e., $E_{u_{i\_j}}$ and $DID_{u_{i\_j}}$, and computes $N_i = E_{u_{i\_j}} \times DID_{u_{i\_j}}$, which should be equal to the public parameter A_1 of the TA in registered region. If it holds, the new TA considers that the u_i is a registered and authenticated user.

4.4. Signature generation

In order to authenticate and preserve the integrity of a message M, u_i produces a short time anonymous signature $Sig = g_1^{\frac{1}{d_k+H(M)}}$ using the short time anonymous key and broadcasts the anonymous message $Msg = (M \| Sig \| Y_k \ \| \ Cert_k)$.

4.5. Identity and message verification

Given $Msg = (M \| Sig \| Y_k \ \| \ Cert_k)$ from the sender u_i, the receiver performs the following steps to verify the validity of Msg:

(1) At first, the receiver searches for L_j that satisfies $E_{u_{i\_j}} = g_1^{-L_j} \ mod \ q$ until reaching the maximum iteration. If L_j is not found, it means that u_i has not been revoked for committing any malicious behavior.

(2) Secondly, the receiver needs to calculate the prerequisite parameters N_i as $N_i = E_{u_{i\_j}} \times DID_{u_{i\_j}}$ and λ_1', λ_2' by Eqs. (11)–(12) as follows:

$$\lambda_1' = \frac{\gamma_1^{\lambda}}{\gamma_1^{\delta_1}} \tag{11}$$

$$\lambda_2' = \frac{\gamma_1^{\lambda} \cdot \gamma_2^{\delta_1}}{\gamma_1^{\delta_1} \cdot \gamma_2^{\lambda}} \tag{12}$$

(3) Thirdly, the receiver computes $c' = H(DID_{u_{i\_j}} \| N_i \| B_1 \| E_{u_{i\_j}} \| \gamma_1 \| \gamma_2 \| Y_k \| \lambda_1' \| \lambda_2')$ and check whether $c = c'$. If it holds, the receiver authenticates sender u_i, and hence it accepts the public key and the anonymous certificate $\{Y_k \ \| \ Cert_k\}$. If this condition is not satisfied, then the message is discarded by the receiver. The receivers can also verify the dummy identity of the sending user to ensure that authenticated users are allowed to send messages. In Appendix, the full proofs of $\lambda_1' = \lambda_1$, $\lambda_2' = \lambda_2$ and $N_i = A_i$ are demonstrated in detail.

(4) Fourthly, once the challenger c' has been verified, the receiver verifies the integrity of the message by checking Eq. (13) as follows:

$$\begin{aligned} & e(Sig, \ Y_k \cdot g_2^{H(M)}) \\ &= e(g_1^{\frac{1}{d_k+H(M)}}, \ g_2^{d_k} \cdot g_2^{H(M)}) \\ &= e(g_1^{\frac{1}{d_k+H(M)}}, \ g_2^{d_k+H(M)}) \\ &= e(g_1, \ g_2)^{\frac{1}{d_k+H(M)} \cdot (d_k+H(M))} \\ &= e(g_1, \ g_2) \end{aligned} \tag{13}$$

If the equation is satisfied, then the message M will be accepted by the receivers. Other, it will be rejected.

4.6. Tracking and revocation

If u_i launches malicious attacks, the anonymous certificate generated by the user u_i, i.e., $Cert_k = \{Y_k \| E_{u_{i\_j}} \| DID_{u_{i\_j}} \| \gamma_1 \| \gamma_2 \| c \| \lambda \| \delta_1 \| \delta_2\}$, will be reported to the nearest TA. The TA can track down its real identity by Eq. (14) as:

$$\frac{\gamma_2^b}{\gamma_1^a} = \frac{(U_i \cdot A_1^{\mu})^b}{(B_1^{\mu})^a} = \frac{U_i^b \cdot A_1^{\mu b}}{B_1^{\mu a}} = \frac{U_i^b \cdot g_1^{\mu ab}}{g_1^{\mu ab}} = U_i^b \tag{14}$$

From this, the TA can effectively trace the real identity RID_{u_i} of the user u_i by looking up the U_i^b in its tracking list. After tracking U_i, the TA can revoke u_i by putting its all revoked identifiers $\{L_j\}$ into the blockchain-based CRL of BAC-CRL to prevent u_i from causing any further damage.

