Contents lists available at ScienceDirect

Journal of Systems Architecture

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

Blockchain-based batch authentication protocol for Internet of Vehicles

Palak Bagga[a], Anil Kumar Sutrala[b], Ashok Kumar Das[a,*], Pandi Vijayakumar[c]

[a] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[b] CA Technologies – A Broadcom Company, Hyderabad 500 032, India
[c] Department of Computer Science and Engineering, University College of Engineering Tindivanam, Villupuram 604 001, India

ARTICLE INFO

Keywords:
Internet of Vehicles (IoV)
Smart city
Batch authentication
Key agreement
Blockchain
Security
AVISPA

ABSTRACT

The vehicles in Internet of Vehicles (IoV) can be used to opportunistically gather and distribute the data in a smart city environment. However, at the same time, various security threats arise due to insecure communication happening among various entities in an IoV-based smart city deployment. To address this issue, we aim to design a novel blockchain-enabled batch authentication scheme in Artificial Intelligence (AI)-envisioned IoV-based smart city deployment. The latest trends and revolutions in technologies incorporate AI/Machine Learning (ML) in blockchaining to produce a secure, efficient and intelligent blockchain based system. The data stored in the blocks in the blockchain are authentic and genuine, which makes the AI/ML algorithms to work at their exceptions in order produce correct predictions on the blockchain data. Through the signing phase of the proposed scheme, each vehicle in a dynamically formed cluster broadcasts a message to its own member and respective road-side unit (RSU). In the proposed scheme, two types of authentication take place: vehicle to vehicle (V2V) authentication allows a vehicle to authenticate its neighbor vehicles in its cluster, while batch authentication permits a group of cluster vehicles to be authenticated by their RSU. At the end, a group key is established among the vehicles and RSU in their cluster. RSU then gathers securely data from its vehicles and form several transactions including the information of vehicles and its own given information to the cluster member vehicles. The transactions are formed later by the nearby fog server associated with RSU and then by the cloud server to form a complete block. The created blocks are mined by the cloud servers in a Peer-to-Peer (P2P) cloud server network through the voting-based Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. The authentic and genuine data of the blockchain are utilized for Big data analytics through AI/ML algorithms. It is shown that the proposed scheme is highly robust against various attacks through formal and informal security analysis, and also through formal security verification tool. A detailed comparative analysis reveals that the proposed scheme achieves superior security and functionality features, and offers comparable storage, communication and computational costs as compared to other existing competing schemes. Finally, the blockchain implementation has been carried out on the proposed scheme to show its effectiveness.

1. Introduction

The increase in number of vehicles in the past years has been both beneficial and harmful to the world. Increase vehicles on roads have inflated the number of accidents, and wait time of the drivers and passengers during traffic jams or blocked roads. However, at the same time it has added to the luxury and ease in living of people. The moving vehicles on the roads form a special type of mobile ad hoc network, called as Vehicular Ad Hoc Network (VANET). The architecture of VANET is designed to provide better road environment, traffic safety and maintaining the privacy of passengers and drivers. The traditional architecture of VANET consists of a *trusted authority* (TA), *road side units* $(RSUs)$ and *on board units* $(OBUs)$/vehicles. The TA registers and authenticates every other entity that participates in the paradigm. The RSUs are placed after every specified distance to manage vehicles in their scope and control the flow of information within their zones. Each zone of RSU consists of some vehicles at every instance. RSUs and trusted authorities are generally at fixed location whereas vehicles do not have constant location and are free to move any where in the network from one RSU zone to another [1–3]. The communication between the RSU and TA can be wired, whereas the vehicles and RSU communicate through wireless medium using the IEEE 802.11p

* Corresponding author.
E-mail addresses: palak.bagga@research.iiit.ac.in (P. Bagga), anilkumarsutrala@gmail.com (A.K. Sutrala), ashok.das@iiit.ac.in (A.K. Das), vijibond2000@gmail.com (P. Vijayakumar).

https://doi.org/10.1016/j.sysarc.2020.101877
Received 11 July 2020; Received in revised form 18 August 2020; Accepted 24 August 2020
Available online 10 September 2020

protocol and Dedicated Short-Range Communications (DSRC). The vehicles also communicate to each other (V2V) and the infrastructure (V2R) to ensure great on road experience and also to safeguard the driver's and passenger's privacy [4]. The communication is basically through messages based on the road conditions or the drivers travel information. The messages being shared among the vehicles can generally include some private details of the driver or the passenger. The new advancements which allow the sensors, infrastructure and intelligent vehicles to communicate by sending messages formed a new paradigm, called the Internet of Vehicles (IoV) [5–7].

There have been various methods formulated in the history of IoV that have worked on the vulnerable side of the paradigm. One of such approaches is the public key infrastructure (PKI), where a certificate authority provides a certificate that contains vehicle's public key [8]. The vehicle uses its private key to sign messages further [9,10]. On the other side, few certificate less schemes, such as a scheme presented in [11], were designed in the recent trends to cope up with this. In those schemes, a key generation center generates a partial private key for the vehicle. The vehicle uses its secret nonces to find the private key.

As the trusted authority needs to keep the track of the vehicles in the network, the communication in IoV deployment requires the vehicles to communicate using their identities and in some cases the messages in the network might contain some secret information. The exposure of secret information makes it easy for the attacker to modify or intercept the message flown within the network. So, this puts a big question on the privacy of the drivers and the fellow passengers. To deal with this, a second approach based on identity based signature came where a private key generator generates a private key for every vehicle corresponding to its identity. This scenario often suffers with a key escrow problem. But, all these scenarios could not overcome the obstacles in the situation where more number of vehicles are participating. As more number of vehicles increases, the certificate issuing and revocation lists also increase. On the other hand, it becomes important to trace an attacker's identity that violated the paradigm. Therefore, it made very clear to the researchers to work on conditional privacy management schemes which safeguard the privacy but also made the identity of the attacker traceable. Few schemes based on group signatures [12–14], certificate revocation list (CRL) and pseudonym generation based authentication were formulated to preserve the privacy. Henceforth, with such complex schemes and increasing vehicles on road raised some issues like long CRL, and hence, more transmission delay, latency, swollen storage space and huge computational cost arise. Besides these factors, the processing time for the schemes based on such scenarios are more than the urgency of flow of information. The bandwidth and computing power of vehicles are limited which make the schemes inefficient. Even the functioning of TA in the scenarios described are not even clearly known to the vehicle because of security reasons. Due to this, the attacker can maliciously fool TA without any knowledge to the vehicles and can enter the network.

The present traditional schemes face two basic problems. Firstly, the flow of the message is made anonymous to ensure the privacy, but this does not guarantee authentication and it does not avoid forge message to enter in the network. To deal with, the advancements in cryptography used aggregation technique as a benchmark in the evolution of security features in IoV. The researchers tried combining various techniques to form the schemes to deal with the issue. They formed the batch verification scheme [15–19], where the RSU can verify the signatures in batches. Such designed schemes are able to deliver average performance, but still to a little extent. They are unable to overcome the shortcomings due to scalability issues or limited computing power. Secondly, focus on conditional privacy and authentication and centralized scenario has reduced the scalability and distribution of information. The single point of failure leaks the vehicular information and hampers the authenticity. So, the need of an hour is to implement a scheme which not only delivers a satisfactory performance in minimum time but also overcomes the shortcomings in the present schemes. The scheme should consider the exponential increase in the number of vehicles, lack of trust among the participating entities as well as computational power of RSU, TA and vehicles.

Keeping in mind the above limitation, blockchain [20] came as a rescue where the participating entities have no trust and are also increasing rapidly in numbers. Therefore, blockchain fits best into such areas and sounds advantageous where vehicles do not trust each other and have to send heavy traffic messages to each other throughout. This technology offers a method to automatically inject trust, checks the credibility of the messages, monitors communication between the entities and analyzes the behavior. It forms a distributed decentralized database by using various technologies, such as digital signatures, cryptographic techniques, hash function, Merkle tree root, and timestamps. All the authorized nodes in the network are allowed to form blocks of transactions which are combined to form a new data structure, called chain. The blocks are combined in the chain in an order in which they are created. Timestamp value in each block is used to determine the sequence of blocks in a blockchain. A link between two blocks is formed by linking the block header with the hash value of the previous block and so on. Every node gets to participate in the chain forming after proving its authenticity. The technology attracted many researchers who were trying to solve the limitation in VANETs as to make the system decentralized. Blockchain technology distributes the responsibility of maintaining privacy and security amongst all the entities in VANET unlike centralized functioning. Few added advantages of the schemes using blockchain are single registration, time series, openness, transparency, traceability, immutability, encryption, message authentication, and scalability.

Blockchain technology can use any of the consensus algorithms, such as Byzantine Fault Tolerance (BFT) [21] and Ripple Protocol Consensus Algorithm (RPCA) [22]. The Byzantine algorithm works in pre-prepare, prepare, commit and reply phases sequentially. A node broadcasts the request to other nodes. The nodes or the servers will receive requests from all the nodes on the network. If multiple servers or the nodes receives same requests from the multiple nodes that request is processed and the reply is sent. The node that initiated the request receives the reply from the nodes and multiple replies forms a consensus. The algorithm can handle one-third of the error nodes. Several consensus algorithms were derived from BFT, such as "Practical Byzantine Fault Tolerance (PBFT)", "Delegated Byzantine Fault Tolerance (DBFT)", "Federated Byzantine Agreement (FBA)" [23, 24]. On the other side, the ripple algorithm works specifically for verification nodes. Each verification node has a set of trusted nodes on the networks. All the nodes receive the transaction on the network and updates its own local ledger. The local ledger is shared and each transaction that is present on the ledger of the trusted nodes gets a vote.

In recent years, the concept of smart cities came up as a plan to mitigate the challenges of very fast and continuous urbanization while keeping better quality of life to its citizens at the same time. IoV provides an enriched information environment by allowing the users (drivers) to share massive information and make easy corresponding decision making. However, with the development of the architecture the risks are also increasing. In an IoV environment, the data is collected, studied and analyzed to accomplish its goal. The major threat to an IoV network is its vulnerable open wireless network. An attacker or any malicious vehicle can anytime intrude, intercept the network to harm the privacy of the drivers or to control the network. Considering the discussed points, the researchers have developed many schemes based on different techniques to ensure both the goal of providing a luxurious experience and to ensure the privacy of the drivers and customers [25]. To mitigate the above issues and enable smart city environment using intelligent vehicles, in this paper, we design a new blockchain-enabled batch authentication scheme in AI-envisioned IoV-based smart city deployment, called BBAS-IoV.

1.1. Network model

The network model for the proposed BBAS-IoV is given in Fig. 1. The model consists of smart city managed by several trusted authorities. It has blockchain center where data securely collected from the vehicles and RSUs are stored in the blocks. A smart city in BBAS-IoV has following entities.

- **TA**: A smart city is managed by various "trusted authorities (TA)" which are responsible to manage their assigned deployment areas in a city. Each TA has various responsibilities that include registering RSUs and vehicles that are in its scope before deploying them in the network and initializing them with the system parameters. TA also forms and delivers certificates consisting of identity, public key, private key to each vehicle and all RSUs.
- **RSU**: A city consists of multiple RSUs. An RSU is responsible to authenticate the messages flown by vehicles within its group or cluster which is dynamically formed with a set of moving vehicles. Like TAs, an RSU also manages all the vehicles, and authenticates the vehicles under its scope. In the designed BBAS-IoV, an RSU receives the messages from all the vehicles in its cluster and authenticates them together in a batch.
- **Vehicle**: The design of the model consists of various vehicles flowing in the network. A group of vehicles on the fly form a dynamic cluster. A vehicle receives a certificate from its associated TA during the registration phase and uses the information in the certificate to generate a hello message along with its signature for verification. Each vehicle within a cluster broadcasts the message and signature within its cluster members. The message is then received by other neighbor vehicles and RSU in its own cluster.
- **Fog server**: One or more RSUs are connected with a fog server, say FS_k, which receives a partial block containing a list of transactions and their compact signature. Upon receiving, the list of transactions and their compact signature, it verifies the attached signature and if the signature is valid, FS_k then forwards the partial block to a cloud server CS_m in the blockchain center.
- **Cloud server**: Upon receiving the partial blocks from its respective fog server(s), a cloud server, say CS_m, converts the partial block to its full block. After that, the full block is mined through the voting based consensus algorithm, which is based on the widely-used "Practical Byzantine Fault Tolerance (PBFT) consensus algorithm" [26].

1.2. Threat model

The threat model considered in the proposed scheme (BBAS-IoV) is the widely accepted "Dolev-Yao threat (DY) model" [27]. According to this model, the end entities involved in the communication (vehicles and RSUs) are not considered to be fully trusted. Moreover, they are assumed to communicate over an insecure, open, public channel. However, the trusted authorities (TA) are fully trusted, and the fog servers and cloud servers are assumed to be semi-trusted. The model also considers an adversary, say $\mathcal{A}$, who has enough computational power that he/she can eavesdrop, update, replay, fabricate or delete the exchanged messages during transmission. This gives a privilege to $\mathcal{A}$ to tamper any message for own benefit. The RSUs are assumed to be semi-trusted, whereas the vehicles are equipped with tamper-resistant onboard units (OBU) where the secret credentials are stored. Furthermore, it is assumed that the RSUs will store the secret credentials in their secure memory (database) so that the stolen verifier attack is not possible by $\mathcal{A}$.

Another model, known as the "Canetti and Krawczyk's adversary model (CK-adversary model)" [28] has been also considered. It has become a "current de facto standard model in modeling authenticated key-exchange protocols". The CK-adversary model is quite similar to the DY threat model with the additional features wherein the adversary $\mathcal{A}$ cannot only send the messages, but also can compromise the secret credentials, secret keys and session states. For an authenticated key agreement scheme, the leakage of some secret credentials from a session hijacking attack should definitely have a minimal impact on the security of the established session key. Therefore, it becomes important to consider this model to analyze the security of a designed authenticated key-exchange protocol.

1.3. Research contributions

The main research contributions are given below.

- In the proposed scheme (BBAS-IoV), we have two types of authentication mechanisms: (a) vehicle to vehicle (V2V) authentication allows a vehicle to authenticate its neighbor vehicles in its cluster and (b) batch authentication permits a group of cluster vehicles to be authenticated by their RSU. At the end, a group key is established among the vehicles and RSU in their cluster. RSU gathers securely data from its vehicles and form several transactions including the information of vehicles and its own given information to the cluster member vehicles. The transactions are formed later by the nearby fog server associated with RSU and then by the cloud server to form a complete (full) block.
- In order to make the proposed BBAS-IoV more effective, we incorporate the blockchain technology using the fog servers and cloud servers as well. The blocks of transactions are mined by a voting-based PBFT consensus algorithm. The genuineness and authenticity of the huge amount of data stored in the blocks in the blockchain allows the use of big data analytics though AI/ML algorithms.
- The proposed BBAS-IoV supports dynamic vehicles and RSUs addition after initial deployment.
- A comprehensive security analysis through the formal security analysis under the widely-used "Real-Or-Random (ROR)" oracle model [29], non-mathematical (informal) security analysis and formal security verification using the broadly-accepted "Automated Validation of Internet Security Protocols and Applications (AVISPA)" [30] software tool reveals that BBAS-IoV can resists various potential attacks against passive/active adversaries.
- We evaluate various cryptographic primitives using the broadly-accepted "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [31] for their execution time under a server and Raspberry PI 3 platforms.
- We then perform a detailed comparative study on "communication and computational overheads", "storage overheads" and "security and functionality features" among BBAS-IoV and other existing competing authentication schemes in IoV-related environment.
- Finally, we provide the blockchain implementation of the BBAS-IoV using the popular Hyperledger Sawtooth [32].

1.4. Paper outline

The remainder of the paper is sketched as follows. The next section gives an overview of the existing schemes relevant to an IoV environment. In Section 3, the detailed discussion on various phases related to our scheme (BBAS-IoV) is provided. While the correctness proof, formal and informal security analysis of the proposed BBAS-IoV are provided in Section 4, the formal security verification of BBAS-IoV is shown in Section 5. The experimental results using MIRACL provided in Section 6 are used in the comparative analysis in Section 7. Section 8 provides the blockchain implementation of the proposed BBAS-IoV. Finally, the paper is concluded in Section 9.

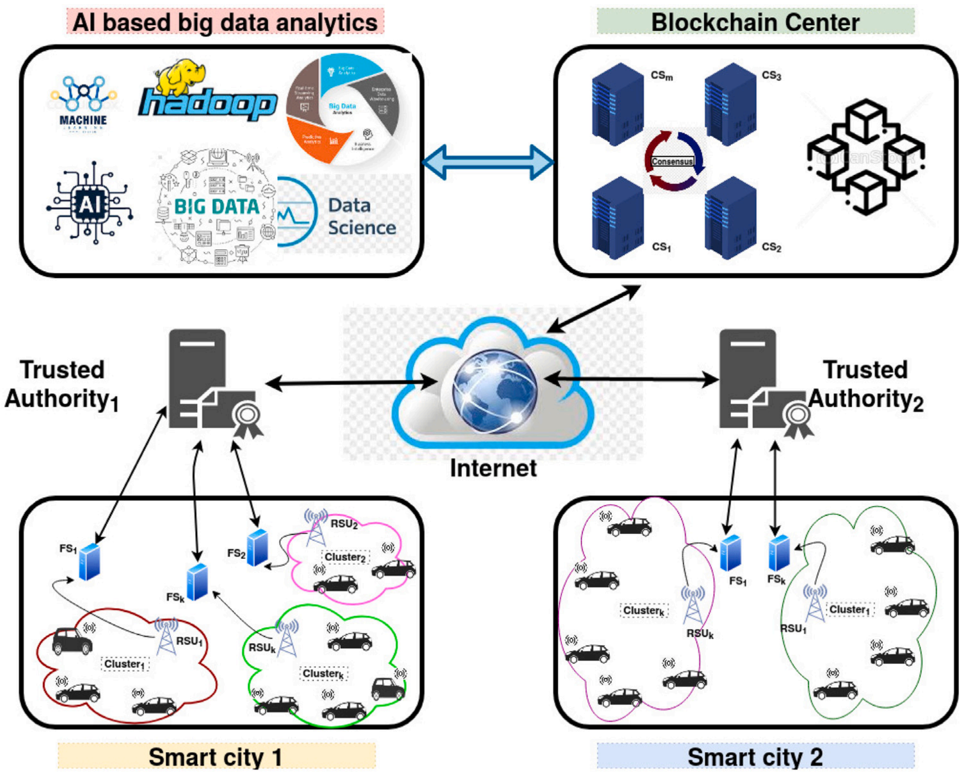


Fig. 1. Blockchain-enabled AI-envisioned IoV-based smart city architecture.

2. Related work

This section describes a detailed review on the existing competing schemes based on pseudonyms, certificate-less and batch verification Following to this are the schemes that have used blockchaining as a concept.