5. Security analysis

Here, the security analysis is briefly listed with respect to data integrity, source authentication, privacy preserving and conditional privacy preservation, as follows:

(1) Data Integrity: Regard to the integrity of BAC-CRL, EC-DSA (ECC, 2023) is used to digitally sign the transactions to ensure no adversary can forge a digital signature of a node under no prior knowledge about the private key of the node. For instance, when an attacker launches Preimage attack on cryptographic hash function only given $Hash_* =$

$SHA256(nonce)$, it needs to use a collision search on elliptic curve to find a value $nonce'$ that satisfies $SHA256(nonce) = SHA256(nonce')$. For such collision with time complexity $O(2^{\frac{n}{2}})$, $Hash_*$ with output length of 256 bits would take $O(2^{128})$ times, which makes it computationally infeasible with extremely low success probability. As for the message integrity, a user of a vehicle or an RSU broadcasts an anonymous message as $Msg = (M \| Sig \| Y_k \| Cert_k)$, in which there is a chance for an external attacker to change the content of the message M to M'. To guarantee the integrity of Msg, the attacker must generate the corresponding $Sig' = g_1^{\frac{1}{d_k + H(M')}}$ for M', where d_k is the temporary short time private key. However, d_k is only known by the particular vehicle and is changed periodically. Therefore, even if d_k is found, it will not be possible to follow the further communication.

(2) Source Authentication: Except for the TAs, all users of the vehicles and RSUs have to generate an anonymous certificate, i.e., $Cert_k = \{Y_k \| E_{u_{i\_j}} \| DID_{u_{i\_j}} \| \gamma_1 \| \gamma_2 \| c \| \lambda \| \delta_1 \| \delta_2\}$, using the given secret parameter U_i. In order to compute the U_i owned by a legitimate user u_i, an adversary has to find γ_2 in $Cert_k$. Here, γ_2 is calculated as $\gamma_2 = U_i \cdot A_1^{\mu}$ and μ is chosen randomly by u_i, so that the value of γ_2 is also random. Therefore, the complexity of finding the values of U_i and μ from the set of k temporary random keys is $O(2^k - 1)$, and it is infeasible for the adversary to pinpoint the exact U_i and μ to break $Cert_k$ within a stipulated time.

(3) Privacy Preserving: Since the dummy identities, i.e., $DID_{u_i} = \{DID_{u_{i\_j}}\}$, is adopted in an anonymous certificate $Cert_k$ for u_i, the message attached with Sig and $Cert_k$ does not reveal the real identity of u_i for attackers. Furthermore, Sig and $Cert_k$ are generated by using the short life d_k and a randomly chosen dummy identity of $\boldsymbol{DID}_{u_i}$, which means these parameters are time-limited and will be changed irregularly and periodically. Hence, zero knowledge is revealed to attackers in our scheme, and the anonymous of the vehicles and RSUs is preserved.

(4) Conditional Privacy Preservation: In the proposed scheme, the TAs are able to track down the real identity of any u_i (RID_{u_i}) that is stored and managed in the TAs. For instance, the anonymous certificate of a rogue u_i, i.e., $Cert_k$, will be reported to its nearest TA when there is a disputation about u_i's messages. Then, RID_{u_i} can be traced by the TA corresponding to the tracking locator U_i^b calculated by $\frac{\gamma_2^b}{\gamma_1^a} = \frac{(U_i \cdot A_1^{\mu})^b}{(B_1^{\mu})^a} = \frac{U_i^b \cdot A_1^{\mu b}}{B_1^{\mu a}} = \frac{U_i^b \cdot g_1^{\mu a b}}{g_1^{\mu a b}} = U_i^b$. After that, the TA can revoke u_i by putting its all revoked identifiers $\{L_j\}$ into the blockchain-based CRL of BAC-CRL to prevent u_i from launching further attacks and executing malicious behaviors.

6. Performance evaluation

6.1. Simulation setup

Our simulation runs based on Python 3.6 with the related Python libraries and Application Program Interfaces (APIs), i.e., python-blockchain, libbitcoin toolkit, PyBitmessage API and Python Paring-Based Cryptography (PyPBC) (Aitzhan and Svetinovic, 2016; Python Paring-Based Cryptography (PyPBC), 2023). Python-bitcoinlib written by Daniel van Flymen is to provide the fundamental technology of blockchain, including the block structure and nodes network. Libbitcoin toolkit as a successor of speslimo SX is a set of cross platform C++ libraries for building Bitcoin applications, which consists of several libraries, and most of which depends on the foundational libbitcoin library. PyBitmessage API is a set of APIs to communicate with the mining nodes by using XML-RPC and OpenSSL. PyPBC created by Geremy Condra can perform the hash operation, exponential operation, point multiplication and paring operation, in which the Type-A curve is used with the default parameters.

Table 1
Simulation parameters.

Parameter type	Metric value
Transmission range	500 m
Wireless protocol	802.11 p
Max vehicle speed	100 km/h
Vehicle arrival interval	1 sec
Beacon interval	Every 1 sec
Number of mining nodes	10
Hash function	SHA256
Difficulty of mining	3
Method of encryption	ECC (2023)

The scenario is simulated with the parameters as shown in Table 1. In the simulations, each vehicle can communicate with other vehicles within 500 m transmission range based on communicated protocol 802.11p. To avoid generating too much data in one simulation, the vehicle inter-arrival interval is set as 1 sec, while the transmission interval or a beacon interval is set as 1 sec. In the nodes network of BAC-CRL, there are 10 mining nodes competing with each other for the digital coin or reward. After the mining and synchronization processes, every node can hold their own results of the CRL distribution and target retrieval, and the average result will be calculated as the performance of BAC-CRL. Moreover, the Hash Function, i.e., SHA256, is used to validate the effort of mining nodes and to build up the connection between two consecutive blocks. To maintain efficiency and security, the difficulty of mining is set to 3, which means nonce must have '000' in the front to solve the Hash puzzle. ECC-based cryptographic scheme (ECC, 2023) is employed for crypto-operation. Compared to other types of asymmetric cryptography such as Rivest-Shamir-Adleman (RSA), ECC scheme requires shorter key length while providing the same security level (ECC, 2023).

6.2. Certificate revocation list size

In this subsection, three representatives of certificate revocation scheme in existing literature, i.e., X.509 CRL (Larisch et al., 2022), Optimized X.509 CRL (Zulfiqar et al., 2022) and bc-CRL (Liang and Ma, 2021), are compared with the proposed BAC-CRL. X.509 CRL defines that each entry in the revoked certificate field consists of 39 bytes, such as a long serial number of a revoked certificate, revocation date and reason for 6 bytes, 13 bytes and 12 bytes, respectively. The mandatory fixed field in X.509 CRL occupies 400 bytes (Zulfiqar et al., 2022). The optimized X.509 CRL is designed to remove the size of mandatory fixed fields in X.509 CRL, so that only 39 bytes are needed for each revocation element. As for bc-CRL and the proposed BAC-CRL, the certificate index (Liang and Ma, 2021) and revoked identifier both in the size of 6 bytes are used to compress the CRL size. Therefore, the total size of X.509 CRL, Optimized X.509 CRL, bc-CRL and BAC-CRL can be calculated as follows, where n_i is the number of revoked identifiers, and p_a is the arrival probability of a revoked vehicle within the communication range of an RSU.

Among TAs or RSUs:
X.509 CRL size = $(400 + n_i \times 39)$ bytes
Optimized X.509 CRL size = $n_i \times 39$ bytes
bc-CRL size = $n_i \times 6$ bytes
BAC-CRL size = $n_i \times 6$ bytes

Among Vehicles:
X.509 CRL size = $(400 + n_i \times 39)$ bytes
Optimized X.509 CRL size = $n_i \times 39$ bytes
bc-CRL size = 0 bytes
BAC-CRL size = $6 \times \sum_i^{n_i} \binom{n_i}{i} i p_a{}^i (1 - p_a)^{n_i - i}$ bytes

The comparison among the size of X.509 CRL, Optimized X.509 CRL, bc-CRL and BAC-CRL is shown in Fig. 10. According to the figure, the Optimized X.509 CRL achieves minor improvement comparing to

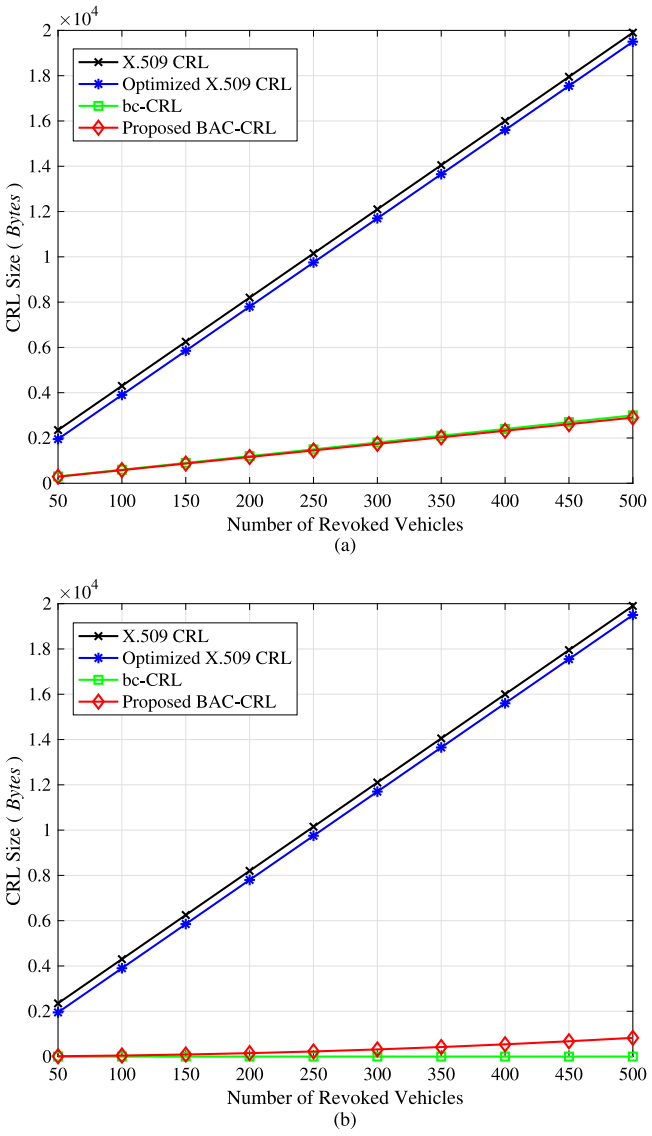

Fig. 10. CRL Size among vehicles in terms of vehicle number: (a) CRL size among TAs and RSUs; (b) CRL size among vehicles.