Tzeng et al. [33] and Bayat et al. [18] identified the weakness of secret key in Lee and Lai [34] scheme where a malicious vehicle is able to form a signature on any message on behalf of legitimate vehicle if it gets exposed to any previous valid signature. Moreover the scheme does not support traceability and could not safeguard the privacy, also is vulnerable to forgery attack. So the authors in [18] presented an enhanced batch authentication scheme improving the vulnerability in Lee et al. scheme to avoid vehicle impersonation attack. The scheme runs three phases. TA initializes the public parameters and loads in RSU and vehicles, each vehicle is also assigned with the real identities. After verifying real identity and password, tampered proof device of each vehicle generates pseudonym in second phase. The vehicles use the pseudonym(pseudo identity and private key) to sign the message and then send it to other vehicles or RSU's. The receiver of the message verifies the message individually or in batches using bilinear pairing and timestamps. The authors successfully removed the weakness of Lee et al. [34] scheme although [18] suffers from modification attack. Tzeng et al. [33] proposed a similar scheme whose system model consists of redundant TA's, application server, non trusted RSU's and tampered proof OBU's. The entities are assumed to be synchronized to a certain extent. First phase similar to most of the authentication scheme is where TA initializes the public parameters. Second phase is headed by tampered proof device that generates the pseudonym on getting the request from the vehicle. To add up to the security TPD's passes the request of the vehicle to an authentication module. Authentication module checks for legality of the vehicles and then, anonymous identity generation module generates an anonymous identity. Vehicles after receiving the pseudo identity from the second phase, uses message signing module to sign the traffic related message before broadcasting. The verification process is performed by RSU sequentially or in batches by using vectors.

Zhong et al. [35] proposed a secure and efficient conditional privacy preserving authentication scheme based on pseudonym signatures. TA's in the model of the scheme are assumed to be credible and responsible for vehicle registration, management and traceability. RSUs are deployed along the roads to manage area of about 300 to 500m. The communication is from vehicle to other vehicle or RSU. The first phase initializes the system parameters. TA loads the real identity, password and other secure material into TPD of each vehicle. TPD of the vehicle executes the next phase to generate pseudonym and accomplish key generation process. A vehicle acquires its secret key, pseudo identity which it can use to sign the message by executing the next phase. TPD signs the message and sends a tuple consisting of signature, timestamp, message and pseudo ID. The receiving OBU or RSU checks the received message for its validity, and verifies the signature. On receiving multiple requests, the scheme performs a batch verification process to reduce computation overheads by using random vectors and small exponent test. The scheme is safe against replay attacks, and is also traceable.

He et al. [36] stressed on ID based conditional privacy preserving authentication scheme without bilinear pairing. The scheme uses elliptic curve cryptographic techniques to accomplish the goal. TA is again the most trusted and the only party that has the real identities of all the vehicles. The model in the scheme also uses application server that is present at the traffic management center to support traffic management. The TA initializes the system parameters and forwards it to RSU and also loads the real identity, password and other public parameters in the tampered proof device of the vehicle. The tampered proof device then generates an anonymous identity and secret key and hands it over to vehicle. The vehicle uses pseudo identity to generate a digital signature on the message it wishes to send. The last phase is performed by any RSU or vehicle who is the verifier of the received message. The digital signature and the validity of the timestamp in the received messages are the criteria of the validation. The scheme even supports batch verification to reduce the computation overheads. Although the scheme is secured from various attacks like replay, man-in-the-middle, modification, impersonation attack, but can still suffer from physical capture insider attack.

Jiang et al. [37] prescribed an anonymous batch authentication based on HMAC for VANETs (ABAH). The authors designed the scheme in a way that it checks the integrity of the message before batch authentication to avoid unnecessary delay during rejections at the time of batch authentication. The continuous checking of CRL's was also replaced by group based mechanism to reduce the communication overheads. The use of HMAC maintains the efficiency of the batch authentication process of the scheme by removing the messages that does not ensure integrity or are from the illegal vehicles. The system model consists of TA, RSU's and vehicles. TA's which are assumed to be secured are responsible to generate secure materials for RSU and vehicles. RSU performs batch authentication and validates multiple vehicle requests, and finally handovers a group key to the group of legitimate vehicles. Another set of batch authentication occurs between vehicles receiving multiple traffic related messages from the fellow vehicles. In commencing phases, TA initializes the public parameters and issues a certificate. The certificate consists of the domain number, public key and signature of the corresponding RSU. In third phase, TA divides the time into slots and delivers the pseudo identity, secret key to the vehicle for different time slots at once. The mutual authentication process occurs between the vehicles and RSU when a vehicle enters any RSU's domain and receives its broad casted certificate. It then verifies the certificate and sends message related to traffic or other details to corresponding domain RSU. On receiving the message, RSU checks the legitimacy of the vehicle's pseudo identity in its revocation list and distributes the group key to the authenticated group. The advantage of the scheme is that it does not broadcast the CRL to every entity of the network to avoid communication overhead. CRLs are only sent to RSU through secure wired links.

Gayathri et al. [15] designed a pairing free certificate less batch verification scheme, that does not provide mutual authentication. The scheme comprises of seven algorithms where in TA and key generation center (KGC) are fully trusted. TA stores the real identities of the vehicles and provides them with a token which is used by KGC to provide partial private key. TA can also revoke the malicious vehicles. It maintains a revocation list which is shared with RSU and KGC via secure channel. RSU is responsible to handover pseudo identities to the vehicles coming into their zone after checking the legitimacy of the vehicle against the revocation list. A vehicle's OBU stores a secret token and the signature received by TA and pseudo identity received by RSU. It signs the message using pseudo identity and private/public key pair and other secret information from its OBU. Similarly each vehicle verifies the messages and signature received by fellow vehicles. RSU performs the batch verification after receiving n different messages from n vehicles.

Zhang et al. [38] presented an effective and anonymous batch authentication scheme based on pseudo identity generation using hash functions and elliptic curve cryptography. Similar to Jiang et al. [37], TA 's divide their scope in domains and are assumed to have enough power to generate certificates for RSU's; certificate, pseudonym and secret key for each vehicle. TA contains a certificate revocation list for both RSU's and vehicles which depends on the number of vehicles and RSUs. TA initializes the public parameters followed by the second phase in which a certificate is issued. The certificate consists of the domain number, public key and signature of the corresponding RSU. In the next phase, series of pseudo identities and secret keys is delivered to vehicle for every j slots throughout the year at once during the time of registration via secure channel. The vehicle captures and verifies the certificate sent by RSU in its domain. It signs and sends the traffic related message and its signature to RSU. Finally the RSU performs a batch verification using exponent test after confirming the absence of vehicle in CRL issued by TA. TA revokes the illegal RSU and vehicles by mentioning their pseudonymous certificate in revocation list. TA identifies the illegal vehicle by calculating its impact range area considering the speed it is moving and the distance it covers in the time the false information sent by vehicle is received by RSU. Similarly, compromised RSU are revoked by sending the CRL to all other vehicles and RSU which falls under the impact range area.

Sutrala et al. [16] used batch verification to preserve privacy of the vehicles in IoV paradigm. The scheme utilized less communication and computation overheads. The scheme starts with initializing the system parameters including the public and private keys of the TA, key generation center (KGC) and for each RSU. The TA registers the vehicle using its private key and KGC's public key. Further KGC generates a partial private key and delivers to every legitimate vehicle before its communication with RSU or other vehicles. Then each vehicle generates a public private pair key after receiving partial private key from KGC. To conserve the real identity of the vehicle authors added another phase to generate pseudo identities for each vehicle. A vehicle chooses some random nonces and requests the nearest RSU for pseudo identities. Finally the vehicle becomes eligible to send the message and communicate with others in the group. The vehicle signs the message and sends the message along with the signature and its pseudo identity. RSU is open to listening the broadcasted messages from the vehicles. To reduce the computation overheads the scheme is designed to support batch verification. RSU collects the messages to verify and validates them in batches against pseudo identities, timestamp and altered messages.

Guehguih et al. [39] proposed a blockchain based privacy preserving authentication and message dissemination scheme by using both public and private blockchains. The use of blockchain waives off the dependency entirely on TAs and also implements the role of traditional RSUs. TA is responsible to form a private blockchain by adding blocks that consists of the information only about the authenticated vehicles in the network. All other entities in the network have only read access to the private chain to get the information about authenticity of fellow vehicles. The public blockchain known as road side unit blockchain (RSU BC) is used to implement the role of RSU's to monitor the event message dissemination and flow of messages in the network. Only special vehicles called as miners are allowed to participate in formation of public chains by broadcasting the mined blocks to other vehicles. Miners initially confirm the authenticity of the vehicle that has sent the event message through the private block chain. Each light weight vehicle not specifically the miner, adds the received block in their chain. The transaction in blocks of private blockchain consist of authentication information where as the blocks of public blockchains consist of event messages. The scheme is secured against bogus information attack, impersonation attack, non repudiation attack.

Lu et al. [40] eliminates the use of periodic certificate issuing and revocation in their blockchain-based privacy-preserving authentication scheme (BPPA). The transparency between the entities and various TAs is achieved by adding the activities or transactions performed by TAs in block and further in permissioned blockchains which are verifiable by all the entities in the network. The transactions include the issuing and revocation of certificates. The certificates are provided to the vehicle on its request to Law enforcement authority which directs CA to issue them. The certificates contain timestamps, link-ability to identity, public key. The transactions made from the certificate authority are grouped into blocks by law enforcement authorities (LEA). The traceability between the certificates and the identities are encrypted using the secret key of LEA and also stored in blockchains. RSUs have authority to access blockchains to verify the activities of CA and LEA to provide consensus and forward the updated blockchains to the vehicles. Chronological Merkle Tree (CMT) and Merkle Patricia tree are used to store the blocks. CMT stores the transaction root and MPT stores the certificate root. The scheme uses less time and storage as compared to the usual schemes by using Merkle trees as a data structure. Conditional privacy of the scheme is ensured by providing multiple certificates corresponding to identities to each vehicle.

Wang et al. [41] suggested an improved authentication mechanism based on blockchain technology. It improves the process by using public key pair authentication method based on crypto accumulator. The system architecture used for the scheme is trusted cloud providers (CA), road side units (RA) and vehicles. The authentication in the scheme is divided into two parts one is between a vehicle and road side unit. Other is to verify RA and CA. A contract node group is formed that comprises of validated road side units, vehicles and trusted providers. Any new node that wishes to join the network needs to send the request to the contract nodes group basically RSU. The request consists of vehicles public key, identity, random number and timestamps. Depending on the node's history and credits a consensus is formed so as to whether to allow a node to join the network or not. Subsequently, RSU performs a validation and attaches the secret key along with the request message received from the vehicle and forwards the request to cloud service providers. CA verifies the identity of the vehicle and of RA in the corresponding blockchain and then issues a digital certificate to the vehicles via RA. The issuer CA broadcasts the information to other CA's to add it into their account book. Other receiving CA's will only store the information in their books if they find the information to be relevant otherwise they will send a negative feedback to the CA that is issuing the digital certificate.

Jheng et al. [42] proposed a secure access authentication scheme between vehicles and roadside units, which is decentralized as it does not fully depends on centralized authority. Their scheme also has a scenario of distributed cloud ledger that reduces the storage burden on the blockchains. All RSUs form a peer to peer network that contributes in the formation of blockchain. The blocks in the blockchain are used to store information about the vehicle's identity and the link between the identity and pseudo identity. The network model consists of a certificate authority (CA) which issues a certificate to an authorized, registered vehicles and also prepares a hash which is used by road side units (RSU) to form a blockchain after verification. CA produces two hash values one to send the pseudonym to the vehicle and other to store the mapping relationship of vehicle's real identity and pseudonym. A cloud server is used to store hash values produced by CA and is acquired by RSUs during verifying the legality of the vehicle. RSU on receiving the request from the vehicle initiates the verification process. After the verification process, an integer negotiation process takes place between RSU and vehicle to confirm the integrity of message and receiver. The random number in the process used is further used by the vehicle to announce the traffic. Finally the transaction including pseudo identity, public key of the vehicle is added in the blockchain. The traffic announcements by vehicles are also verified by RSUs and then hash value of the event transaction is added in the blockchains. The actual event details corresponding to the hash value is stored in the cloud server which is accessed by the fellow vehicles on getting a notification of occurrence of any event by various RSUs.

Feng et al. [43] formulated a BPAS: Blockchain-Assisted Privacy-Preserving Authentication System for Vehicular Ad Hoc Networks using smart contract and hyperledger fabric as an open platform. Fuzzy extractor and attribute-based encryption are the backbones of the protocol. The entities involved in the scheme are TAs, RSUs, vehicles. Smart contracts are used to store vehicle public key table (VPKT) in the form of blockchains which are appended by TAs. The use of smart contract makes the scheme self-verifying, self executing and decentralized by providing application binary interfaces (ABI). It manages VPKT by implementing algorithms to insert, update and revoke entries in VPKT. TA makes use of the application to insert, revoke or verify the vehicle's public key in the table. TA initializes the parameters along with setting up ABE and blockchain within the network using practical Byzantine fault tolerance consensus. Further smart contract is deployed in the blockchain that forms a data structure to save VPKT. The registration phase lets the TA to update the VPKT and OBU with essential parameters for further processing. On receiving some message related to traffic, RSU or vehicle checks the timestamp value and forwards the message to blockchain managers. The blockchain managers eventually screen the message against the ABE and VPKT to know the legality of the sender and the message.

Luo et al. in the paper [44] designed a blockchain based decentralized or distributed privacy protection scheme. Each vehicle uses location information, trust worthiness of other vehicles to implement k-anonymity and forms anonymous cloaking region request utilizing location of $k-1$ near by vehicles. The system model has a registration authority (RA) which provides a certificate containing pseudonym to vehicles and also forwards vehicle's pseudonym and trust level to all RSUs. The authors made this scheme distributed by replacing the use of single trusted authority by many RSU's handling blockchain. RSUs store and add the blocks containing information about vehicle's pseudonym and trust level in the form of a transaction by implementing consortium algorithm. A vehicle initiates request to form a cloaking region. The eligibility to participate in the region is based on the degree of trust level of both the requesting and nearby vehicles. Dirichlet distribution is used to calculate trust levels. The method uses both current behavior and historical trust level of vehicle as the measuring criteria. Every time the current behavior of the vehicles is calculated it is also saved in the blockchain by RSU to maintain and update the historic data. Once the calculations are done the vehicles passing the degree of trust level send their location to the requester. A requester constructs an anonymous cloaking region using cooperating vehicles and forwards the request to location service provider. To safeguard the identity of the vehicle each vehicle undergoes a process to change its pseudonym after participating in x cloaking regions by requesting RA. RA delivers a new certificate containing a new pseudonym and updated trust level to both vehicle and RSU to add in the blockchain.

Tan and Chung [45] advanced the security properties of VANETs by presenting a mutual authentication and group key management scheme using features of cloud computing and edge computing layers for RSUs. Instead of using certificates, cloud assisted infrastructure is used in the scheme. The system model divides the architecture into three layers. The first layer known as the cloud layer consists of TA. It performs all the important functions and is fully trusted. The second layer or edge layer comprises of RSU that forms the edge clusters making the scheme decentralized. The last layer is the user layer consisting of vehicles. The first phase of the scheme is the offline registration where a unique identity and partial secret is given to each RSU and vehicle. Both RSU and vehicle generates a temporary session ID using the assigned unique ID and timestamp values to communicate. All important information related to vehicle and RSU is stored in cloud server. In the second phase vehicle validates the RSU by verifying its publicly issued certificate. Following to this it sends a request to RSU by creating its signature. RSU verifies the signature or a group of signatures in a batch by forwarding the request to cloud server for actual identification. To isolate V2R communication a separate session key establishment takes place amongst the group of vehicles(V2V) and their nearest RSU over an allocated channel by implementing Chinese remainder theorem and consortium blockchain algorithm for updation/revocation the key while tracing real time group membership. Assigning heavy computational tasks to TA (cloud layer), light computations like key management to vehicles and decentralization improves the response time and lowers latency of the scheme.

Sharma and Chakraborty [46] framed a Blockchain for Authentication and Privacy Preservation (BlockAPP) protocol for IoV environment to provide seamless access control. The system model in the protocol consists of a registration server, service providers, vehicles. Registration server manages vehicle's identity. A vehicle uses this identity to access service from service providers after being verified. Blockchain is stored with central trusted authority which can be read and written by service provider but can only be written by registration server. The transactions can be of three types one for each phase: registration transaction

Table 1
Summary of characteristics of existing competing protocols.

Scheme and year	Cryptographic techniques	Characteristics
Zhang et al. [38] (2020)	*Hash chains *Elliptic curve cryptosystem	It is an effective anonymous batch authentication scheme which provides message integration, identity revocation, unlinkability and backward secrecy. It avoids replay attack, but involves huge computational overheads due to periodic issuing of CRL.
Tan and Chung [45] (2020)	*Blockchaining *Elliptic curve cryptography *Bilinear pairing *Hash functions *Chinese Remainder Theorem	It is a light weight certificate less scheme that is unforged against chosen message attack. It reduces the latency by assigning heavy tasks to the TA and lighter task like key management to vehicles.
Feng et al. [43] (2020)	*Blockchain technology *Fuzzy extractor *Attribute based encryption	This scheme is based on blockchain that maintains the privacy using three factor security and verifies the messages. It has a mechanism to revoke the vehicles, but does not manage the keys.
Luo et al. [44] (2020)	*Blockchain *Dirichlet distribution	It is a blockchain enabled trust-based location privacy protection scheme that uses k-anonymity to construct cloaking regions based on trust value determined by current behavior and historic trust of the vehicles. It does not match authentication or verification criteria.
Wang et al. [41] (2019)	*Blockchain *Digital certificates *Public key pair crypto accumulator	It uses blockchain technology to authenticate any new node joining the network and improves key distribution. It does not have provision of batch authentication of messages and upgradation of the key.
Lu et al. [40] (2019)	*Blockchaining *Secure hashing algorithm *Advanced encryption standard *Elliptic curve cryptography *Elliptic curve digital signature	Their scheme uses blockchain to record the functionalities of the TA to provide transparency.
Guehguih et al. [39] (2019)	*Blockchaining *Public key infrastructure *Elliptic curve cryptography *Hash function	The scheme is free from RSU deployment and is secured against bogus and impersonation attack. The scheme stores the event messages in the blockchain increasing the efficiency but does not support batch authentication.
Zheng et al. [42] (2019)	*Elliptic curve cryptography (ECC) *Public key infrastructure *Hash functions	Their scheme provides secure and anonymous authentication along with the privacy preservation. Blockchain holds the messages to be sent by the vehicles and forbids the distribution of malicious message and also reduces the storage overheads.
Gayathri et al. [15] (2018)	*Elliptic curve cryptography *Elliptic curve discrete logarithm problem (ECDLP)	Their scheme is pairing-free certificateless authentication scheme with batch verification. It faces communication overheads due to revocation list traveling from the TA to RSU and KGC, and might suffer from OBU capture attack.
Tzeng et al. [33] (2017)	*Modules for each phase *Small exponent test *Bilinear pairing	Their scheme is a identity based batch verification scheme that is resistant to replay attacks. But with large number of vehicles, their scheme can be inefficient due to use of vectors in batch verification.
Jiang et al. [37] (2016)	*Bilinear pairing *Hash based message authentication code *Elliptic curve discrete logarithm problem (ECDLP) *Computational Diffie–Hellman problem (CDH)	Their scheme provides authentication, conditional privacy, identity revocation, backward privacy, message integrity and supports batch verification. It is not scalable, has storage overheads and suffers from Denial of Service (DoS) attack on RSU.
Zhong et al. [35] (2016)	*Elliptic curve cryptography *Discrete logarithm problem	Their batch verification scheme provides authentication and it follows conditional privacy by being unlinkable yet traceable, non repudiation.
Bayat et al. [18] (2015)	*Elliptic curve cryptography *Computational Diffie–Hellman problem (CDH) *Bilinear pairing	Their authentication scheme resists replay attack, and verifies multiple messages simultaneously and preserves privacy. Their scheme has a slight computational delay in generating pseudonyms, and cannot resist modification attack.

which is added by registration server for every vehicle registered, authentication or authorization transaction which is read and written by service provider to authenticate the vehicle and to provide the service to the authorized vehicle. A vehicle after getting authenticated by TA sends the registration request to the registration server to receive pseudo identity. After sending the pseudo identity, RS adds the valid registration transaction to blockchain and removes the details. Next the vehicle sends the access request to the service provider. The service provider checks for the vehicle's registration transaction in the blockchain to verify its authenticity. If valid, it adds an access log in the blockchain. A legitimate vehicle forms a digital signature on the service it needs and sends it to the service provider to avail the service. The service provider adds the authorization transaction in the blockchain and offers the service to the vehicle.