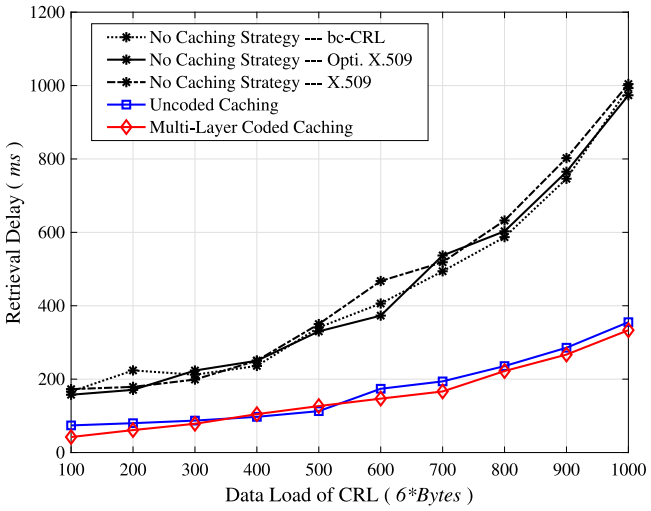

Fig. 11. Delay of target retrieval in different CRL Schemes.

Table 2
Comparison of overheads among CRL schemes.

Measurement	Type of CRL scheme	Metric value
Number of handshake	X.509 CRL	4 Handshake
	Optimized X.509 CRL	4 Handshake
	bc-CRL	2 Handshake
	Proposed BAC-CRL	1 Handshake
Average processing time	X.509 CRL	12.52924 ms
	Optimized X.509 CRL	12.52924 ms
	bc-CRL	6.26462 ms
	Proposed BAC-CRL	3.13231 ms

Table 3
Average cryptography processing time.

Cryptography scheme	Processing time
EC-IES Encryption	0.59157 ms
EC-IES Decryption	0.77922 ms
EC-DSA Signing	0.59595 ms
EC-DSA Verification	1.16557 ms

X.509 CRL as it deletes the mandatory fields in the CRL header among both infrastructures and vehicles. By introducing a compact certificate index and a revoked identifier, bc-CRL and BAC-CRL can significantly compress the CRL size among TAs and RSUs comparing to the two X.509 CRL schemes. Furthermore, by using multi-layer coded caching, only the revoked identifiers of appeared malicious vehicles are distributed and cached to vehicles, resulting in the great decrease of CRL redundancy on the vehicle level. Even though the CRL size of bc-CRL among vehicles is slightly less than that of BAC-CRL, it relies too heavily on RSUs to exclude revoked vehicles, leading to the RSUs in bc-CRL are easily overwhelmed for simultaneously conducting cryptography, key management and revocation. Therefore, we can claim that our proposed BAC-CRL can reach a significant lower revocation list size compared with other existing CRL schemes.

6.3. Overheads of CRL schemes

Here, the comparison of overheads among X.509 CRL, Optimized X.509 CRL, bc-CRL and BAC-CRL is illustrated in Table 2. In the two X.509 CRL schemes, whenever a vehicle detects a malicious behavior, a report is sent to the nearest TA through the connected RSU. After verifying the report, the TA needs to send back the updated CRL to the RSUs in its security domain. Finally, the RSUs are responsible for distributing the latest CRL to all the vehicles within their regions. Thus, 4 times of handshake is needed for the CRL distribution of X.509 CRL and Optimized X.509 CRL schemes. Unlike the X.509 CRL schemes, bc-CRL embeds a malicious report within the other service message which is sent from a Public Key Infrastructure (PKI) to an RSU relayed by a TA. Hence, 2 times of handshake has to be taken for a CRL distribution in bc-CRL. In BAC-CRL, the malicious vehicles that have been revoked in any TA will be synchronized with using the blockchain technology, and then all RSUs can access the latest CRL and distribute the perceived revoked identifiers to all connected vehicles. In general, only 1 time of handshake is needed, except for the security domain that firstly discovers the revoked vehicles (4 times of handshake needed).

Based on the message handshake procedures of X.509 CRL, Optimized X.509 CRL, bc-CRL and BAC-CRL, each message exchange should involve ciphertext encryption and decryption, signature verification and signature generation. The average cryptography processing time is as shown in Table 3. Therefore, overall costs 12.52924 ms, including 4 times of encryption, signing, verification and decryption, is needed for both two X.509 CRL schemes, and a half of that efforts, i.e., 6.26462 ms, must be taken for bc-CRL scheme, while BAC-CRL only requires a quarter of that efforts, i.e., 3.13231 ms, as shown in Table 2.

Table 4
Important variables and parameters used in BAC-CRL.

Notations	Description
t_{syn}	$t_{syn} = t_B + t_A + t_V + t_C + t_S$ is the total delay of synchronization, in which t_A and t_V are critical.
(K_t, N_t, M_t)	The number of vehicles, revoked identifiers in demand and the size of cache memory in time slot t.
$\bar{R}^*$	$\bar{R}^* = min(\bar{R})$ is the mean minimum of caching system in t, which is subject to $\bar{R} \triangleq lim\ sup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}(R_t)$.
$\{L_i\}_l^*, \{L_j\}_a^*$	The two caches deployed both in vehicles and RSUs to store vehicles' revoked identifiers, i.e., $L_{i\&j}$.
$\{L_k\}_d^*$	The cache only deployed in vehicles to record L_k of the malicious vehicle that is newly reported to a TA.
$L_k \oplus L_{k+1}$	$L_k \oplus L_{k+1}$ is the combined output of L_k and L_{k+1} by applying the element-wise XOR operation.
a, b	The master secret keys generated by a TA, and will be computed as $A_1 = g_1^a$ and $B_1 = g_1^b$ for validation.
$param$	$param = (q, e, g_1, g_2, G_1, G_2, G_T, A_1, B_1, H)$ are the system parameters based on bilinear parameters (Azees et al., 2017).
$\boldsymbol{DID}_{u_i}$	$\boldsymbol{DID}_{u_i} = \{DID_{u_{i\_j}}\}$ is the set of dummy identities of user u_i, where $DID_{u_{i\_j}} = g_1^{L_j+a}\ mod\ q$.
U_i	$U_i = g_1^{\frac{1}{n_i+a+b}}$ is the locator in each TA's tracking list database, where $n_i \in Z_q^*$ is an orientation number.
$\boldsymbol{E}_{u_i}$	$\boldsymbol{E}_{u_i} = \{E_{u_{i\_j}}\}$ is the carrier of L_j as $E_{u_{i\_j}} = g_1^{-L_j}\ mod\ q$, which is only used in the authentication.
γ_1, γ_2	Two parameters used to compute a vehicles' U_i with the secret keys a, b as $\frac{\gamma_2^b}{\gamma_1^a} = \frac{(U_i \cdot A_1^{n_i})^b}{(B_1^{n_i})^a} = \frac{U_i^b \cdot A_1^{n_i b}}{B_1^{n_i a}} = U_i^b$.

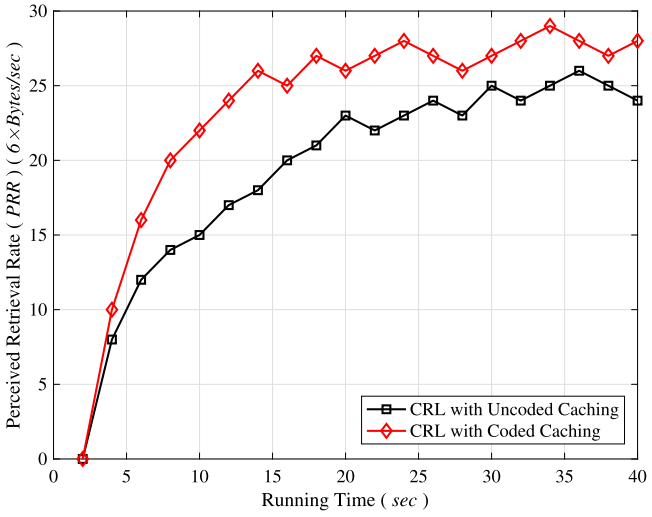

Fig. 12. Comparison of perceived retrieval rate for coded and uncoded caching.

In order to evaluate the global performance of different CRL schemes, including the CRL without cache (the X.509 CRL, Optimized X.509 CRL and bc-CRL), the CRLs with the uncoded caching and multi-layer coded caching (the proposed BAC-CRL) strategies, the delay of target retrieval and Perceived Retrieval Rate (PRR) (Lu et al., 2019) are compared, as shown in Figs. 11 and 12. PRR can be calculated as $PRR = \frac{No.\ of\ Cache\ Items \times Bytes}{Total\ Executing\ time}$. The experiment results indicate that CRL-based scheme with coded and uncoded caching strategies or our BAC-CRL outperform the other three CRL schemes with attaining the much lower delay of target retrieval. Although there is no cognizable performance gap on the retrieval delay, the multi-layer coded caching strategy can not only achieve the overall higher value of PRR but also reach the climactic rate at 17 s, which is 13 s earlier than that of the uncoded caching strategy. The main reason is that multi-layer cache is employed into all individual vehicle in BAC-CRL, in which the identifiers of incoming revoked vehicles from local and neighboring region are pre-cached to accelerate the process of target retrieval and increase the hit rate. Moreover, the required information is combined by applying the element-wise XOR operation during the delivery process for reducing the data load over the shared channel and improve the transmission efficiency.