To protect the private key in mobile devices, He et al. [47] proposed a novel two-party distributed signing protocol for the identity-based signature scheme in IEEE 1363 standard, which allows two devices to generate a valid signature without revealing the entire private key of the user. Later, Feng et al. [48] also proposed a new share conversion protocol and constructed the first multi-party distributed signing protocol for the identity-based signature scheme in IEEE 1363 standard.

Table 1 summarizes the schemes described in related work. The table contains the information related to the cryptographic techniques applied and the limitations, benefits of the schemes.

3. The proposed scheme

In this section, we propose a new blockchain-enabled privacy-preserving batch authentication in AI-envisioned IoV-based smart city environment. In short, we call this scheme as BBAS-IoV. BBAS-IoV follows the architecture described in Section 1.1. We assume that the trusted authorities (TA) have high computational power and are unconditionally secured. The proposed BBAS-IoV has following phases: (1) initial setup phase, (2) vehicle and RSU registration, (3) message

Table 2
Notations and their meanings.

Symbol	Description
TA_k	Trusted authority
RSU_i	Road Side Unit
V_j	jth vehicle in any cluster C_k
C_k	kth cluster of vehicles of
ID_j	Unique identity of vehicle V_j
ID_{RSU_i}	Unique identity of RSU_i, managing cluster C_k
RID_{V_j}	Pseudo identity of V_j
(Pub_{TA_k}, pr_{TA_k})	Public and private keys pair of TA_k
Pub_{V_j}	Public key of vehicle V_j
$(pp_{V_{j0}}, pp_{V_{j1}})$	Partial private key pair of a vehicle V_j
pr_{V_j}	Private key of a vehicle V_j
Pub_{RSU_i}	Public key of an RSU_i
pr_{RSU_i}	Private key of an RSU_i
$Cert_{V_j}$	Certificate for vehicle V_j
Sig_{V_j}	Signature generated by V_j
GK_k	Group key for members of cluster C_k
G_1, G_2	An additive group and a multiplicative group of prime order q, respectively.
G	A generator of G_1
X_k, r_{V_j}	Random nonces chosen by TA_k
a_{V_j}, b_{V_j}, rn_{V_j}	Random nonces chosen by V_j
rn_{RSU_i}	Random nonces chosen by RSU_i
$h(\cdot), H_0(\cdot), H_1(\cdot)$, $H_2(\cdot), H_3(\cdot), H_4(\cdot)$	Secure cryptographic One-way hash functions
TS_{V_j}, TS_{RSU_i}	Current system timestamps generated by V_j and RSU_i respectively
$ECDSA.Sig(\cdot)$	Signature generation algorithm in the elliptic curve digital signature algorithm (ECDSA)
$ECDSA.Ver(\cdot)$	Signature verification algorithm in ECDSA
$EP(\cdot)/DP(\cdot)$	ECC-based public key encryption/decryption function
$EC(\cdot)/DC(\cdot)$	Symmetric key encryption/decryption function
ΔT	Maximum transmission delay
$x * y$	Modular multiplication of elements $x, y \in Z_q$
$\|\|$	Data concatenation

signing phase, (4) batch authentication phase, (5) group key management phase, (6) blockchain formation phase, (7) AI-based secure big data analytics using blockchain phase, and (8) dynamic nodes addition phase. The notations used through out the description of the scheme is described in Table 2.

To protect the replay attacks against an adversary, we apply the system current timestamps. For this goal, it is an assumption that all the involved entities in the network are synchronized with their clocks. It is a typical assumption that is applied in the design of many authentication protocols in various networking environments [49–59].

In this paper, a dynamic clustering approach suggested by Kakkasageri and Manvi [60] has been considered to generate various clusters of vehicles on the fly. Their approach allows the cluster members to be initially picked based on the "relative speed of vehicles" along with the "direction for dynamic clustering". After that a cluster head may be picked among the available cluster members based on various metrics, such as "connectivity degree", "average speed" and "time to leave the road intersection". They also pointed out that the cluster members with similar mobility pattern can later reconnect with the cluster head after "passing an intersection of the lane". Their simulation study showed that the cluster formation stage was efficient. Hence, we consider here the dynamic clusters as it was the case in [50] because the clusters are dynamically formed.

Various phases associated with the proposed BBAS-IoV are elaborated in the subsections below.

3.1. Initial setup phase

This phase is performed by all the fully trusted TAs (TA_k) to set up the system parameters. Suppose the TA_k manages registering of the vehicles containing in a particular zone of the smart city. It performs the following steps to generate the system parameters and compute its public and private keys pair.

IS_1: The TA_k first chooses a non-singular elliptic curve $E_q(u, v)$ of the form: $y^2 = x^3 + ux + v \pmod q$ such that $4u^3 + 27v^2 \neq 0 \pmod q$. Next, the TA_k picks an additive group G_1 with point at infinity $\mathcal{O}$ and a multiplicative group G_2 with identity 1 of prime order q. It selects G as a randomly-chosen generator of G_1 and e as a bilinear mapping e: $G_1 \times G_1 \rightarrow G_2$ with the following three properties [61,62]:

- **Bilinearity**: For all $P, Q, R \in G_1$, $e(P + Q, R) = e(P, R)$ $e(Q, R)$ and $e(P, Q + R) = e(P, Q)e(P, R)$. In general, for all $a, b \in Z_q^* = \{1, 2, \ldots, q - 1\}$, $e(aP, bQ) = e(P, Q)^{ab}$.
- **Non-degeneracy**: $e(P, P) \neq 1_{G_1}$ for all $P \in G_1$, where 1_{G_1} is the identity in G_1.
- **Computability**: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

IS_2: TA_k then randomly selects $pr_{TA_k} \in Z_q^*$, $X_k \in \{0, 1\}^*$ and sets pr_{TA_k} as its private key. It then uses its private key to compute its public key as $Pub_{TA_k} = pr_{TA_k} \cdot G$, where $k \cdot G$ denotes the "elliptic curve point (scalar) multiplication" with $k \in Z_q^*$ which is defined as repeated "elliptic curve point addition": $k \cdot G = \underbrace{G + G + \cdots + G}_{k \text{ times}}$.

IS_3: Further, TA_k selects six "one-way collision-resistant cryptographic hash functions" defined by h: $\{0, 1\}^* \rightarrow Z_q^*$, H_0, H_4: $\{0, 1\}^* \rightarrow G_1$, H_1: $G_1 \rightarrow G_1$, H_2: $\{0, 1\}^* \times G_1 \rightarrow Z_q^*$, H_3: $\{0, 1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$.

IS_4: Finally, TA_k publicly publishes $\{G_1, G_2, e, G, E_q(u, v), X_k, Pub_{TA_k}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$ as the system parameters.

3.2. Registration phase

In the registration phase, the TAs are responsible for registering the vehicles and RSUs prior to their deployment in the IoV-enabled smart city environment.

3.2.1. Vehicle registration

Suppose TA_k manages a group of vehicles $V_1, V_2, \ldots, V_n$ that belong to a particular region of a smart city. TA_k registers each vehicle V_j, $(j = 1, 2, \ldots, n)$ before deploying them in the network. In this phase, TA_k chooses a unique identity and a public key for the vehicle V_j. It also performs some computations to generate a partial private key for each V_j. At the end of the phase, TA_k handovers a pseudo identity, hashed public key, partial private keys packed in a certificate to the vehicle V_j. The steps involved in the phase are described as follows.

VR_1: TA_k generates a random unique identity ID_j for any vehicle V_j of the region of the smart city. It then calculates pseudo identity RID_{V_j} corresponding to its unique identity by $RID_{V_j} = H_0(ID_j)$.

VR_2: TA_k chooses a random key $P_{V_j} \in Z_q^*$ for vehicle V_j. It then calculates the public key corresponding to P_{V_j} as $Pub_{V_j} = H_1(P_{V_j})$.

VR_3: TA_k also computes the partial private key pair for each vehicle V_j as $pp_{V_{j0}} = pr_{TA_k} \cdot RID_{V_j}$ and $pp_{V_{j1}} = pr_{TA_k} \cdot Pub_{V_j}$.

VR_4: Next, TA_k generates a random secret $r_{V_j} \in Z_q^*$ and computes $R_{V_j} = r_{V_j} \cdot G$ for V_j. Further, TA_k computes a certificate $Cert_{V_j}$ for vehicle V_j as $Cert_{V_j} = pr_{TA_k} + h(RID_{V_j} \| Pub_{V_j} \| R_{V_j}) * r_{V_j} \pmod q$.

VR_5: TA_k loads the credentials $\{RID_{V_j}$, R_{V_j}, $Cert_{V_j}$, $(pp_{V_{j0}}$, $pp_{V_{j1}})$, Pub_{V_j}, $h(\cdot)$, $H_0(\cdot)$, $H_1(\cdot)$, $H_2(\cdot)$, $H_3(\cdot)$, $H_4(\cdot)\}$ in the tamper-resistant on-board unit (OBU) of the vehicle V_j before its deployment in the smart city environment. In addition, TA_k also deletes the private key r_{V_j} in order to thwart against privileged-insider attack.

VR_6: V_j chooses a random private key $pr_{V_j} \in Z_q^*$, computes the corresponding public key $PB_{V_j} = pr_{V_j} \cdot G$, and sets its private key as $\{pr_{V_j}, (pp_{V_{j0}}, pp_{V_{j1}})\}$. Finally, the tamper-resistant OBU of V_j contains the information $\{RID_{V_j}, R_{V_j}, Cert_{V_j}, pr_{V_j}, (pp_{V_{j0}}, pp_{V_{j1}}), Pub_{V_j}, PB_{V_j}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$.

3.2.2. RSU registration

TA_k is also responsible to register all the RSUs that are under its region. TA_k registers an RSU, say RSU_i which falls under a region of the smart city by executing following steps.

RR_1: TA_k generates a random unique identity ID_{RSU_i} for an RSU_i and calculates pseudo identity RID_{RSU_i} corresponding to its unique identity by $RID_{RSU_i} = H_0(ID_{RSU_i})$.

RR_2: TA_k chooses a random key $P_{RSU_i} \in Z_q^*$ and calculates its respective public key for RSU_i as $Pub_{RSU_i} = H_1(P_{RSU_i})$.

RR_3: TA_k also generates a random secret $r_{RSU_i} \in Z_q^*$ and computes $R_{RSU_i} = r_{RSU_i} \cdot G$. Furthermore, TA_k computes a certificate $Cert_{RSU_i}$ for RSU_i as $Cert_{RSU_i} = pr_{TA_k} + h(RID_{RSU_i} \parallel Pub_{RSU_i}) * r_{RSU_i} \pmod q$.

RR_4: TA_k then loads the credentials $\{RID_{RSU_i}, Pub_{RSU_i}, R_{RSU_i}, Cert_{RSU_i}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$ in the memory of RSU_i prior to its deployment in the smart city region. In addition, TA_k also deletes the private key r_{RSU_i} in order to thwart against privileged-insider attack.

RR_5: RSU_i chooses a random $pr_{RSU_i} \in Z_q^*$ and sets it as its secret key, and computes the public key $PB_{RSU_i} = pr_{RSU_i} \cdot G$. Finally, RSU_i contains the information $\{RID_{RSU_i}, PB_{RSU_i}, Pub_{RSU_i}, R_{RSU_i}, Cert_{RSU_i}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$.

3.3. Signing phase

Assume that a cluster C_k has n vehicles, say $V_1, V_2, \ldots, V_n$ and their RSU, say RSU_i, where the cluster can be formed using the methods described in [50,60].

This phase is performed by all the vehicles $V_1, V_2, \ldots, V_n$ of the cluster C_k. In this phase, each vehicle V_j broadcasts a hello message to all the neighboring vehicles and its corresponding RSU_i. We solve two purposes by executing this phase. First, each vehicle in the cluster knows about the fellow vehicles of the same cluster. Secondly, after receiving hello messages from all the vehicles of a cluster, RSU_i authenticates the vehicles in a batch before receiving any traffic related information from them. We describe authentication in the next section. The steps executed by any vehicle V_j of the cluster C_k coming under the scope of RSU_i is described as follows.

S_1: Vehicle V_j randomly chooses a random nonce $rn_{V_j} \in Z_q^*$ and generates the current timestamp value as TS_{V_j}. It then forms $ma_j = \{RID_{V_j}, rn_{V_j}, Cert_{V_j}, Pub_{V_j}, R_{V_j}, Pub_{TA_k}, PB_{V_j}, TS_{V_j}\}$.

S_2: To compute a signature on ma_j, vehicle V_j randomly chooses two fresh random nonces as $a_{V_j}, b_{V_j} \in Z_q^*$. It computes $Sig_{V_{j1}} = a_{V_j} \cdot (RID_{V_j} + Pub_{V_j})$ and $Sig_{V_{j2}} = b_{V_j} \cdot (pr_{V_j} \cdot G)$. Furthermore, it computes $h_{2j} = H_2(ma_j \parallel Sig_{V_{j1}})$ and $h_{3j} = H_3(ma_j \| Sig_{V_{j1}} \| Sig_{V_{j2}})$. Then, it uses the public parameter X_k published by TA_k to compute $Prox_{V_j} = H_4(X_k)$. After these computations, it finally generates a signature Sig_{V_j} on ma_j as $Sig_{V_j} = (a_{V_j} + h_{2j})\,(pp_{V_{j0}} + pp_{V_{j1}}) + (b_{V_j} + h_{3j})\, pr_{V_j} \cdot Prox_{V_j}$.

S_3: Lastly, vehicle V_j broadcasts the hello message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ to its cluster members including RSU_i.

3.4. Authentication phase

We describe authentication procedure in two instant scenarios: (a) vehicle to vehicle (V2V) authentication that is between two vehicles of the same cluster, and (b) vehicle to RSU (V2RSU) authentication that is performed between a vehicle and its RSU in a cluster. The following sections describe both the scenarios in detail.

3.4.1. V2V authentication

In V2V authentication, if any vehicle V_l receives the broadcasted hello message $Hello_{Msg_{V_j}}$ from its neighbor V_j, it authenticates and verifies the signature received in the message. The following steps are performed in this phase:

VA_1: Suppose V_l receives a hello message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ from its neighbor vehicle V_j. After reception of $Hello_{Msg_{V_j}}$ at time $TS^*_{V_j}$, V_l first verifies the received timestamp TS_{V_j} by the condition: $|TS^*_{V_j} - TS_{V_j}| \leq \Delta T$, where ΔT is the maximum transmission delay associated with a message. If the timestamp is valid, V_l further validates the certificate $Cert_{V_j}$ of V_j attached in the message by the condition: $Cert_{V_j} \cdot G = Pub_{TA_k} + h(RID_{V_j} \| Pub_{V_j} \| R_{V_j}) \cdot R_{V_j}$. If the certificate is legitimate, the next step is followed; otherwise, the phase is instantly terminated by V_l.

VA_2: V_l computes $h_{2j} = H_2(ma_j \parallel Sig_{V_{j1}})$ and $h_{3j} = H_3(ma_j \| Sig_{V_{j1}} \| Sig_{V_{j2}})$. Then, it uses the public parameter X_k published by the TA_k to compute $Prox_{V_j} = H_4(X_k)$.

VA_3: V_l verifies the signature Sig_{V_j} by using bilinear pairing through the following equation:

$$\begin{aligned} e(G, Sig_{V_j}) &= e(Pub_{TA_k}, Sig_{V_{j1}} + h_{2j} \cdot (RID_{V_j} + Pub_{V_j})) \\ &\quad \cdot e(Prox_{V_j}, Sig_{V_{j2}} + h_{3j} \cdot PB_{V_j}). \end{aligned}$$

If the signature is valid, V_l knows about the presence of legitimate vehicle V_j in its cluster.

Similarly, all other vehicles come to know about the other neighboring vehicles present in their cluster.

3.4.2. Batch authentication

Vehicle to RSU (V2RSU) authentication is the process where RSU_i receives the broadcasted hello message(s) $Hello_{Msg_{V_j}}$ from one or more vehicles V_j present under its cluster.

In order to save time and lower the computational overheads, we propose a batch authentication in which RSU_i authenticates all the vehicles $V_1, V_2, \ldots, V_n$ of a cluster C_k simultaneously. When all the vehicles broadcast their hello messages, RSU_i receives n signatures from vehicles $V_1, V_2, \ldots, V_n$ of the cluster C_k. Thus, the signatures received by RSU_i are $Sig_{V_1}, Sig_{V_2}, \ldots, Sig_{V_n}$ on $ma_1, ma_2, \ldots, ma_n$ from $V_1, V_2, \ldots, V_n$, respectively.

RSU_i performs the following steps to authenticate the received message(s):

BA_1: After reception of the messages $Hello_{Msg_{V_j}}$ at time $TS^*_{V_j}$, RSU_i first verifies the received timestamp TS_{V_j} by the condition: $|TS^*_{V_j} - TS_{V_j}| \leq \Delta T$ for all vehicles $V_1, V_2, \ldots, V_n$. If the timestamp validation is successful, RSU_i proceeds to the next step.

BA_2: To verify all the signatures, RSU_i chooses random $w_1, w_2, \ldots, w_n \in Z_q^*$. Then, for all $j = 1, 2, \ldots, n$, it calculates $h_{2j} = H_2(ma_j \parallel Sig_{V_{j1}})$ and $h_{3j} = H_3(ma_j \| Sig_{V_{j1}} \| Sig_{V_{j2}})$. Furthermore, it uses the public parameter X_k published by TA_k to compute $Prox_{V_j} = H_4(X_k)$.

BA_3: The validity of batch of signatures is checked by checking the validity of the following equation:

$$\begin{aligned} e(G, \sum_{j=1}^{n} w_j \cdot Sig_{V_j}) &= e(Pub_{TA_k}, \sum_{j=1}^{n} [w_j \cdot Sig_{V_{j1}} \\ &\quad + w_j h_{2j}(RID_{V_j} + Pub_{V_j})]) \cdot \\ &\quad e(Prox_{V_j}, \sum_{j=1}^{n} [w_j \cdot Sig_{V_{j2}} \\ &\quad + w_j \cdot h_{3j} \cdot PB_{V_j}]). \end{aligned}$$

If the verification is successful, RSU_i considers all its members $V_1, V_2, \ldots, V_n$ as authentic.