7. Conclusions

In this paper, the proposed BAC-CRL is designed for authentication in VANETs, which is composed of the blockchain-based CRL and the multi-layer coded caching strategy. In BAC-CRL, the TAs and RSUs in various regions are clustered into a blockchain network by the blockchain-based CRL, which uniformly manages and stores multipart CRLs for preventing resource-constrained OBUs to be overwhelmed. To the best of our knowledge, we are the first to craft the multi-layer coded caching strategy for VANETs to efficiently distribute CRLs that satisfies the revocation requirements, aiming to reduce the communication overhead and shorten the processing time cost. Furthermore, the innovative anonymous authentication method is developed that utilizes the proposed BAC-CRL to attain conditional privacy protection with efficient certificate revocation. Security analysis shows that the proposed BAC-CRL is secure, while simulation experiments demonstrate the effectiveness and efficiency of BAC-CRL as the proposed scheme significantly outperforms the least-recently popular certificate revocation schemes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix. Important parameters, proof, and evaluation

[Important Parameters]:

Here, we summarize the important parameters and variables used in the proposed BAC-CRL into Table 4.

[Proof of Correctness]:

$$N_i = E_{u_{i\_j}} \times DID_{u_{i\_j}}$$
$$= g_1^{-L_j} \times g_1^{L_j+a}$$

$$= g_1^a = A_1$$

$$\lambda_1' = \frac{\gamma_1^{\lambda}}{\gamma_1^{\delta_1}} = \frac{\gamma_1^{\mu+d_k}}{\gamma_1^{d_k-k_1}} = \gamma_1^{\mu+d_k-d_k+k_1} = \gamma_1^{\mu+k_1} = \lambda_1$$

$$\lambda_2' = \frac{\gamma_1^{\lambda} \cdot \gamma_2^{\delta_2}}{\gamma_1^{\delta_1} \cdot \gamma_2^{\lambda}} = \frac{\gamma_1^{\mu+d_k} \cdot \gamma_2^{d_k-k_2}}{\gamma_1^{d_k-k_1} \cdot \gamma_2^{\mu+d_k}} = \frac{\gamma_1^{\mu+d_k-d_k+k_1}}{\gamma_2^{\mu+d_k-d_k+k_2}} = \frac{\gamma_1^{\mu+k_1}}{\gamma_2^{\mu+k_2}} = \lambda_2$$

[Evaluation of Resource Consumption on TA]:

Considering the storage and processing burden of a TA varies depending on several factors, including the number of nodes in blockchain network, the complexity of transactions being processed, etc. Thus, some assumptions are made as follows.

Sizes of Block Elements:
Block Index = 4 bytes
Previous Block Hash = 32 bytes
Merkle Tree Root = 32 bytes
Timestamp = 4 bytes
Nonce = 4 bytes
Difficulty = 4 bytes

Sizes of Transaction Elements:
Transaction Hash = 32 bytes
Transaction Index = 4 bytes
Source TA = 20 bytes
Destination TA = 20 bytes
Digital Signature = 64 bytes

Average Processing Time:
(10 nodes, Difficulty ≤ 3, $n_i = 600$)
Synchronization Time (t_{syn}) ≈ 4 s
Mining Time (t_m) ≈ 60 s
Retrieval Time (t_{retrv}) ≈ 420 ms
Required Retrieval Targets = n_r, $n_r \ll n_i$

With the conditions aforementioned, the evaluation of resource consumption on a TA can be shown as Table 5, where n_t, n_b and t_{retrv} are the numbers of transactions and blocks and the time of target retrieval respectively. Even though the use of blockchain requires extra storage and processing resources, the proposed BAC-CRL is effective especially for these reasons:

(1) For trust and security proposes, many TAs in different regions have already used blockchain in a variety of applications to reduce the risks of tampering fraud, and other security threats, since the use of blockchain can help reduce the risks of tampering, fraud, and other security threats by providing a transparent and immutable record of all transactions and activities within the network (Sai et al., 2021; Li et al., 2017; Liang and Ma, 2021). Most importantly, we do not confine the proposed BAC-CRL with any specific blockchain, which is compatible with the majority of existing blockchain, and therefore it will not impose additional burden on TAs.

Table 5
Evaluation of resource consumption on TA.