Signing Phase

Vehicle (V_j)	Road-side Unit (RSU_i)/Vehicle (V_l)
Select random $rn_{V_j}, a_{V_j}, b_{V_j} \in Z_q^*$ and current timestamp value TS_{V_j}. Compute $Sig_{V_{j1}} = a_{V_j} \cdot (RID_{V_j} + Pub_{V_j})$, $Sig_{V_{j2}} = b_{V_j} \cdot (pr_{V_j} \cdot G)$, $h_{2j} = H_2(ma_j \Vert Sig_{V_{j1}})$, $h_{3j} = H_3(ma_j \Vert Sig_{V_{j1}} \Vert Sig_{V_{j2}})$, $Prox_{V_j} = H_4(X_k)$, Generate signature as $Sig_{V_j} = (a_{V_j} + h_{2j})\ (pp_{V_{j0}} + pp_{V_{j1}}) + (b_{V_j} + h_{3j})\ pr_{V_j} \cdot Prox_{V_j}$. $\xrightarrow{Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}}$ (via public channel)	

V2V Authentication Phase

Vehicle (V_j)	Vehicle (V_l)
	Receive message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$. Check if $\lvert TS^*_{V_j} - TS_{V_j}\rvert \leq \Delta T$? If valid, verify if $Cert_{V_j} \cdot G = Pub_{TA_k} + h(RID_{V_j} \Vert Pub_{V_j} \Vert R_{V_j}) \cdot R_{V_j}$? If legitimate, compute $h_{2j} = H_2(ma_j \Vert Sig_{V_{j1}})$, $h_{3j} = H_3(ma_j \Vert Sig_{V_{j1}} \Vert Sig_{V_{j2}})$, $Prox_{V_j} = H_4(X_k)$. Verify if $e(G, Sig_{V_j}) = e(Pub_{TA_k}, Sig_{V_{j1}} + h_2(RID_{V_j} + Pub_{V_j}))\cdot$ $e(Prox_{V_j}, Sig_{V_j2} + h_3 \cdot PB_{V_j})$? If it is valid, V2V authentication is successful.

Batch Authentication Phase

Vehicles $V_1, V_2, \cdots, V_n$	Road-side Unit (RSU_i)
	Check if $\lvert TS^*_{V_j} - TS_{V_j}\rvert \leq \Delta T$? for all vehicles $V_1, V_2, \cdots, V_n$. If valid, select random $w_1, w_2, \cdots, w_n \in Z_q^*$. Compute $h_{2j} = H_2(ma_j \Vert Sig_{V_{j1}})$, $h_{3j} = H_3(ma_j \Vert Sig_{V_{j1}} \Vert Sig_{V_{j2}})$, $Prox_{V_j} = H_4(X_k)$, for all $j = 1, 2, \cdots, n$. Verify if $e(G, \sum_{j=1}^{n} wj\ Sig_{V_j}) = e(Pub_{TA_k}, \sum_{j=1}^{n} wj\ Sig_{V_{j1}}$ $+ wj\ h_{2j}(RID_{V_j} + Pub_{V_j}))\cdot e(Prox_{V_j}, \sum_{j=1}^{n} wj\ Sig_{V_{j2}} + wj \cdot h_{3j} \cdot PB_{V_j})$? If it is valid, batch authentication is successful, and all the vehicles $V_1, V_2, \cdots, V_n$ in the cluster are authenticated by (RSU_i).

Group Key Management Phase

Vehicles (V_j)	Road-side Unit (RSU_i)
	Generate a rn_{RSU_i}, $sk \in Z_q^*$ timestamp value TS_{RSU_i}. Compute $RN_{RSU_i} = rn_{RSU_i}.G$. Generate group key $GK_k = h(RID_{V_1} \Vert RID_{V_2} \Vert \cdots \Vert RID_{V_n}$ $\Vert rn_{V_1} \Vert rn_{V_2} \Vert \cdots \Vert rn_{V_n} \Vert Cert_{V_1} \Vert Cert_{V_2} \Vert \cdots$ $\Vert Cert_{V_n} \Vert RID_{RSU_i} \Vert rn_{RSU_i} \Vert Cert_{RSU_i} \Vert pr_{RSU_i} \Vert sk)$, Encrypted group key $E_{PB_{V_j}}(GK_k \Vert rn_{V_j} \Vert TS_{RSU_i})$ Form message $mb_j = \{TS_{RSU_i}, Cert_{RSU_i}, Pub_{RSU_i}$, $PB_{RSU_i}, R_{RSU_i}, RN_{RSU_i}, Pub_{TA_k}, rn_{V_j}, E_{PB_{V_j}}(GK_k \Vert rn_{V_j} \Vert TS_{RSU_i})\}$, signature $Sig_{RSU_i,V_j} = pr_{RSU_i} + h(mb_j \Vert RN_{RSU_i} \Vert Cert_{V_j} \Vert RID_{V_j}$ $\Vert TS_{V_j} \Vert TS_{RSU_i}) * rn_{RSU_i} \pmod q$. $\xleftarrow{Res_{RSU_i,V_j} = \{mb_j, Sig_{RSU_i,V_j}\}}$
Check if $\lvert TS^*_{RSU_i} - TS_{RSU_i}\rvert \leq \Delta T$? If valid check $Cert_{RSU_i} \cdot G = Pub_{TA_k} + h(RID_{RSU_i}$ $\Vert Pub_{RSU_i}) \cdot R_{RSU_i}$? Verify the signature by $Sig_{RSU_i,V_j} \cdot G = PB_{RSU_i} + h(mb_j$ $\Vert RN_{RSU_i} \Vert Cert_{V_j} \Vert RID_{V_j} \Vert TS_{V_j} \Vert TS_{RSU_i}) \cdot RN_{RSU_i}$? If valid extract group key by $(GK_k \Vert rn'_{V_j} \Vert TS'_{RSU_i})$ $= D_{pr_{V_j}}[E_{PB_{V_j}}(GK_k \Vert rn_{V_j} \Vert TS_{RSU_i})]$ Check if $rn'_{V_j} = rn_{V_j}$ and $TS'_{RSU_i} = TS_{RSU_i}$? If both are valid, the group key GK_k is authentic.	

All vehicles $V_1, V_2, \cdots, V_n$ and RSU in their cluster share the group key GK_k

Fig. 2. Signing, V2V authentication, batch authentication and group key management phases in BBAS-IoV.

3.5. Group key management phase

The above described batch authentication phase allows RSU_i to verify all the members V_j of the group simultaneously. After all the members of the cluster are authenticated by RSU_i, it generates a group key GK_k for the vehicles that are members of the cluster C_k. The group key GK_k is shared with all the vehicles of the group and RSU_i which is then used for further communication in the network. The steps to generate a group key GK_k executed by RSU_i are as follows.

GK_1: RSU_i initially generates a random nonce $rn_{RSU_i} \in Z_q^*$, a random secret key $sk \in Z_q^*$ and the current timestamp value as TS_{RSU_i}, and computes public $RN_{RSU_i} = rn_{RSU_i}.G$.

GK_2: RSU_i then derives the group key GK_k for the vehicles V_1, $V_2, \ldots, V_n$ which are members of the cluster C_k using its own private key pr_{RSU_i}, the generated random secret key sk and other information received from its cluster members (vehicles) as $GK_k = h(RID_{V_1} \Vert RID_{V_2} \Vert \cdots \Vert RID_{V_n} \Vert rn_{V_1} \Vert rn_{V_2} \Vert \cdots \Vert rn_{V_n} \Vert Cert_{V_1} \Vert Cert_{V_2} \Vert \cdots \Vert Cert_{V_n} \Vert RID_{RSU_i} \Vert rn_{RSU_i} \Vert Cert_{RSU_i} \Vert pr_{RSU_i} \Vert sk)$.

GK_3: RSU_i further computes the encrypted group key for all legitimate vehicles of the cluster C_k using each vehicle's public key PB_{V_j} as $EP_{PB_{V_j}}(GK_k \| rn_{V_j} \| TS_{RSU_i})$. Finally, for each vehicle V_j in its cluster, RSU_i forms $mb_j = \{TS_{RSU_i}, Cert_{RSU_i}, Pub_{RSU_i}, PB_{RSU_i}, R_{RSU_i}, RN_{RSU_i}, Pub_{TA_k}, rn_{V_j}, EP_{PB_{V_j}}(GK_k \| rn_{V_j} \| TS_{RSU_i})\}$ and the signature on mb_j for V_j as $Sig_{RSU_i,V_j} = pr_{RSU_i} + h(mb_j \| RN_{RSU_i} \| Cert_{V_j} \| RID_{V_j} \| TS_{V_j} \| TS_{RSU_i}) * rn_{RSU_i} \pmod q$, and then sends the response message $Res_{RSU_i,V_j} = \{mb_j, Sig_{RSU_i,V_j}\}$ to V_j.

GK_4: When a vehicle V_j receives the message Res_{RSU_i,V_j}, it first checks the validity of the received timestamp TS_{RSU_i} by the condition $|TS^*_{RSU_i} - TS_{RSU_i}| \le \Delta T$, where $TS^*_{RSU_i}$ denotes the received time of the message Res_{RSU_i,V_j}. If it is valid, V_j checks the validity of the certificate $Cert_{RSU_i}$ of RSU_i by $Cert_{RSU_i} \cdot G = Pub_{TA_k} + h(RID_{RSU_i} \| Pub_{RSU_i}) \cdot R_{RSU_i}$. Upon successful verification of $Cert_{RSU_i}$, V_j also checks the validity of the signature Sig_{RSU_i,V_j} on the received mb_j by the condition: $Sig_{RSU_i,V_j} \cdot G = PB_{RSU_i} + h(mb_j \| RN_{RSU_i} \| Cert_{V_j} \| RID_{V_j} \| TS_{V_j} \| TS_{RSU_i}) \cdot RN_{RSU_i}$. If the signature is valid, V_j considers RSU_i as legitimate and proceeds to the next step.

GK_5: V_j extracts the group key GK_k by decrypting the encrypted group key using its own private key pr_{V_j} as $(GK_k \| rn'_{V_j} \| TS'_{RSU_i}) = DP_{pr_{V_j}}[EP_{PB_{V_j}}(GK_k \| rn_{V_j} \| TS_{RSU_i})]$. V_j then checks if $rn'_{V_j} = rn_{V_j}$ and $TS'_{RSU_i} = TS_{RSU_i}$. If both are valid, V_j stores this group key GK_k in its memory. Similarly, all other vehicles of the cluster extract and store the same group key GK_k in their memory.

It is also worth noticing that each vehicle in the cluster can use the same group key GK_k for secure communication with its neighbor vehicles, as well as for the secure communication with their RSU_i.

The signing, V2V authentication, batch authentication and group key management phases are summarized in Fig. 2.

3.6. Incorporating batch authentication in blockchain formation

Through the signing, V2V authentication, batch authentication and group key management phases in the proposed scheme (BBAS-IoV), all the vehicles (V_j) and their RSU_i in a cluster C_k establish a shared group key GK_k. Using this authentic group key GK_k, a vehicle V_j can send the data securely to RSU_i and also RSU_i can also send the data securely to V_j.

We consider the following two types of transactions (Tx_{V_j,RSU_i}, Tx_{RSU_i,V_j}):

- **Transactional data from a vehicle V_j to RSU_i**: This transaction has the format as follows. $Tx_{V_j,RSU_i} = \{RID_{V_j}, Timestamp_{V_j}, Data_{V_j}, PB_{V_j}, Sig_{Data_{V_j}} = ECDSA.Sig_{pr_{V_j}}[RID_{V_j}, Timestamp_{V_j}, Data_{V_j}, PB_{V_j}]\}$, where RID_{V_j}, $Timestamp_{V_j}$, $Data_{V_j}$, PB_{V_j} and $Sig_{Data_{V_j}}$ are the pseudo identity of V_j, transaction creation time, information (data), public key of V_j and signature using the ECDSA signature generation algorithm with the help of the private key pr_{V_j} of V_j, respectively.
- **Transactional data from RSU_i to a vehicle V_j**: This kind of transaction has the following format. $Tx_{RSU_i,V_j} = \{RID_{RSU_i}, Timestamp_{RSU_i}, Data_{RSU_i}, PB_{RSU_i}, Sig_{Data_{RSU_i}} = ECDSA.Sig_{pr_{RSU_i}}[RID_{RSU_i}, Timestamp_{RSU_i}, Data_{RSU_i}, PB_{RSU_i}]\}$, where RID_{RSU_i}, $Timestamp_{RSU_i}$, $Data_{RSU_i}$, PB_{RSU_i} and $Sig_{Data_{RSU_i}}$ represent the pseudo identity of RSU_i, transaction creation time, information (data), public key of RSU_i and signature using the ECDSA signature generation algorithm with the help of the private key pr_{RSU_i} of RSU_i, respectively.

In order to create a block, and then to verify and add that block in the blockchain, the following steps are executed:

- Each vehicle V_j in a cluster C_k sends the encrypted transactions $EC_{GK_k}[Tx_{V_j,RSU_i}]$ securely to its RSU_i by means of encrypting Tx_{V_j,RSU_i} using the AES-128 symmetric encryption with the help of the already established group key GK_k during the batch authentication phase. On the other side, RSU_i also sends the encrypted transactions $EC_{GK_k}[Tx_{RSU_i,V_j}]$ securely to its cluster member vehicles V_j by encrypting Tx_{V_j,RSU_i} using the AES-128 symmetric encryption with the help of the same established group key GK_k with its members.
- Next, RSU_i decrypts the encrypted transactions $EC_{GK_k}[Tx_{V_j,RSU_i}]$ received from its member vehicles V_j using the group key GK_k to obtain $Tx_{V_j,RSU_i} = DC_{GK_k}[EC_{GK_k}[Tx_{V_j,RSU_i}]]$.
- Assume that RSU_i creates a pool of transactions, say Tx_l, where $Tx_l \in \{Tx_{V_j,RSU_i}, Tx_{RSU_i,V_j}\}$. Let a block, say $Block$, be formed using n_t transactions Tx_l from the transactions pool. For this purpose, RSU_i generates a compact signature on these n_t transactions using the ECDSA signature generation algorithm with the help of its private key pr_{RSU_i} as $Sig_{Block} = ECDSA.Sig_{pr_{RSU_i}}[\{Tx_l | l = 1, 2, \dots, n_t\}]$. RSU_i then sends $\langle \{Tx_l | l = 1, 2, \dots, n_t\}, Sig_{Block} \rangle$ to its nearby fog server, say FS_k for mining purpose in the public blockchain. Note that we do not require here to send encrypted transactions to FS_k because the blockchain is public and also in between the communication the transactions cannot be modified by an adversary as there is an attachment of the compact signature. The signature cannot be created by the adversary as it requires the private key of RSU_i.
- After receiving n_t transactions with their compact signature, FS_k creates a *partial block*, say $ParBlock$. The structure of such a block is shown in Fig. 3.
- FS_k then sends the created partial block shown in Fig. 3 to a cloud server, say CS_m for mining the full block. The full block, say $FullBlock$ created by CS_m is shown in Fig. 4. $FullBlock$ contains the following fields:
  - **Block header:** This has the following components:
    - $BVer$: It presents the block version, which is a unique serial number.
    - $PBHash$: It is the hash of the previous block in the public blockchain.
    - TS_{Block}: It denotes the time when the block, $Block$ has been created by FS_k.
    - MTR_{Block}: This is the Merkle tree root for all the n_t transactions $\{Tx_l | l = 1, 2, \dots, n_t\}$ stored in the block payload.
    - PB_{RSU_i}: This is the public key of RSU_i, which is needed for compact signature verification of the block.
  - **Block payload**: The payload contains $\{Tx_l | l = 1, 2, \dots, n_t\}$.
  - Sig_{Block}: This field is the compact signature on the block transactions, $\{Tx_l | l = 1, 2, \dots, n_t\}$.
  - $CBHash$: This is the hash of the current entire block including Sig_{Block}.
- CS_m applies the voting based consensus algorithm as described in Algorithm 1, which is based on the widely-used "Practical Byzantine Fault Tolerance (PBFT) consensus algorithm" [26]. We assume that the each cloud server CS_m has its own ECC-based private–public keys pair (pr_{CS_m}, $Pub_{CS_m} = pr_{CS_m} \cdot G$), where $pr_{CS_m} \in Z_q^*$ is the private key and Pub_{CS_m} is made public to all other cloud servers in the P2P CS network. At first, a leader, say L_{CS} is selected among a group of Peer-to-Peer (P2P) cloud servers network in the blockchain center using the existing leader selection algorithm provided in [63]. L_{CS} later sends the block $FullBlock$ to its peer nodes. Next, via block verification the block $FullBlock$ is committed and added in the public blockchain.

Block Header	
Timestamp	TS_{Block}
Merkle Tree Root	MTR_{Block}
Signer's Public Key	PB_{RSU_i}
Block Payload	
List of n_t Transactions	$\{Tx_l \mid l = 1, 2, \cdots, n_t\}$
ECDSA signature on Block	Sig_{Block}

Fig. 3. Structure of a partial block, *ParBlock* by a fog server FS_k.

Block Header	
Block Version	$BVer$
Previous Block Hash	$PBHash$
Timestamp	TS_{Block}
Merkle Tree Root	MTR_{Block}
Signer's Public Key	PB_{RSU_i}
Block Payload	
List of n_t Transactions	$\{Tx_l \mid l = 1, 2, \cdots, n_t\}$
ECDSA signature on Block	Sig_{Block}
Current Block Hash	$CBHash$

Fig. 4. Structure of a full block, *FullBlock* by a cloud server CS_m.

3.7. AI-based secure big data analytics using blockchain

For smart applications such as smart city, data trading, device personalization and product manufacturing, IoV produces huge amount of data everyday. The data is so enormous and complex that it cannot be processed by present traditional data processing solutions. Blockchain is one such kind of technology that stores huge amount of data within distributed ledger. The centerpieces of the blockchaining like immutability, decentralization, and transparency make it the one of the potential choices for such scenarios. But due to decentralization, public blockchains are prone to various security issues, such as majority attack, fake identity generation through Sybil attack to change or control the consensus voting, double-spending attacks, client-side security threats, and mining pool attacks [64,65].

To address the above concerns, the latest trends and revolutions in technologies incorporate Artificial Intelligence (AI)/Machine Learning (ML) in blockchaining to produce a secure, efficient and intelligent blockchain based system. ML is a branch of science that deals in making a computer learn through past experiences, supervised or unsupervised learning without any program or human interventions. It possesses high power data processing capabilities. The combination of both the technologies works by keeping blockchain as an integral part of the system, and adopts ML/AI algorithms to upgrade efficiency and security of an application. ML can be used to detect any data security issue and monitor blockchain system, for predictions and decision making [66]. Blockchain ensures the security of data by building in confidence where as AI/ML identifies untrusted nodes based on past learning and experiences. The technology sounds promising in the cases where deterministic algorithms are not up to the mark in terms of security and efficiency [67].