	Storage consumption	Average processing time
Blockchain	$140n_t + 80n_b$	$t_{syn} + t_m$
CRLs	$n_i \times 6$	$t_{retrv} \times n_r$
Total	$140n_t + 6n_i + 80n_b$	$t_{syn} + t_m + t_{retrv}n_r$
CRLs share	$\frac{3 \times n_i}{3n_i+70n_t+40n_b}$	$t_{retrv}/(\frac{t_{syn}}{n_r} + \frac{t_m}{n_r} + t_{retrv})$

(2) On one hand, the resource-rich nodes in VANETs, i.e., TAs in different regions, are in charge of the management and maintain of the blockchain to simplify the network structure for the improvement of CRLs distribution and management. On the other hand, the multi-layer coded caching strategy in the proposed BAC-CRL can prevent the resource-constrained nodes in VANETs (e.g., vehicles) from overwhelming by distributing the minimum bit of CRLs that satisfies the revocation requirements to reduce the communication overhead and shorten the processing time cost.

References

Aitzhan, Nurzhan Zhumabekuly, Svetinovic, Davor, 2016. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Dependable Secure Comput. 15 (5), 840–852.

Azam, Farooque, Yadav, Sunil Kumar, Priyadarshi, Neeraj, Padmanaban, Sanjeevikumar, Bansal, Ramesh C., 2021. A comprehensive review of authentication schemes in vehicular ad-hoc network. IEEE Access 9, 31309–31321.

Azees, Maria, Vijayakumar, Pandi, Deboarh, Lazarus Jegatha, 2017. EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. IEEE Trans. Intell. Transp. Syst. 18 (9), 2467–2476.

2023. Bilinear pairing cryptography. [Online]. Available: https://www.ma-th.uwaterloo.ca/ajmeneze/publications/pairings.pdf, Access on: 2023.

2023. Certificate authority. [Online]. Available: https://en.wikipedia.org/wik-i/Certificate_authority, Access on: 2023.

Chatzigiannis, Panagiotis, Baldimtsi, Foteini, Kolias, Constantinos, Stavrou, Angelos, 2021. Black-box iot: Authentication and distributed storage of iot data from constrained sensors. In: Proceedings of the International Conference on Internet-of-Things Design and Implementation. pp. 1–14.

Chuat, Laurent, Legner, Markus, Basin, David, Hausheer, David, Hitz, Samuel, Müller, Peter, Perrig, Adrian, 2022. PILA: Pervasive internet-wide low-latency authentication. In: The Complete Guide To SCION: From Design Principles to Formal Verification. Springer International Publishing, Cham, pp. 461–469.

Dykstra, Dave, Altunay, Mine, Teheran, Jeny, 2021. Secure command line solution for token-based authentication. In: EPJ Web of Conferences, Vol. 251. EDP Sciences, p. 02036.

2023. Elliptic curve cryptosystem (ECC). [Online]. Available: https://en.wi-kipedia.org/wiki/Elliptic-curve_cryptography, Access on: 2023.

Ge, Xiaoxue, Wang, Liming, An, Wei, Zhou, Xiaojun, Li, Benyu, 2022. CRchain: An efficient certificate revocation scheme based on blockchain. In: Algorithms and Architectures for Parallel Processing: 21st International Conference, ICA3PP 2021.

Jan, Sagheer Ahmed, Amin, Noor Ul, Shuja, Junaid, Abbas, Assad, Maray, Mohammed, Ali, Mazhar, 2022. Selwak: A secure and efficient lightweight and anonymous authentication and key establishment scheme for iot based vehicular ad hoc networks. Sensors 22 (11), 4019.

Kumar, Ashish, Saha, Rahul, Conti, Mauro, Kumar, Gulshan, Buchanan, William J., Kim, Tai Hoon, 2022. A comprehensive survey of authentication methods in Internet-of-Things and its conjunctions. J. Netw. Comput. Appl. 103414.

Larisch, James, Aqeel, Waqar, Lum, Michael, Goldschlag, Yaelle, Kannan, Leah, Torshizi, Kasra, Wang, Yujie, et al., 2022. Hammurabi: A framework for pluggable, logic-based X. 509 certificate validation policies. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 1857–1870.

Li, Zhetao, Kang, Jiawen, Yu, Rong, Ye, Dongdong, Deng, Qingyong, Zhangm, Yan, 2017. Consortium blockchain for secure energy trading in industrial internet of things. IEEE Trans. Ind. Inform 14 (8), 3690–3700.

Liang, Junwei, Lin, Qiuzhen, Chen, Jianyong, Zhu, Yingying, 2019. A filter model based on hidden generalized mixture transition distribution model for intrusion detection system in vehicle ad hoc networks. IEEE Trans. Intell. Transp. Syst. 21 (7), 2707–2722.