In IoV paradigm, the AI/ML algorithm works by taking huge amount of data, called as Big data, collected from various deployed sensors, vehicles and other entities stored on a blockchain as input, analyzing it statistically by applying Big data analytic without the use of human intervention, and produces predictions or decisions as an output quickly and accurately. Big data analytics basically enhances the capability of final decision making on the trusted huge volume of data. The combination of Big data, blockchain and AI/ML technologies addresses the issues like identifying blockchain breaches automatically, detects the cybercrime activities in bitcoin system, and enhances the capabilities of an application by making it faster and smarter and building exactly trained ML models to bring out advantages [68].

Algorithm 1 Consensus for block verification and addition in blockchain

Input: A full block, $FullBlock = \{BVer, PBHash, TS_{Block}, MTR_{Block}, PB_{RSU_i}, \{Tx_l \mid l = 1, 2, \cdots, n_t\}, Sig_{Block}, CBHash\}$; private–public key pairs $(pr_{CS_m}, Pub_{CS_m} = pr_{CS_m} \cdot G)$ of all other cloud servers CS_m in the P2P CS network; $n_{f_{CS}}$: number of faulty cloud servers (nodes) in the P2P CS network.
Output: Commitment for block addition.

```
1: $L_{CS}$ generates a random number $rn_{L_{CS}} \in Z_q^*$, a current timestamp $TS_{L_{CS}}$ and a voting request $VReq_{L_{CS}}$.
2: for each peer cloud server node $CS_m$ do
3:     $L_{CS}$ encrypts $rn_{L_{CS}}$, $TS_{L_{CS}}$ and $VReq_{L_{CS}}$ using the ECC-based encryption algorithm with the help of the public key $Pub_{CS_m}$ of $CS_m$ as $EncVreq = EP_{Pub_{CS_m}}[rn_{L_{CS}}, TS_{L_{CS}}, VReq_{L_{CS}}]$ and sends a block verification request message $\{FullBlock, EncVreq, TS_{L_{CS}}\}$ to $CS_m$ via open channel.
4: end for
5: for each follower node $CS_m$ in the P2P CS network do
6:     Let the message $\{FullBlock, EncVreq, TS_{L_{CS}}\}$ be received at time $TS^*_{L_{CS}}$ by $CS_m$.
7:     $CS_m$ checks validity of $TS_{L_{CS}}$ by the condition: $|TS^*_{L_{CS}} - TS_{L_{CS}}| \le \Delta T$.
8:     if timestamp is valid then
9:         $CS_m$ decrypts $EncVreq$ using its own private key $pr_{CS_m}$ to retrieve $[rn'_{L_{CS}}, TS'_{L_{CS}}, VReq'_{L_{CS}}] = DP_{pr_{CS_m}}[EncVreq]$.
10:        if $TS'_{L_{CS}} = TS_{L_{CS}}$ then
11:            $CS_m$ verifies $MTR_{Block}$, $Sig_{Block}$ and $CBHash$ on the block $FullBlock$.
12:            if all the verifications by $CS_m$ are successful then
13:                $CS_m$ sends a voting response cum block verification status message $\{EP_{Pub_{L_{CS}}}[rn'_{L_{CS}}, VRep_{CS_m}, VerStat_{CS_m}]\}$ to the leader $L_{CS}$, where $VRep_{CS_m}$ and $VerStat_{CS_m}$ are the voting response corresponding to $VReq'_{L_{CS}}$ and block verification status, respectively.
14:            end if
15:        end if
16:    end if
17: end for
18: Initialize $VCount = 0$, where $VCount$ represents the number of valid votes.
19: for each voting response cum block verification status message $\{EP_{Pub_{L_{CS}}}[rn'_{L_{CS}}, VRep_{CS_m}, VerStat_{CS_m}]\}$ from the follower peer nodes $CS_m$ do
20:    $L_{CS}$ decrypts the message using its own private key $pr_{L_{CS}}$ to retrieve $[rn^*_{L_{CS}}, VRep_{CS_m}, VerStat_{CS_m}] = DP_{pr_{L_{CS}}}[EP_{Pub_{L_{CS}}}[rn'_{L_{CS}}, VRep_{CS_m}, VerStat_{CS_m}]]$.
21:    if $((rn^*_{L_{CS}} = rn_{L_{CS}})$ and $(VRep_{CS_m} = valid)$ and $(VerStat_{CS_m} = valid))$ then
22:        Set $VCount = VCount + 1$.
23:    end if
24: end for
25: if $(VCount \ge 2 * n_{f_{CS}} + 1)$ then
26:    Add the block $FullBlock$ to the blockchain.
27:    Broadcast commitment message to the P2P CS network.
28: end if
```

There exists a potential attack in AI/ML security, known as "data poisoning attack" [64], in which malicious data can be injected by illegal users to delude the "training data sets for puzzling AI/ML algorithms". This may lead to "incorrect predications" on data sets. As a result, such an attack may play a crucial factor for damaging the businesses and organizations for "financial terms". In proposed BBAS-IoV, the transactions $\{(Tx_{V_j,RSU_i}, Tx_{RSU_i,V_j})\}$ stored in the public blockchain are genuine and authentic as the signature verification and block verification assure the legitimacy of these transactions. This suggests that data poising attacks are prevented from injecting malicious transactions in the blocks. Thus, the authentic and genuine data containing in the blocks in the blockchain in the proposed BBAS-IoV are then useful for correct predictions on the data sets.

3.8. Dynamic nodes addition phase

The phase allows the respective TA, say TA_k to dynamically register new vehicles and new RSUs in a particular region of the smart city.

3.8.1. New vehicle addition

Suppose a new vehicle V_{new} wishes to join the network that is managed by TA_k. The steps involved in the phase are described as follows.

VA_1: TA_k generates a random unique identity ID_{new} for V_{new}. It then calculates pseudo identity $RID_{V_{new}}$ corresponding to its unique identity by $RID_{V_{new}} = H_0(ID_{new})$.

VA_2: TA_k chooses a random key $P_{V_{new}} \in Z_q^*$ for vehicle V_{new} to calculate the public key as $Pub_{V_{new}} = H_1(P_{V_{new}})$. It also calculates the partial private keys pair for vehicle V_{new} as $pp_{V_{new_0}} = pr_{TA_k} \cdot RID_{V_{new}}$ and $pp_{V_{new_1}} = pr_{TA_k} \cdot Pub_{V_{new}}$.

VA_3: TA_k generates a random secret $r_{V_{new}} \in Z_q^*$ and computes public $R_{V_{new}} = r_{V_{new}} \cdot G$ for V_{new}. Furthermore, TA_k computes a certificate $Cert_{V_{new}}$ for vehicle V_{new} as $Cert_{V_{new}} = pr_{TA_k} + h(RID_{V_{new}} \| Pub_{V_{new}} \| R_{V_{new}}) * r_{V_{new}} \pmod q$.

VA_4: TA_k loads the credentials $\{RID_{V_{new}}, R_{V_{new}}, Cert_{V_{new}}, (pp_{V_{new_0}}, pp_{V_{new_1}}), Pub_{V_{new}}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$ in the tamper-resistant on-board unit (OBU) of the vehicle V_{new} before its deployment in the smart city environment. In addition, TA_k also deletes the private key $r_{V_{new}}$ in order to thwart against privileged-insider attack.

VA_5: V_{new} chooses a random private key $pr_{V_{new}} \in Z_q^*$, computes the corresponding public key $PB_{V_{new}} = pr_{V_{new}} \cdot G$, and sets its private key as $\{pr_{V_{new}}, (pp_{V_{new_0}}, pp_{V_{new_1}})\}$. Finally, the tamper-resistant OBU of V_{new} contains the information $\{RID_{V_{new}}, R_{V_{new}}, Cert_{V_{new}}, pr_{V_{new}}, (pp_{V_{new_0}}, pp_{V_{new_1}}), Pub_{V_{new}}, PB_{V_{new}}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$.

3.8.2. New RSU addition

TA_k registers a new RSU, say RSU_{new} under a region by executing following steps.

RA_1: TA_k generates a random unique identity $ID_{RSU_{new}}$ for an RSU_{new} and calculates pseudo identity $RID_{RSU_{new}}$ corresponding to its unique identity by $RID_{RSU_{new}} = H_0(ID_{RSU_{new}})$.

RA_2: TA_k chooses a random key $P_{RSU_{new}} \in Z_q^*$ and calculates its respective public key for RSU_i as $Pub_{RSU_{new}} = H_1(P_{RSU_{new}})$.

RA_3: TA_k also generates a random secret $r_{RSU_{new}} \in Z_q^*$ and computes $R_{RSU_{new}} = r_{RSU_{new}} \cdot G$. Furthermore, TA_k computes a certificate $Cert_{RSU_{new}}$ for RSU_{new} as $Cert_{RSU_{new}} = pr_{TA_k} + h(RID_{RSU_{new}} \| Pub_{RSU_{new}}) * r_{RSU_{new}} \pmod q$.

RA_4: TA_k then loads the credentials $\{RID_{RSU_{new}}, Pub_{RSU_{new}}, R_{RSU_{new}}, Cert_{RSU_{new}}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$ in the memory of RSU_{new} prior to its deployment in the smart city region. In addition, TA_k also deletes the private key $r_{RSU_{new}}$ in order to thwart against privileged-insider attack.

RA_5: RSU_{new} chooses a random $pr_{RSU_{new}} \in Z_q^*$ and sets it as its secret key, and computes the public key $PB_{RSU_{new}} = pr_{RSU_{new}} \cdot G$. Finally, RSU_{new} has the credentials $\{RID_{RSU_{new}}, PB_{RSU_{new}}, Pub_{RSU_{new}}, R_{RSU_{new}}, Cert_{RSU_{new}}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$.

4. Security analysis

We first provide the correctness proof of our V2V authentication and batch authentication phases that are described in Section 3.4. Next, we prove the group key security of the proposed scheme during the batch authentication phase under the widely-accepted “Real-Or-Random (ROR) random oracle model” [29]. We also provide the “informal (non-mathematical) security analysis” to show the robustness of the proposed scheme against other potential attacks.

4.1. Correctness proof

In this section, we provide the correctness proof of our V2V authentication and batch authentication phases that are described in Section 3.4 using Theorems 1 and 2.

Theorem 1. *During the V2V authentication phase described in Section 3.4.1, a vehicle V_l in a cluster C_k authenticates its other neighbor vehicle V_j, if the received message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ is valid.*

Proof. A vehicle V_l receives the broadcasted hello message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ from another vehicle V_j. It authenticates and verifies the signature received in the message by the equation:

$$e(G, Sig_{V_j}) = e(Pub_{TA_k}, Sig_{V_{j1}} + h_{2j} \cdot (RID_{V_j} + Pub_{V_j})) \cdot e(Prox_{V_j}, Sig_{V_{j2}} + h_{3j} \cdot PB_{V_j}),$$

where $Sig_{V_j} = (a_{V_j} + h_{2j}) \cdot (pp_{V_{j0}} + pp_{V_{j1}}) + (b_{V_j} + h_{3j})\, pr_{V_j} \cdot Prox_{V_j}$, $Sig_{V_{j1}} = a_{V_j} \cdot (RID_{V_j} + Pub_{V_j})$ and $Sig_{V_{j2}} = b_{V_j} \cdot (pr_{V_j} \cdot G)$.

Now,

$$\begin{aligned}
& e(G, Sig_{V_j}) \\
&= e(G, (a_{V_j} + h_{2j}) \cdot (pp_{V_{j0}} + pp_{V_{j1}}) + (b_{V_j} + h_{3j}) pr_{V_j} \cdot Prox_{V_j}) \\
&= e(G, (a_{V_j}(pp_{V_{j0}} + pp_{V_{j1}}) + h_{2j} \cdot (pp_{V_{j0}} + pp_{V_{j1}})) \cdot e(G, (b_{V_j} \cdot pr_{V_j} \cdot Prox_{V_j} + h_{3j} \cdot pr_{V_j} \cdot Prox_{V_j}))) \\
&= e(G, (a_{V_j}(pr_{TA_k} \cdot RID_{V_j} + pr_{TA_k} \cdot Pub_{V_j}) + h_{2j} \cdot (pr_{TA_k} \cdot RID_{V_j} + pr_{TA_k} \cdot Pub_{V_j})) \cdot e(Prox_{V_j}, (b_{V_j} \cdot pr_{V_j} \cdot G + h_{3j} \cdot pr_{V_j} \cdot G))) \\
&= e(pr_{TA_k} \cdot G, (a_{V_j}(RID_{V_j} + Pub_{V_j}) + h_{2j} \cdot (RID_{V_j} + Pub_{V_j})) \cdot e(Prox_{V_j}, (b_{V_j} \cdot pr_{V_j} \cdot G + h_{3j} \cdot pr_{V_j} \cdot G))) \\
&= e(Pub_{TA_k}, Sig_{V_{j1}} + h_{2j} \cdot (RID_{V_j} + Pub_{V_j})) \cdot e(Prox_{V_j}, Sig_{V_j}2 + h_{3j} \cdot PB_{V_j}).
\end{aligned}$$

Since the above equation is valid, V_l authenticates V_j. Hence, the theorem follows. □

Theorem 2. *During the batch authentication phase described in Section 3.4.2, a road-side unit RSU_i authenticates its all n member vehicles $V_1, V_2, \dots, V_n$ in a cluster C_k provided that all the received hello messages $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$, $(j = 1, 2, \dots, n)$, from the vehicles are valid.*

Proof. RSU_i receives n signatures $Sig_{V_1}, Sig_{V_2}, \dots, Sig_{V_n}$ on $ma_1, ma_2, \dots, ma_n$ from the vehicles $V_1, V_2, \dots, V_n$ in the cluster C_k, respectively. It authenticates all the vehicles simultaneously in a batch by using the following equation:

$$e(G, \sum_{j=1}^{n} w_j \cdot Sig_{V_j}) = e(Pub_{TA_k}, \sum_{j=1}^{n} [w_j \cdot Sig_{V_{j1}} + w_j h_{2j} \cdot (RID_{V_j} + Pub_{V_j})]) \cdot$$

$$e(Prox_{V_j}, \sum_{j=1}^{n}[w_j \cdot Sig_{V_{j2}} + w_j \cdot h_{3j} \cdot PB_{V_j}]).$$

Now, we have,

$$\begin{aligned}
& e(G, \sum_{j=1}^{n} w_j.Sig_{V_j}) \\
&= e(G, \sum_{j=1}^{n} w_j((a_{V_j} + h_{2j}) \cdot (pp_{V_{j0}} + pp_{V_{j1}}) \\
&\quad + (b_{V_j} + h_{3j}).pr_{V_j} \cdot Prox_{V_j})) \\
&= e(G, \sum_{j=1}^{n} w_j(a_{V_j}(pp_{V_{j0}} + pp_{V_{j1}}) + (h_{2j} \cdot (pp_{V_{j0}} + pp_{V_{j1}})))) \cdot \\
&\quad e(G, \sum_{j=1}^{n} w_j(b_{V_j}.pr_{V_j}.Prox_{V_j} + h_{3j}.pr_{V_j}.Prox_{V_j})) \\
&= e(G, \sum_{j=1}^{n} w_j(a_{V_j}(pr_{TA_k} \cdot RID_{V_j}) + pr_{TA_k} \cdot Pub_{V_j}) + \\
&\quad (h_{2j} \cdot (pr_{TA_k} \cdot RID_{V_j}) + pr_{TA_k} \cdot Pub_{V_j})) \cdot \\
&\quad e(Prox_{V_j}, \sum_{j=1}^{n} w_j(b_{V_j}.pr_{V_j}.G + h_{3j}.pr_{V_j}.G)) \\
&= e(pr_{TA_k}.G. \sum_{j=1}^{n} w_j(a_{V_j}(RID_{V_j} + Pub_{V_j}) \\
&\quad + (h_{2j} \cdot (RID_{V_j} + Pub_{V_j})))) \cdot \\
&\quad e(Prox_{V_j}, \sum_{j=1}^{n} w_j(b_{V_j}.pr_{V_j}.G + h_{3j}.pr_{V_j}.G)) \\
&= e(Pub_{TA_k}, \sum_{j=1}^{n}[w_j.Sig_{V_{j1}} + w_j.h_{2j} \cdot (RID_{V_j} + Pub_{V_j})]) \cdot \\
&\quad e(Prox_{V_j}, \sum_{j=1}^{n}[w_j.Sig_{V_{j2}} + w_j \cdot h_{3j} \cdot PB_{V_j}]).
\end{aligned}$$

Since the above equation is valid, RSU_i authenticates all n member vehicles V_1, V_2, …, V_n in its cluster C_k. Thus, the theorem follows. □

4.2. Formal security analysis using ROR model

This section describes the formal security analysis under the widely-accepted Real-Or-Random (ROR) model [29] to prove the group key security in the proposed scheme (BBAS-IoV).

In the ROR model, an adversary, say $\mathcal{A}$, has access to different types of queries as listed in Table 3. In the following, we now discuss briefly the components associated with the ROR model.

- **Participants.** There are two types of participants, namely vehicles (V_j) and an RSU_i in a cluster C_k, which are involved during the signing, V2V authentication and batch authentication phases. Let $\Pi_{V_j}^{l_1}$ and $\Pi_{RSU_i}^{l_2}$ represent the l_1^{th} and l_2^{th} instances of V_j and RSU_i, respectively. These are also called as the "random oracles".
- **Accepted state.** We say an instance Π^l is in an "accepted state", if it gets the last authenticated communicated message. Assume that all the communicated sent and received messages are concatenated in sequentially order. We call these as a "session identification, say sid of Π^l for a running current session".
- **Partnering.** We say two instances (Π^{l_1} and Π^{l_2}) are "partners to each other" if the following criteria are satisfied:
  - Both Π^{l_1} and Π^{l_2} are in "accepted states".
  - Both Π^{l_1} and Π^{l_2} share the same sid. They also require to "mutually authenticate each other".
  - Both Π^{l_1} and Π^{l_2} are also "mutual partners of each other".

Table 3
Queries and their purposes.

Query	Purpose
$Execute(\Pi_{V_j}^{l_1}, \Pi_{RSU_i}^{l_2})$	This query helps $\mathcal{A}$ to eavesdrop (intercept) the messages communicated among a group of vehicles V_j and their RSU_i
$Reveal(\Pi^l)$	Under this query, the group key GK_k shared between Π^l and its respective partner is known to $\mathcal{A}$
$Test(\Pi^l)$	This query allows $\mathcal{A}$ to appeal Π^l for GK_k and Π^l provides a "random outcome of a flipped unbiased coin, say c"

- **Freshness.** We say $\Pi_V^{l_1}$ or $\Pi_{RSU}^{l_2}$ is *fresh* if the established group key GK_k among the vehicles V_j and the RSU_i is not disclosed to the adversary $\mathcal{A}$ using the Reveal(Π^l) query listed in Table 3.
- **Hash function.** A "one-way cryptographic hash function $H(\cdot)$" is modeled as a random oracle, say $\mathcal{H}$.

Prior to proving Theorem 3, we first define the computational "Elliptic Curve Discrete Logarithm Problem (ECDLP)" and the "semantic security" of our proposed BBAS-IoV.