Liang, Junwei, Ma, Maode, 2020a. ECF-MRS: An efficient and collaborative framework with Markov-based reputation scheme for IDSs in vehicular networks. IEEE Trans. Inf. Forensics Secur. 16, 278–290.

Liang, Junwei, Ma, Maode, 2020b. FS-MOEA: A novel feature selection algorithm for IDSs in vehicular networks. IEEE Trans. Intell. Transp. Syst..
Liang, Junwei, Ma, Maode, 2021. Co-maintained database based on blockchain for IDSs: A lifetime learning framework. IEEE Trans. Netw. Serv. Manag..
Liang, Junwei, Maode, Ma, 2020. Incremental database based on distributed ledger technology for IDSs. In: GLOBECOM 2020-2020 IEEE Global Communications Conference. pp. 1–6.
Lu, Feng, Shi, Ziqian, Gu, Lin, Jin, Hai, Yang, Laurence Tianruo, 2019. An adaptive multi-level caching strategy for distributed database system. Future Gener. Comput. Syst. 97, 61–68.
2023. Merkle tree. [Online]. Available: https://en.wikipedia.org/wiki/Mer-kle_tree, Access on: 2023.
Nayeen Mahi, Md.Julkar, Chaki, Sudipto, Ahmed, Shamim, Biswas, Milon, Kaiser, Shamim, Islam, Mohammad.Shahidul, Sookhak, Mehdi, Barros, Alistair, Whaiduzzaman, Md, 2022. A review on VANET research: Perspective of recent emerging technologies. IEEE Access.
Ozcelik, Ilker, Skjellum, Anthony, 2021. Cryptorevocate: a cryptographic accumulator based distributed certificate revocation list. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, pp. 0865–0872.
Pedarsani, Ramtin, Maddah-Ali, Mohammad Ali, Niesen, Urs, 2015. Online coded caching. IEEE/ACM Trans. Netw 24 (2), 836–845.
2023. Pre-standardization study on pseudonym change management. [Online]. Available: https://www.et-si.org/deliver/etsi_tr/103400_1034-99/103415/01.01.01_60/tr_103415v010101p.pdf, Accessed on: 2023.
2023. Python paring-based cryptography (PyPBC). [Online]. Available: https://crypto.stanford.edu/p-bc/download.html, Access on: 2023.
Sai, Ashish Rajendra, Buckley, Jim, Fitzgerald, Brian, Gear, Andrew Le, 2021. Taxonomy of centralization in public blockchain systems: A systematic literature review. Inf. Process. Manage. 58 (4), 102584.
Saleem, Tania, Janjua, Muhammad Umar, Hassan, Muhammad, Ahmad, Talha, Tariq, Filza, Hafeez, Khadija, Salal, Muhammad Ahsan, Bilal, Muhammad Danish, 2022. ProofChain: An X. 509-compatible blockchain-based PKI framework with decentralized trust. Comput. Netw. 213, 109069.
2023. Tamper-evident technology. [Online]. Available: https://en.wikipedi-a.org/wiki/Tamper-evident_technology, Accessed on: 2023.
Wang, Qianpeng, Gao, Deyun, Chen, Du, 2020. Certificate revocation schemes in vehicular networks: A survey. IEEE Access 8, 26223–26234.
Zulfiqar, Maryam, Janjua, Muhammad Umar, Hassan, Muhammad, Ahmad, Talha, Saleem, Tania, Stokes, Jack W., 2022. Tracking adoption of revocation and cryptographic features in X. 509 certificates. Int. J. Inf. Secur. 1–16.

Junwei Liang received the B.Sc. degree from Guangdong University of Petrochemical Technology, the M.Sc. degree from Shenzhen University and Ph.D. degree from Nanyang technological University in 2014, 2017 and 2021, respectively. He is working as assistant professor at the School of Software, Shenzhen Institute of Information Technology, Shenzhen, China.

His current research interests include intrusion detection systems, evolutionary computation, artificial intelligence and authentication.

Muhammad Sadiq received his MS(CS) degree from Riphah International University Pakistan in 2015. Mr. Sadiq completed Ph.D. from, the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently associated with SZIIT. His research interests are artificial intelligence, cloud computing, cloud security, computer vision, etc. Mr. Sadiq has several publications in the last few years.

Geng Yang received his Doctor of Engineering degree from Hong Kong Polytechnic University in 2018, M.Sc. in Electronic & Information Engineering (EIE, Multimedia Signal Processing and Communications) with Distinction from PolyU in 2010 and BEng in Telecommunications from Xidian University, China in 2009. He is currently an associate professor at the School of Software, Shenzhen Institute of Information Technology, Shenzhen, China.

Dongsheng Cheng received his Doctor of Engineering degree from Sun Yat-sen University in 2011. He is currently a professor at the School of Software, Shenzhen Institute of Information Technology, Shenzhen, China. His current research interests include internet of things, evolutionary computation, artificial intelligence and cyber security.