Definition 1 (*Elliptic Curve Discrete Logarithm Problem (ECDLP)*). Let a non-singular elliptic curve $E_q(u, v)$ be of the form: $y^2 = x^3 + ux + v \pmod q$ such that $4u^3 + 27v^2 \neq 0 \pmod q$. Given two points $P, Q \in E_q(u, v)$, the task is to find an integer $k \in Z_q^*$ such that $Q = k \cdot P$, where k is called the discrete logarithm to the base P.

ECDLP becomes intractable when the chosen prime q is large. For example, q should be chosen at least 160 bits to make ECDLP computationally infeasible.

Definition 2 (*Semantic Security*). Let $Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly})$ denote the "advantage of an adversary $\mathcal{A}$ running in polynomial time t_{poly} in breaking the semantic security of the proposed BBAS-IoV in order to derive the group key GK_k among a group of vehicles V_j and an RSU_i in a cluster C_k. Then, $Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly}) = |2Pr[c' = c] - 1|$, where c and c' are respectively the "correct" and "guessed" bits, and $Pr[E]$ denotes the probability of a random event E.

Theorem 3. *Suppose an adversary $\mathcal{A}$ is running in polynomial time t_{poly} to derive the group key GK_k established among a road-side unit (RSU_i) and its cluster n member vehicles V_1, V_2, …, V_n in a cluster C_k in the proposed scheme, BBAS-IoV. Let q_h, $|Hash|$, and $Adv_{\mathcal{A}}^{ECDLP}(t_{poly})$ denote "the number of $\mathcal{H}$ queries, the range space of a one-way collision-resistant hash function, and the advantage of breaking the "Elliptic Curve Discrete Logarithm Problem (ECDLP)", respectively. The advantage of $\mathcal{A}$ in deriving the group key GK_k in time t_{poly} can be then estimated by*

$$Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly}) \leq \frac{q_h^2}{|Hash|} + 2Adv_{\mathcal{A}}^{ECDLP}(t_{poly}).$$

Proof. We follow the proof of this theorem in a similar way as that was done in [51,53,69–71].

We define the three associated games, say $Game_l^{\mathcal{A}}$, $(l = 0, 1, 2)$, where the starting and ending games are $Game_0^{\mathcal{A}}$ and $Game_2^{\mathcal{A}}$, respectively. Let $Succ_{\mathcal{A}}^{Game_l}$ be defined as "an event wherein $\mathcal{A}$ can guess the random bit c in the game $Game_l^{\mathcal{A}}$ correctly". Then, the advantage of $\mathcal{A}$ in winning the game $Game_l^{\mathcal{A}}$ is defined as $Adv_{\mathcal{A},Game_l}^{BBAS-IoV} = Pr[Succ_{\mathcal{A}}^{Game_l}]$.

The detailed description of each game is elaborated below.

- $Game_0^{\mathcal{A}}$: The starting game $Game_0^{\mathcal{A}}$ is the game corresponds to the real attack. Therefore, the semantic security of BBAS-IoV defined in Definition 2 gives the following equation:

$$Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly}) = |2.Adv_{\mathcal{A},Game_0}^{BBAS-IoV} - 1|. \quad (1)$$

- $Game_1^{\mathcal{A}}$: This game is modeled as an 'eavesdropping attack", wherein the adversary $\mathcal{A}$ can eavesdrop all the communicated messages: the hello message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ which is sent to all the members and RSU_i by a vehicle V_j in a cluster C_k during the signing phase (see Section 3.3), and response message $Res_{RSU_i,V_j} = \{mb_j, Sig_{RSU_i,V_j}\}$ sent to all member vehicles V_j by RSU_i in the cluster C_k during the group key management phase (see Section 3.5), with the help of the $Execute$ query as shown in Table 3.
  At the end, $\mathcal{A}$ needs to execute both the queries, namely $Reveal$ and $Test$ in order to assure if the derived group key $GK_k = h(RID_{V_1}\|RID_{V_2}\|\cdots\|RID_{V_n}\|rn_{V_1}\|rn_{V_2}\|\cdots\|rn_{V_n}\|Cert_{V_1}\|Cert_{V_2}\| \cdots\|Cert_{V_n}\|RID_{RSU_i}\|rn_{RSU_i}\|Cert_{RSU_i}\|pr_{RSU_i}\|sk)$ is actual key or just a random number. Note that the group key security is based on both the long term secret pr_{RSU_i} of RSU_i as well as the short term secrets rn_{RSU_i} and sk of RSU_i, which cannot be revealed through the eavesdropping of the messages during communication. As a result, it is clear that an "eavesdropping attack" cannot increase $\mathcal{A}$'s winning probability in the game $Game_1^{\mathcal{A}}$. On the other side, both the games $Game_0^{\mathcal{A}}$ and $Game_1^{\mathcal{A}}$ are "indistinguishable". Thus, the following result is obvious:

$$Adv_{\mathcal{A},Game_1}^{BBAS-IoV} = Adv_{\mathcal{A},Game_0}^{BBAS-IoV}. \tag{2}$$

- $Game_2^{\mathcal{A}}$: In this game, the adversary $\mathcal{A}$ needs to do the simulation of hash queries $\mathcal{H}$ and also requires to solve the computational ECDLP defined in Definition 2 in order to derive the group key GK_k. The security of the group key GK_k hinges on both the long term secret pr_{RSU_i} of RSU_i as well as the short term secrets rn_{RSU_i} and sk of RSU_i. To derive pr_{RSU_i} and rn_{RSU_i} from the public $PB_{RSU_i} = pr_{RSU_i}\cdot G$ and $RN_{RSU_i} = rn_{RSU_i}.G$ respectively, the adversary $\mathcal{A}$ needs to solve the ECDLP in polynomial time t_{poly}, whose advantage is $Adv_{\mathcal{A}}^{ECDLP}(t_{poly})$. Furthermore, obtaining sk is equivalent to finding the collision in the hash function. Thus, in the absence of the simulation of hash queries $\mathcal{H}$ and solving ECDLP, both the games $Game_1^{\mathcal{A}}$ and $Game_2^{\mathcal{A}}$ are "indistinguishable". Therefore, if we apply the birthday paradox, we arrive to the following relation:

$$|Adv_{\mathcal{A},Game_1}^{BBAS-IoV} - Adv_{\mathcal{A},Game_2}^{BBAS-IoV}| \le \frac{q_h^2}{2|Hash|} + Adv_{\mathcal{A}}^{ECDLP}(t_{poly}). \tag{3}$$

Since all the relevant queries related to the above games are now executed, and it is only remaining in guessing the random bit c once the $Reveal$ and $Test$ queries are executed. It thus follows that

$$Adv_{\mathcal{A},Game_2}^{BBAS-IoV} = \frac{1}{2}. \tag{4}$$

Solving Eqs. (1), (2) and (4), we arrive to the following result:

$$\begin{aligned}\frac{1}{2}.Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly}) &= |Adv_{\mathcal{A},Game_0}^{BBAS-IoV} - \frac{1}{2}| \\ &= |Adv_{\mathcal{A},Game_1}^{BBAS-IoV} - Adv_{\mathcal{A},Game_2}^{BBAS-IoV}|.\end{aligned} \tag{5}$$

Furthermore, Eqs. (3) and (5) give the following derivation:

$$\frac{1}{2}.Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly}) \le \frac{q_h^2}{2|Hash|} + Adv_{\mathcal{A}}^{ECDLP}(t_{poly}). \tag{6}$$

Now, the final step is to multiply both sides of Eq. (6) by a factor of 2 to obtain the required result after simplification:

$$Adv_{\mathcal{A}}^{BBAS-IoV}(t_{poly}) \le \frac{q_h^2}{|Hash|} + 2Adv_{\mathcal{A}}^{ECDLP}(t_{poly}). \quad \square$$

4.3. Informal security analysis

We analyze the proposed BBAS-IoV informally in the following propositions, and show that BBAS-IoV is resilient against various known attacks.

Proposition 1. *BBAS-IoV is secure against replay attack.*

Proof. The design of BBAS-IoV includes two messages being exchanged between the entities V_j and RSU_j. The first message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$, where $ma_j = \{RID_{V_j}, rn_{V_j}, Cert_{V_j}, Pub_{V_j}, R_{V_j}, Pub_{TA_k}, PB_{V_j}, TS_{V_j}\}$ is formed by a vehicle, and the second message $Res_{RSU_i,V_j} = \{mb_j, Sig_{RSU_i,V_j}\}$, where $mb_j = \{TS_{RSU_i}, Cert_{RSU_i}, Pub_{RSU_i}, PB_{RSU_i}, R_{RSU_i}, RN_{RSU_i}, Pub_{TA_k}, rn_{V_j}, EP_{PB_{V_j}}(GK_k\|rn_{V_j}\| TS_{RSU_i})\}$ is formed by RSU_i during group key management. Both the messages include the current timestamps TS_{V_j} and TS_{RSU_i}. On receiving the messages, the entities check the validity of the message by comparing the received timestamp and the current timestamp. For a valid message and its freshness, the difference of the timestamp values should be a small value, ΔT. Hence, by including timestamp in every message, it is assured that BBAS-IoV is secured against replay attacks because no adversary can be successful in replaying the intercepted messages. $\square$

Proposition 2. *BBAS-IoV is resilient against man-in-the-middle attack.*

Proof. Suppose an adversary $\mathcal{A}$ intercepts a message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$. Further, $\mathcal{A}$ may try to create valid signature on behalf of V_j to send it to RSU_i as an authentication request message. $\mathcal{A}$ may choose random nonces as $a_{V_j}, b_{V_j} \in Z_q^*$ to generate the request message. Then, $\mathcal{A}$ can compute $Sig_{V_{j1}} = a_{V_j}\cdot(RID_{V_j} + Pub_{V_j})$, $Sig_{V_{j2}} = b_{V_j}\cdot(pr_{V_j}\cdot G)$, $h_{2j} = H_2(ma_j \| Sig_{V_{j1}})$ and $h_{3j} = H_3(ma_j\|Sig_{V_{j1}}\|Sig_{V_{j2}})$. $\mathcal{A}$ may then use the public parameter X_k published by TA_k to compute $Prox_{V_j} = H_4(X_k)$. After these computations, $\mathcal{A}$ may finally try to create a signature Sig_{V_j} on ma_j as $Sig_{V_j} = (a_{V_j}+h_{2j})\,(pp_{V_{j0}}+pp_{V_{j1}}) + (b_{V_j}+h_{3j})\, pr_{V_j}\cdot Prox_{V_j}$. But, the attempt to create $Sig_{V_{j2}}$, Sig_{V_j} will be impossible as the long term secrets pr_{V_j} and $(pp_{V_{j0}}, pp_{V_{j1}})$ are not known to the adversary $\mathcal{A}$. Thus, BBAS-IoV is secured against man-in-the-middle attack. $\square$

Proposition 3. *BBAS-IoV is resilient against privileged-insider attack.*

Proof. During the vehicle registration phase (see Section 3.2.1), TA_k loads the credentials $\{RID_{V_j}, R_{V_j}, Cert_{V_j}, (pp_{V_{j0}}, pp_{V_{j1}}), Pub_{V_j}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$ in the tamper-resistant on-board unit (OBU) of a vehicle V_j. So, a privileged insider user of the TA being an adversary will not know secrets stored in V_j. Similarly, as described in RSU registration phase (see Section 3.2.2), TA_k loads the credentials $\{RID_{RSU_i}, Pub_{RSU_i}, R_{RSU_i}, Cert_{RSU_i}, h(\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$ in memory of RSU_i while keeping ID_{RSU_i} as secret. In addition, TA_k also deletes the random secrets r_{V_j} and r_{RSU_i} in order to thwart against other attacks. Thus, privileged-insider attack is resisted in BBAS-IoV. $\square$

Proposition 4. *Vehicle impersonation attack is protected in BBAS-IoV.*

Proof. In order to impersonate a registered vehicle V_j, an adversary $\mathcal{A}$ may intercept the request message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ on $ma_j = \{RID_{V_j}, rn_{V_j}, Cert_{V_j}, Pub_{V_j}, R_{V_j}, Pub_{TA_k}, PB_{V_j}, TS_{V_j}\}$ and try to create another valid request message, say $Hello^a_{Msg_{V_j}} = \{ma^a_j, Sig^a_{V_{j1}}, Sig^a_{V_{j2}}, Sig^a_{V_j}\}$ on behalf of V_j.

Assume that $\mathcal{A}$ picks a random nonce $rn^a_{V_j} \in Z_q^*$ and generates the current timestamp value as $TS^a_{V_j}$ to form $ma^a_j = \{RID_{V_j}, rn^a_{V_j}, Cert_{V_j}, Pub_{V_j}, R_{V_j}, Pub_{TA_k}, PB_{V_j}, TS^a_{V_j}\}$. In addition, assume that $\mathcal{A}$ also selects random secrets $a^a_{V_j}, b^a_{V_j} \in Z_q^*$ to compute $Sig^a_{V_{j1}} = a^a_{V_j}\cdot(RID_{V_j}+Pub_{V_j})$ and $Sig^a_{V_{j2}} = b^a_{V_j}\cdot(pr_{V_j}\cdot G)$. Furthermore, $\mathcal{A}$ may attempt to compute $h^a_{2j} = H_2(ma^a_j \| Sig^a_{V_{j1}})$ and $h^a_{3j} = H_3(ma^a_j\|Sig^a_{V_{j1}}\|Sig^a_{V_{j2}})$, and a signature $Sig^a_{V_j}$ on ma^a_j as $Sig^a_{V_j} = (a^a_{V_j}+h^a_{2j})\,(pp_{V_{j0}}+pp_{V_{j1}}) + (b^a_{V_j} + h^a_{3j})\, pr_{V_j}\cdot Prox_{V_j}$, where $Prox_{V_j} = H_4(X_k)$. But, to create $Hello^a_{Msg_{V_j}}$, the adversary $\mathcal{A}$ needs to know long term secrets pr_{V_j} and $(pp_{V_{j0}}, pp_{V_{j1}})$. Therefore, it is computationally infeasible for the adversary $\mathcal{A}$ to create another valid request message without knowing the long and short term secrets. Hence, BBAS-IoV is secured against vehicle impersonation attack. □

Proposition 5. *RSU impersonation attack is protected in BBAS-IoV.*

Proof. In this attack, suppose an adversary $\mathcal{A}$ intercepts the message $Res_{RSU_i,V_j} = \{mb_j, Sig_{RSU_i,V_j}\}$ during the group key management phase described in Section 3.5, and tries to create a valid message, say $Res^a_{RSU_i,V_j} = \{mb^a_j, Sig^a_{RSU_i,V_j}\}$ on behalf of RSU_i to send it to the member vehicles V_j of RSU_i. Note that $mb_j = \{TS_{RSU_i}, Cert_{RSU_i}, Pub_{RSU_i}, PB_{RSU_i}, R_{RSU_i}, RN_{RSU_i}, Pub_{TA_k}, rn_{V_j}, EP_{PB_{V_j}}(GK_k \| rn_{V_j} \| TS_{RSU_i})\}$ and the signature on mb_j for V_j as $Sig_{RSU_i,V_j} = pr_{RSU_i} + h(mb_j \parallel RN_{RSU_i} \parallel Cert_{V_j} \parallel RID_{V_j} \parallel TS_{V_j} \parallel TS_{RSU_i}) * rn_{RSU_i} \pmod q$. To do so, $\mathcal{A}$ may attempt to generate a random nonce $rn^a_{RSU_i} \in Z^*_q$, a random secret key $sk^a \in Z^*_q$ and current timestamp $TS^a_{RSU_i}$, and compute public $RN^a_{RSU_i} = rn^a_{RSU_i}.G$. However, to compute the signature $Sig^a_{RSU_i,V_j} = pr_{RSU_i} + h(mb_j \parallel RN^a_{RSU_i} \parallel Cert_{V_j} \parallel RID_{V_j} \parallel TS_{V_j} \parallel TS^a_{RSU_i}) * rn^a_{RSU_i} \pmod q$, $\mathcal{A}$ requires the long term private key pr_{RSU_i} of RSU_i. In addition, $\mathcal{A}$ cannot also compute the group key $GK_k = h(RID_{V_1} \| RID_{V_2} \| \cdots \| RID_{V_n} \| rn_{V_1} \| rn_{V_2} \| \cdots \| rn_{V_n} \| Cert_{V_1} \| Cert_{V_2} \| \cdots \| Cert_{V_n} \| RID_{RSU_i} \| rn^a_{RSU_i} \| Cert_{RSU_i} \| pr_{RSU_i} \| sk^a)$ as it involves the long term private key pr_{RSU_i} of RSU_i. As a result, $\mathcal{A}$ will fail to create a legitimate message on behalf of RSU_i, and hence, the proposed BBAS-IoV is resilient against RSU impersonation attack. □

Proposition 6. *BBAS-IoV is resilient against ephemeral secret leakage (ESL) attack.*

Proof. During the group key management phase described in Section 3.5, RSU_i generates a group key $GK_k = h(RID_{V_1} \| RID_{V_2} \| \cdots \| RID_{V_n} \| rn_{V_1} \| rn_{V_2} \| \cdots \| rn_{V_n} \| Cert_{V_1} \| Cert_{V_2} \| \cdots \| Cert_{V_n} \| RID_{RSU_i} \| rn_{RSU_i} \| Cert_{RSU_i} \| pr_{RSU_i} \| sk)$ for all the vehicles V_j of the cluster C_k. The group key GK_k involves use of both long term secret pr_{RSU_i} and short term secrets rn_{RSU_i} and sk. The security of GK_k is then based on the following two cases:

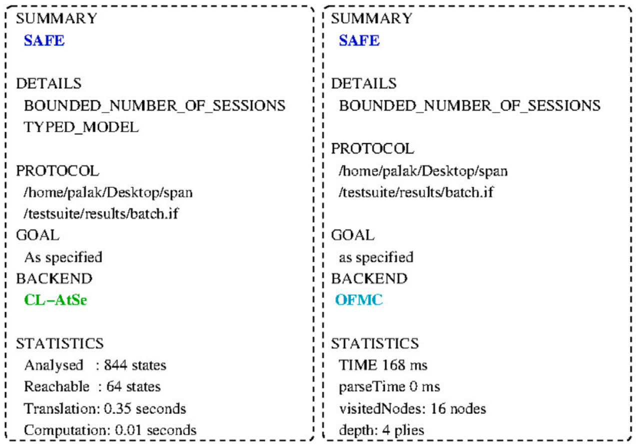
- **Case 1.** If only the long term secret pr_{RSU_i} of RSU_i is compromised, without having the short term secrets rn_{RSU_i} and sk of RSU_i, the group key GK_k cannot be derived by an adversary $\mathcal{A}$.
- **Case 2.** If the adversary $\mathcal{A}$ only knows the short term secrets rn_{RSU_i} and sk of RSU_i, derivation of GK_k cannot be done by $\mathcal{A}$ without having the long term secret pr_{RSU_i} of RSU_i.

Thus, the adversary $\mathcal{A}$ acquires both long term secret pr_{RSU_i} and short term secrets rn_{RSU_i} and sk, then only the group key GK_k is compromised. Therefore, based on the CK-adversary model [28] discussed in the threat model (see Section 1.2), even the adversary does the session hijacking attacks to compromise the short term secrets rn_{RSU_i} and sk of RSU_i, the group key GK_k is still secure because it requires the long term secret pr_{RSU_i} of RSU_i. As a result, ESL attack is protected in the proposed BBAS-IoV under the CK-adversary model. □

5. Formal security verification using AVISPA: simulation study

The formal security verification of the proposed scheme (BBAS-IoV) is done by using widely accepted simulation tool, known as "Automated Validation of Internet Security Protocols and Applications (AVISPA)" [30]. In this section, we describe the security validation of BBAS-IoV against an adversary who may be passive or active. The output of the simulation of BBAS-IoV on the push button tool assures the safety of the scheme against "replay" attack and "man-in-the-middle" attack.

AVISPA tool consists of four back-ends: (a) "On-the-fly Model-Checker (OFMC)", (b) "Constraint Logic based Attack Searcher (CL-AtSe)", (c) "SAT-based Model-Checker (SATMC)", and (d) "Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)". The process of simulation of a protocol initiates by implementing the code in "High-Level Protocol Specification Language (HLPSL)" and then converting it into "Intermediate Format (IF)" with the help of HLPSL2IF translator. Following this, the code in the IF format is provided to any of the back-ends as input, and the output in the "Output Format (OF)" is collected.



```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/palak/Desktop/span
 /testsuite/results/batch.if
GOAL
 As specified
BACKEND
 CL-AtSe

STATISTICS
 Analysed   : 844 states
 Reachable  : 64 states
 Translation: 0.35 seconds
 Computation: 0.01 seconds
```

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
 /home/palak/Desktop/span
 /testsuite/results/batch.if

GOAL
 as specified
BACKEND
 OFMC

STATISTICS
 TIME 168 ms
 parseTime 0 ms
 visitedNodes: 16 nodes
 depth: 4 plies
```

Fig. 5. Simulation results of the proposed BBAS-IoV under CL-AtSe and OFMC backends.

The result in the OF is displayed in various sections which are described below.

- First section is the "SUMMARY" section that displays whether the scheme is "safe, unsafe, or inconclusive".
- The next section is "DETAILS" that has a detailed explanation supporting the result displayed in the "SUMMARY" section, so as to why the protocol is safe or unsafe.
- The next section is "PROTOCOL" section that defines the "HLPSL specification of the target protocol in intermediate form".
- Following the section "PROTOCOL", there is a "GOAL" section that defines "the goal of the analysis which is being performed by AVISPA using HLPSL specification".
- "BACKEND" section displays the name of the back-end amongst the four mentioned backends (OFMC, Cl-AtSe, SATMC and TA4SP) is the last section in the result.

The HLPSL implementation of the proposed BBAS-IoV consists of four basic roles for the TA, V_j, V_l and RSU_i along with two mandatory roles of *session* and *goal & environment*. We have considered registration phase described in Section 3.2 which is done through the secure channel, and the signing phase described in Section 3.3, authentication phase for both V2V authentication phase explained in Section 3.4.1 and batch authentication phase referring to Section 3.4.2 and group key management phase in Section 3.5 that forms a group key and sends to each vehicle in the cluster.

To verify the presence or absence of attack, an intruder (i) in AVISPA takes an active role during communication among various entities. The intruder i is given all the knowledge of public parameters and is even designed to imitate all other roles. The "Dolev-Yao (DY) threat model" [27] is implemented in AVISPA, where the intruder i cannot only eavesdrop the messages but is also given a privilege to modify, delete or insert false messages during communication. The broadly used "Security Protocol ANimator for AVISPA (SPAN)" tool [72] is used to perform formal security verification simulation. The results of simulation of the proposed BBAS-IoV is shown in Fig. 5 which clearly displays the output result via two broadly-used backends: OFMC and CL-AtSe. From the results, it is assured that BBAS-IoV is secured against replay and man-in-the-middle-attacks.

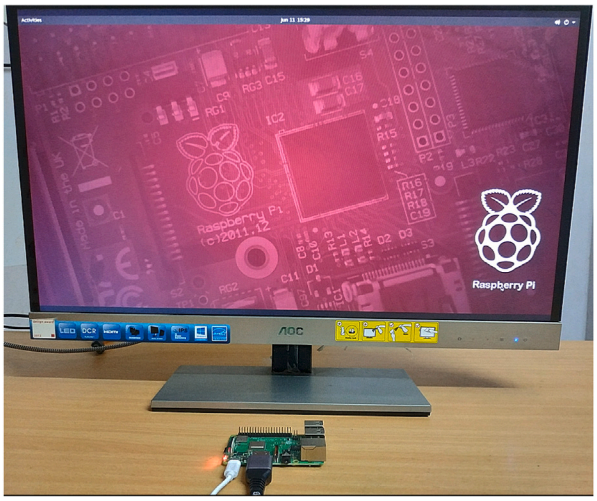


Fig. 6. Experimental setup using Raspberry PI 3.

6. Experimental results using MIRACL

Under this section, we evaluate various cryptographic primitives using the broadly-accepted "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [31] for their execution time. It is worth noticing that MIRACL, a C/C++ based programming software library, has been already recognized by the cryptographers as the "gold standard open source SDK for elliptic curve cryptography (ECC)".

We utilize the symbols T_{exp}, T_{ecm}, T_{eca}, T_{senc}/T_{sdec}, T_h, T_{mul}, T_{add}, T_{bp}, T_{mtp}, T_{ecenc} and T_{ecdec} to denote the time required to execute "modular exponentiation", "elliptic curve point (scalar) multiplication", "elliptic curve point addition", "symmetric key (Advanced Encryption Standard (AES-128)) encryption/decryption", "one-way hash function using SHA-256 hashing algorithm", "modular multiplication over Z_q", "modular addition over Z_q", "bilinear pairing operation", "point-to-map function", "ECC encryption", and "ECC decryption", respectively. The elliptic curve point addition and multiplication are carried out a non-singular elliptic curve of the type: "$y^2 = x^3 + ux + v \pmod q$" such that $4u^3 + 27v^2 \neq 0 \pmod q$. Note that $T_{ecenc} = 2T_{ecm} + T_{eca}$ and $T_{ecdec} = T_{ecm} + T_{eca}$.

We consider the following two platforms for MIRACL.

- **Platform 1.** The platform considered for MIRACL is as follows: "Ubuntu 18.04.4 LTS, with memory: 7.7 GiB, processor: Intel® Core™ i7-8565U CPU @ 1.80 GHz × 8, OS type: 64-bit and disk: 966.1 GB". The experiments are executed for each cryptographic primitive for 100 runs. After that the maximum, minimum and average run-time in milliseconds are recorded for each cryptographic primitive from these 100 runs. The experimental results are then tabulated in Table 4.
- **Platform 2.** The platform considered for MIRACL is as follows: "Model: Raspberry PI 3 B+ Rev 1.3, with CPU: 64-bit, Processor: 1.4 GHz Quad-core, 4 cores, Memory (RAM): 1 GB, and OS: Ubuntu 20.04 LTS, 64-bit". The experimental setup using Raspberry PI 3 is shown in Fig. 6. The experiments are also executed for each cryptographic primitive for 100 runs. We calculate the maximum, minimum and average run-time in milliseconds for each cryptographic primitive from these 100 runs. Finally, the experimental results are provided in Table 5.

Table 4
Execution time (in milliseconds) for a server of cryptographic primitives using MIRACL.

Primitive	Max. time (ms)	Min. time (ms)	Average time (ms)
T_h	0.149	0.024	0.055
T_{exp}	0.248	0.046	0.072
T_{ecm}	2.998	0.284	0.674
T_{eca}	0.002	0.001	0.002
T_{senc}	0.003	0.001	0.001
T_{sdec}	0.002	0.001	0.001
T_{mul}	0.007	0.001	0.002
T_{add}	0.003	0.001	0.001
T_{bp}	7.951	4.495	4.716
T_{mtp}	0.199	0.092	0.114
T_{ecenc}	5.998	0.569	1.350
T_{ecdec}	3.000	0.285	0.676

Table 5
Execution time (in milliseconds) for a Raspberry PI 3 of cryptographic primitives using MIRACL.

Primitive	Max. time (ms)	Min. time (ms)	Average time (ms)
T_h	0.643	0.274	0.309
T_{exp}	0.493	0.178	0.228
T_{ecm}	4.532	2.206	2.288
T_{eca}	0.021	0.015	0.016
T_{senc}	0.038	0.017	0.018
T_{sdec}	0.054	0.009	0.014
T_{mul}	0.016	0.009	0.011
T_{add}	0.013	0.008	0.010
T_{bp}	32.790	27.606	32.084
T_{mtp}	0.406	0.381	0.385
T_{ecenc}	8.885	4.427	4.592
T_{ecdec}	4.453	2.221	2.304

7. Comparative analysis

In this section, we give a detailed comparative study on "communication and computational overheads", "storage overheads" and "security and functionality features" among our proposed scheme (BBAS-IoV) and other existing competing authentication schemes in IoV-related environment, such as the schemes of Tan and Chung [45], Jiang et al. [37], Tzeng et al. [33] and Bayat et al. [18].

7.1. Storage overheads comparison

The basic entities involved in IoV system model shown in Fig. 1 are vehicles, RSUs and TAs. Out of the three, the TAs are given huge computational power and sufficient amount of storage capacity, while RSUs are more resource-rich than the vehicles. Thus, the vehicles are the resource constraint devices. Hence, it is important to analyze the storage overheads of vehicles and RSUs. We analyze the storage overheads of our scheme (BBAS-IoV) and other considered schemes by calculating the amount of storage required by each vehicle and RSU during the initial registration phase. For the proposed BBAS-IoV, in the vehicle registration phase defined in Section 3.2.1, TA loads $\{RID_{V_j}, R_{V_j}, Cert_{V_j}, pr_{V_j}, (pp_{V_{j0}}, pp_{V_{j1}}), Pub_{V_j}, PB_{V_j}\}$ in the OBU of a deployed vehicle V_j. The storage overhead for V_j is then $[160 + 320 + 160 + 160 + (160 + 160) + 160 + 320] = 1600$ bits. To calculate the storage overheads for RSU_i, TA loads $\{RID_{RSU_i}, PB_{RSU_i}, Pub_{RSU_i}, R_{RSU_i}, Cert_{RSU_i}\}$ in RSU_i before deployment as described in Section 3.2.2. The storage required for RSU_i is $[160 + 320 + 160 + 320 + 160] = 1120$ bits.

Similar calculation shows that in Tan and Chung's scheme [45], a vehicle stores 320 bits and RSU stores information of 1120 bits. In Tzeng et al.'s scheme [33], a vehicle and RSU store 416 and 1152 bits of information, respectively. For Bayat et al.'s scheme [18], the storage cost of a vehicle and RSU are 1120 and 1632 bits, respectively. Jiang et al.'s scheme [37] stores 896 bits in a vehicle and 1440 bit in an RSU. The summarized storage overheads comparison is presented in Table 6.

Table 6
Storage overheads comparison.

Scheme	Vehicle's storage (in bits)	RSU's storage (in bits)
Bayat et al. [18]	1120	1632
Jiang et al. [37]	896	1140
Tzeng et al. [33]	416	1152
Tan and Chung [45]	320	1120
BBAS-IoV	1600	1120

Table 7
Communication costs comparison.

Scheme	Single verification	Batch verification	Group key management
Bayat et al. [18]	992	$992n$[b]	–
Jiang et al. [37]	$3748 + 256t$	1440+ $(2308 + 256t)n$[b]	$(900 + 256t)n$[a]
Tzeng et al. [33]	1152	$1152n$[b]	–
Tan and Chung [45]	2496	$1504n$[b] + 992	$3296n$[a] + 448
BBAS-IoV	2912	$2912n$[b]	$2752n$[a]

Note: –: phase is not defined in the scheme; t: degree of a polynomial used in Jiang et al.'s scheme [37].
[a]Group key management for n vehicles.
[b]Batch verification of n messages.

Table 8
Computation costs in the proposed scheme (BBAS-IoV).

Signing phase	
Entity	Computational cost
Individual vehicle (V_j) in a cluster C_k	$3T_{ecm} + 3T_{eca} + 3T_h$ ≈ 7.839 ms
V2V Authentication phase	
Entity	Computational cost
Individual vehicle (V_l) in a cluster C_k	$3T_{bp} + 4T_{ecm} + 4T_{eca} + 3T_h$ ≈ 106.395 ms
Batch authentication phase	
Entity	Computational cost
Road-side Unit (RSU_i) in a cluster C_k	$3T_{bp} + 5nT_{ecm} + (3n+1)T_{eca} + (2n+1)T_h$ $\approx (14.205 + 3.486n)$ ms
Group key management phase	
Entity	Computational cost
Road-side Unit (RSU_i) in a cluster C_k	$T_{ecm} + nT_{ecenc} + (n+1)T_h$ $\approx (0.729 + 1.405n)$ ms
Individual vehicle (V_j) in a cluster C_k	$4T_{ecm} + 2T_{eca} + T_{ecdec}$ ≈ 11.488 ms

7.2. Communication costs comparison

To calculate the communication costs in the proposed BBAS-IoV and other schemes, we assume that an identity, a random number (nonce/secret), the system timestamp, an "elliptic curve point of the form $P = (P_x, P_y)$, where x and y co-ordinates of P are P_x and P_y respectively" and the hash output (digest) using SHA-256 hashing algorithm [73] require 160 bits, 32 bits, $(160+160) = 320$ bits and 256 bits, respectively. Under these assumptions, we calculate the communication costs of the proposed BBAS-IoV for the signing, authentication and group key management phases as follows.

- During the signing phase described in Section 3.3, a vehicle V_j requires to broadcast the hello message $Hello_{Msg_{V_j}} = \{ma_j, Sig_{V_{j1}}, Sig_{V_{j2}}, Sig_{V_j}\}$ to its cluster members including RSU_i, where $ma_j = \{RID_{V_j}, rn_{V_j}, Cert_{V_j}, Pub_{V_j}, R_{V_j}, Pub_{TA_k}, PB_{V_j}, TS_{V_j}\}$. It requires $[(320+160+160+320+320+320+320+32)+(320+320+320)] = 2912$ bits.
- During the group key management phase described in Section 3.5, RSU_i requires to send n response messages $Res_{RSU_i,V_j} = \{mb_j, Sig_{RSU_i,V_j}\}$ to V_j to its each vehicle V_j of the cluster, where $mb_j = \{TS_{RSU_i}, Cert_{RSU_i}, Pub_{RSU_i}, PB_{RSU_i}, R_{RSU_i}, RN_{RSU_i}, Pub_{TA_k}, rn_{V_j}, EP_{PB_{V_j}}(GK_k \| rn_{V_j} \| TS_{RSU_i})\}$. Each message requires the communication cost of $[(32+160+320+320+320+320+320+160+(320+320))+160] = 2752$ bits. The total communication cost due to n response messages Res_{RSU_i,V_j} is then $2752n$ bits.

We have also calculated communication costs of other schemes to evaluate their performance. The scheme proposed by Tan and Chung [45] requires the costs of 2496, $1504n$+992 and $3296n$+448 bits for single verification, batch verification and group key management, respectively. In Tzeng et al.'s scheme [33], a signed message needs 1152 bits that is sent from a vehicle to RSU in single verification. And for batch verification of n vehicles, the cost is $1152n$ bits.

The communication cost of a single message broadcasted by vehicle V_i in Bayat et al.'s scheme [18] requires $(480 + 160 + 320 + 32) = 992$ bits, and it is $992n$ bits for batch verification. In Jiang et al.'s scheme [37], the total communication cost is $3748+256t$, where t is the degree of the polynomial used during the mutual authentication phase. Moreover, the group key decryption only requires a single message of $900 + 256t$ bits to be received by a vehicle. Table 7 summarizes the communication costs of BBAS-IoV and other analyzed schemes for single verification, batch verification and group key management. It is noticed that the communication costs needed for the proposed BBAS-IoV are comparable with other existing competing schemes during the single verification, batch verification and group key management phases.

7.3. Computational costs comparison

For computational costs computation of the proposed BBAS-IoV and other existing competing schemes for various phases, we use the experimental results reported in Section 6. The average execution time (in milliseconds) for a server of various cryptographic primitives using MIRACL reported in Table 4 are used for RSU, whereas the average execution time (in milliseconds) for a Raspberry PI of various cryptographic primitives using MIRACL provided in Table 5 are used for an OBU of a vehicle. It is worth noticing that a t-degree polynomial computation denoted by T_{poly} requires $T_{poly} = t(T_{mul} + T_{add})$, where T_{mul} and T_{add} are the multiplication and addition in Z_q, respectively, and T_{poly} is computed using the Horner's rule [74].

In Table 8, we have shown the computational costs needed by each individual vehicle and an RSU during the signing, V2V authentication and batch authentication phases of the proposed BBAS-IoV. Along with this, we have also analyzed the costs of single verification, batch verification and group key management of other existing schemes. Tan and Chung's scheme needs the computational costs of $14T_h + 16T_{ecm} + 4T_{bp} + 2T_{eca} + 2T_{exp}$ for single verification and $(12n+2)T_h + (12n+4)T_{ecm} + (2n+2)T_{bp} + 2nT_{exp} + 2nT_{eca}$ for batch verification. In addition, in their scheme the group key management requires the computational cost of $(2 + 4n)T_h + 3nT_{poly} + (5n + 1)T_{ecm}$. In Tzeng et al.'s scheme [33], the total computation cost for signing and verifying comes out as $4T_{ecm} + T_{mtp} + 2T_h + 2T_{bp} + T_{eca}$. The costs of generating and verifying a single message in Bayat et al.'s scheme [18] are $1T_{ecm} + 1T_{epa} + 1T_h$ and $3T_{bp} + 1T_{ecm} + 1Tmtp + 1T_h$, respectively. In Jiang et al.'s scheme [37], the total cost of signing and verifying is $5T_h + 7T_{ecm} + 2T_{eca} + 3T_{bp} + T_{enc/dec} + T_{poly}$.

We have divided the computational costs comparisons among the proposed BBAS-IoV and other existing schemes into three groups: (a) single verification which includes the total cost of verification between a vehicle and a RSU/vehicle (shown in Table 9), (b) batch verification includes the total cost of computation performed by n vehicles and their RSU in a group/cluster (shown in Table 10), and (c) group key management that represents the computation by RSU and its n members vehicles in the cluster/group (shown in Table 11). It is also

Table 9
Computation costs comparison for V2V (single) authentication.

Scheme	Vehicle (V_j)	RSU
Bayat et al. [18]	$T_{ecm}+T_{eca}+T_h$ ≈ 2.553 ms	$T_{ecm}+T_{mtp}+3T_{bp}+T_h$ ≈ 14.991 ms
Jiang et al. [37]	$T_h+4T_{ecm}+T_{eca}$ ≈ 9.477 ms	$T_h+2T_{ecm}+T_{eca}+3T_{bp}$ ≈ 15.553 ms
Tzeng et al. [33]	$3T_{ecm}+T_{mtp}+T_h$ ≈ 7.558 ms	$T_{ecm}+T_h+2T_{bp}+T_{eca}$ ≈ 10.163 ms
Tan and Chung [45]	$8T_h+9T_{ecm}+2T_{eca}$ ≈ 23.096 ms	$4T_h+3T_{ecm}+2T_{bp}+2T_{exp}$ ≈ 11.818 ms
BBAS-IoV	$3T_{ecm}+3T_{eca}+3T_h$ ≈ 7.839 ms	$3T_{bp}+4T_{ecm}+4T_{eca}+3T_h$ ≈ 17.017 ms

Table 10
Computation costs comparison for V2RSU (batch) authentication.

Scheme	Group of n vehicles	RSU
Bayat et al. [18]	$(T_{ecm}+T_{eca}+T_h)n$[a] $\approx 2.553n$ ms	$(T_{ecm}+T_{mtp}+3T_{bp}+T_h)n$[a] $\approx 14.991n$ ms
Jiang et al. [37]	$(T_h+4T_{ecm}+T_{eca})n$[a] $\approx 9.477n$ ms	$(T_h+2T_{ecm}+T_{eca}+3T_{bp})n$[a] $\approx 15.553n$ ms
Tzeng et al. [33]	$(3T_{ecm}+T_{mtp}+T_h)n$[a] $\approx 7.558n$ ms	$(T_{ecm}+T_h+2T_{bp}+T_{eca})n$[a] $\approx 10.163n$ ms
Tan and Chung [45]	$(8T_h+9T_{ecm}+2T_{eca})n$[a] $\approx 23.096n$ ms	$(4T_h+3T_{ecm}+2T_{bp}+2T_{exp})n$[a] $\approx 11.818n$ ms
BBAS-IoV	$(3T_{ecm}+3T_{eca}+3T_h)n$[a] $\approx 7.839n$ ms	$(3T_{bp}+5nT_{ecm}+(3n+1)T_{eca}$ $+(2n+1)T_h)$[a] $\approx (14.205+3.486n)$ ms

Note:
[a]Batch verification of n vehicles.

Table 11
Computation costs comparison for group key management.

Scheme	Group of n vehicles	RSU
Bayat et al. [18]	–	–
Jiang et al. [37]	$(2nT_h+nT_{sdec})$[a] $\approx 0.632n$ ms	$(2nT_h+nT_{ecm}+nT_{senc}$ $+nT_{poly})$[a] $\approx (0.785n+0.009nt)$ ms
Tzeng et al. [33]	–	–
Tan and Chung [45]	$(3mT_h+3mT_{ecm})$[a] $\approx 7.791m$ ms	$(2T_h+T_{ecm}+3mT_{poly})$[a] $\approx (0.784+0.009m^2)$ ms
BBAS-IoV	$(4T_{ecm}+2T_{eca}+T_{ecdec})n$[a] $\approx 11.488n$ ms	$(T_{ecm}+nT_{ecenc}+(n+1)T_h)$[a] $\approx (0.729+1.405n)$ ms

Note: –: phase is not defined in the scheme; t: degree of a polynomial used in Jiang et al.'s scheme [37]; m: number of vehicles willing to be in group in Tan and Chung's scheme [45].
[a]Group key management for n vehicles.

noticed that the computational costs needed for the proposed BBAS-IoV are comparable with other existing competing schemes during the single verification, batch verification and group key management phases.

7.4. Security and functionality features comparison

In this section, we compare the proposed BBAS-IoV against other existing competing schemes on functionality and security features. The comparison is tabulated in Table 12 on various functionality and security features (FSF_1–FSF_{12}). It is worth to notice that the proposed BBAS-IoV and Tan and Chung's scheme implement blockchain. Moreover, only the proposed BBAS-IoV provides dynamic vehicle addition phase, AI/ML based security for Big data analytics and formal security verification using AVISPA tool. Considering all the features (FSF_1–FSF_{12}), BBAS-IoV provides better security and more functionality features as compared to those for other existing competing schemes.

Table 12
Functionality & security features comparison.

Feature	Bayat et al. [18]	Jiang et al. [37]	Tzeng et al. [33]	Tan and Chung [45]	BBAS-IoV
FSF_1	✓	✓	✓	✓	✓
FSF_2	✓	✓	✓	✓	✓
FSF_3	✓	✓	✓	×	✓
FSF_4	✓	✓	✓	✓	✓
FSF_5	✓	✓	✓	✓	✓
FSF_6	×	×	×	×	✓
FSF_7	✓	✓	✓	✓	✓
FSF_8	×	✓	×	✓	✓
FSF_9	×	×	×	✓	✓
FSF_{10}	×	×	×	×	✓
FSF_{11}	×	×	×	×	✓
FSF_{12}	×	✓	×	×	✓

Note: FSF_1: "Replay attack"; FSF_2: "man-in-the-middle attack"; FSF_3: "insider attack"; FSF_4: "vehicle impersonation attack"; FSF_5: "RSU impersonation attack"; FSF_6: "dynamic vehicle addition phase"; FSF_7: "batch verification"; FSF_8: "group key management"; FSF_9: "support to blockchain-based solution"; FSF_{10}: "support to AI/ML based security for Big data analytics"; FSF_{11}: "formal security verification using AVISPA tool"; FSF_{12}: "ESL attack under the CK-adversary model for group key construction".
✓: a scheme is secure or assists a feature; ×: a scheme is insecure or does not assist a feature.

8. Blockchain implementation

In this section, we provide the blockchain implementation of our proposed BBAS-IoV using the popular Hyperledger Sawtooth [32].

We have considered the block version ($BVer$), previous block hash ($PBHash$), timestamp (TS_{Block}), Merkle tree root (MTR_{Block}), signer's public key (PB_{RSU_i}), ECDSA block signature (Sig_{Block}) and current/last block hash ($CBHash$) using SHA-256 hashing algorithm are of sizes 32, 256, 32, 256, 320, 320, and 256 bits, respectively, for a full block as shown in Fig. 4. In addition, each transaction in the block requires 2016 bits. Thus, the total block size of a block becomes $1472+2016t_n$ bits.

We now discuss the practical implementation of our proposed scheme to estimate its impact on the performance parameters. We have simulated the implementation using Hyperledger Sawtooth [32], which is an enterprise blockchain platform for distributed applications. The architecture of Hyperledger Sawtooth is outlined in Fig. 7. A typical Sawtooth network comprises of validator nodes, which are responsible for adding the blocks to the blockchain. The elegant feature of Sawtooth in separating the core system from the application level facilitates one to write custom transaction processors subject to the application requirements and register it with the validator nodes. Further, one can develop corresponding clients to generate transactions and send it to the validator nodes via REpresentational State Transfer (REST) interfaces, which is "architectural style for distributed hypermedia systems". Validator nodes verify the transactions by contacting the corresponding registered transaction processors and upon successful verification they will provide the consensus. Hyperledger Sawtooth majorly supports "Practical Byzantine Fault Tolerance (PBFT)" [26], "Proof of Elapsed Time (PoET)" [75] and "Raft" [76] consensus algorithms. We have used the PBFT [26] consensus algorithm to execute the simulation of the proposed BBAS-IoV. In PBFT, one of the validator nodes acts as a leader node which is responsible for creating and adding a block to the blockchain. The leader role gets changed in a round-robin fashion among the validator nodes and a non-leader node has to wait for its turn to create and add a block the blockchain. The PBFT consensus algorithms allows at least 1/3 of the nodes to be faulty, that is, a leader node inserts a block into the blockchain once it gets consensus from 2/3 of the peer nodes in the Sawtooth network.

We have simulated the proposed BBAS-IoV using five P2P servers, each of which acting as a validator node and having the configuration "Ubuntu 18.04, Intel(R) Core(TM) i9-9880H CPU @ 2.30 GHz, 2 GB RAM". The performance of the proposed BBAS-IoV is measured for the following two cases:

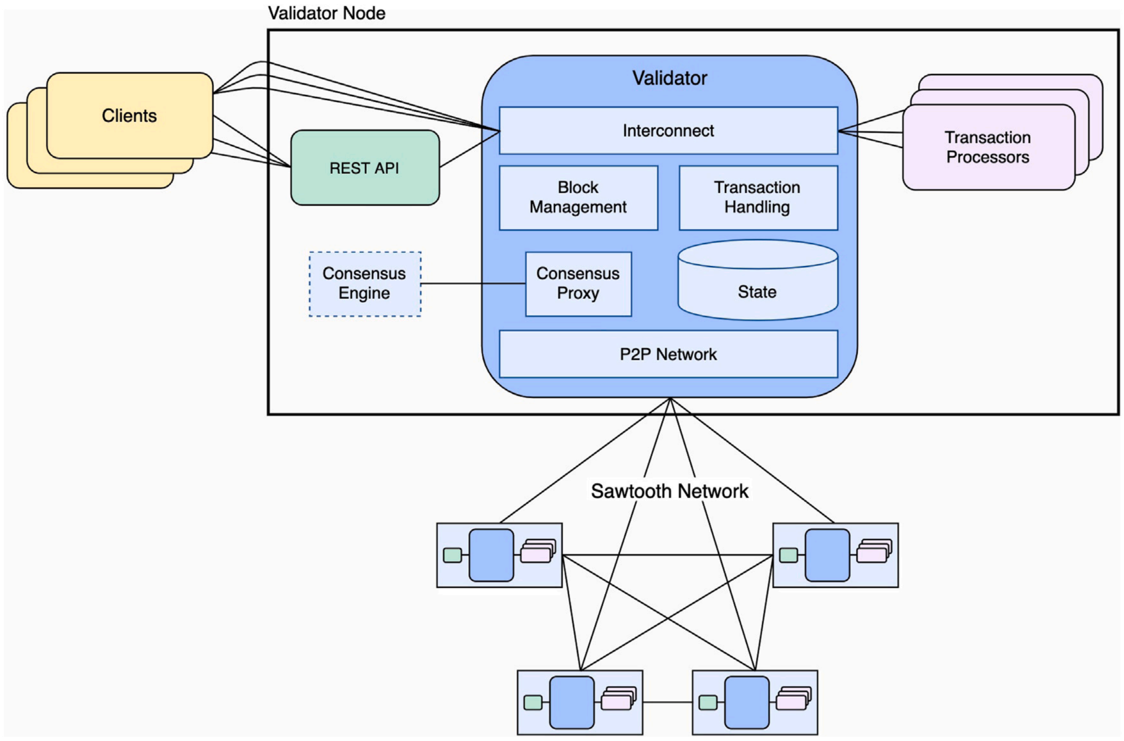


Fig. 7. Hyperledger Sawtooth blockchain platform architecture [32].

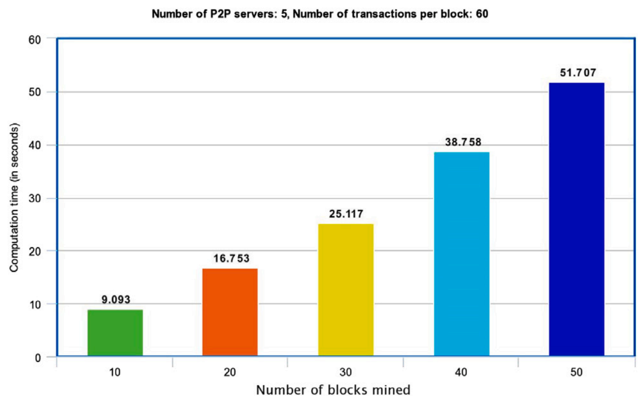


Fig. 8. Blockchain simulation results in Case 1.

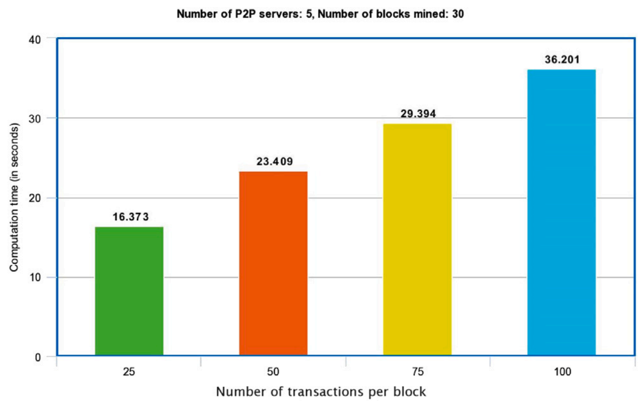


Fig. 9. Blockchain simulation results in Case 2.

- **Case 1:** In this case, we performed the simulation by having a fixed number of transactions per block, which is 60. The simulation results listed in Fig. 8 shows the number of blocks mined into the blockchain versus the computational time (in milliseconds). It is noticed that the computational time linearly increases when the number of blocks mined also increases.
- **Case 2:** In this scenario, we performed the simulation by mining a fixed number of blocks to the blockchain, which is 30, with the varied number of transactions per block. The simulation results are provided in Fig. 9. The simulation results are based on the number of transactions containing in each block versus the computational time (in milliseconds). It is also noticed that the computational time linearly increases when the number of transactions containing in each block increases.

9. Conclusion

To enable smart city environment using intelligent vehicles, we designed a novel blockchain-enabled batch authentication scheme in AI-envisioned IoV-based smart city deployment. The proposed scheme (BBAS-IoV) registers the vehicles and RSUs prior to their deployment. Each vehicle is a part of a dynamic cluster that broadcasts a message to get authenticated from its fellow vehicles and RSU. We proposed both V2V single authentication and batch authentication in which cluster vehicles are authenticated by their RSU simultaneously. Following to these procedures, we also provided a mechanism how RSU can create a group key that to be shared by all the vehicles of the cluster. To make the proposed BBAS-IoV more effective, we incorporated the blockchaining mechanism using the fog servers and cloud computing as well. The blocks of transactions are mined by a voting-based PBFT consensus algorithm. The genuineness and authenticity of the huge amount of data stored in blockchain also gave us an opportunity to use big data analytics though AI/ML algorithms. The mingling of these technologies made the proposed BBAS-IoV efficient and smarter to work effectively in smart cities. Detailed security analysis and comparative study exhibit the proposed BBAS-IoV is superior in terms of security, supports more functionality features and provides comparable storage, communication and computational overheads as compared to those for other existing competing schemes. The blockchain based simulation of the proposed BBAS-IoV shows its effectiveness in terms of computational time.

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.

Acknowledgments

We thank the anonymous reviewers, the associate editor and the Editor-in-Chief for their valuable feedback on the paper. This work was supported by the Mathematical Research Impact Centric Support (MATRICS) project funded by the Science and Engineering Research Board (SERB), India (Reference No. MTR/2019/000699) and also by the Ripple Centre of Excellence Scheme, CoE in Blockchain (Sanction No. IIIT/R&D Office/Internal Projects/001/2019), IIIT Hyderabad, India.

References

[1] R.S. Bali, N. Kumar, Secure clustering for efficient data dissemination in vehicular cyber-physical systems, Future Gener. Comput. Syst. 56 (2016) 476–492.
[2] N. Kumar, N. Chilamkurti, J.H. Park, ALCA: agent learning–based clustering algorithm in vehicular ad hoc networks, Pers. Ubiquitous Comput. 17 (8) (2013) 1683–1692.
[3] N. Kumar, R. Iqbal, S. Misra, J.J. Rodrigues, Bayesian coalition game for contention-aware reliable data forwarding in vehicular mobile cloud, Future Gener. Comput. Syst. 48 (2015) 60–72.
[4] S. Garg, A. Singh, S. Batra, N. Kumar, L.T. Yang, UAV-empowered edge computing environment for cyber-threat detection in smart vehicles, IEEE Network 32 (3) (2018) 42–51.
[5] F. Yang, S. Wang, J. Li, Z. Liu, Q. Sun, An overview of internet of vehicles, China Commun. 11 (10) (2014) 1–15.
[6] W. Zhang, X. Xi, The innovation and development of internet of vehicles, China Commun. 13 (5) (2016) 122–127.
[7] N. Kumar, J.J.P.C. Rodrigues, N. Chilamkurti, Bayesian coalition game as-a-service for content distribution in internet of vehicles, IEEE Internet Things J. 1 (6) (2014) 544–555.
[8] N. Kumar, R. Iqbal, S. Misra, J.J. Rodrigues, An intelligent approach for building a secure decentralized public key infrastructure in VANET, J. Comput. System Sci. 81 (6) (2015) 1042–1058.
[9] S. Ibrahim, M. Hamdy, A comparison on VANET authentication schemes: public key vs. symmetric key, in: Tenth International Conference on Computer Engineering Systems (ICCES'15). Cairo, Egypt, 2015, pp. 341–345.
[10] M. Asghar, L. Doss, R.R.M. Pan, A Scalable and Efficient PKI Based Authentication Protocol for VANETs, in: 28th International Telecommunication Networks and Applications Conference (ITNAC'18). Sydney, NSW, Australia, 2018, pp. 1–3.
[11] J. Li, Y. Ji, K.R. Choo, D. Hogrefe, CL-CPPA: certificate-less conditional privacy-preserving authentication protocol for the internet of vehicles, IEEE Internet Things J. 6 (6) (2019) 10332–10343.
[12] X. Zhu, S. Jiang, L. Wang, H. Li, W. Zhang, Z. Li, Privacy-preserving authentication based on group signature for VANETs, in: IEEE Global Communications Conference (GLOBECOM'13). Atlanta, GA, USA, 2013, pp. 4609–4614.
[13] K. Lim, K.M. Tuladhar, X. Wang, W. Liu, A scalable and secure key distribution scheme for group signature based authentication in VANET, in: IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON'17). New York, USA, 2017, pp. 478–483.
[14] X. Yue, B. Chen, X. Wang, Y. Duan, M. Gao, Y. He, An efficient and secure anonymous authentication scheme for VANETs based on the framework of group signatures, IEEE Access 6 (2018) 62584–62600.
[15] N.B. Gayathri, G. Thumbur, P.V. Reddy, M.Z.U. Rahman, Efficient pairing-free certificateless authentication scheme with batch verification for vehicular Ad-Hoc networks, IEEE Access 6 (2018) 31808–31819.
[16] A.K. Sutrala, P. Bagga, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, P. Lorenz, On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment, IEEE Trans. Veh. Technol. 69 (5) (2020) 5535–5548.
[17] S. Jiang, X. Zhu, L. Wang, A conditional privacy scheme based on anonymized batch authentication in Vehicular Ad Hoc Networks, in: EEE Wireless Communications and Networking Conference (WCNC'13), Shanghai, China, 2013, pp. 2375–2380.
[18] M. Bayat, M. Barmshoory, M. Rahimi, M.R. Aref, A secure authentication scheme for VANETs with batch verification, Wirel. Netw. 21 (5) (2015).
[19] P.V. Kumar, M. Maheshwari, Prevention of Sybil attack and priority batch verification in VANETs, in: International Conference on Information Communication and Embedded Systems (ICICES'14), Chennai, India, 2014, pp. 1–5.
[20] N. Satoshi, Bitcoin: A peer-to-peer electronic cash system, 2018, URL: https://git.dhimmel.com/bitcoin-whitepaper/ Accessed on 2020.
[21] L. Lamport, R. Shostak, M. Pease, The Byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (3) (1982) 382–401.
[22] X. Wang, P. Zeng, N. Patterson, F. Jiang, R. Doss, An improved authentication scheme for internet of vehicles based on blockchain technology, IEEE Access 7 (2019) 45061–45072.
[23] G.S. Veronese, M. Correia, A.N. Bessani, L.C. Lung, P. Verissimo, Efficient byzantine fault-tolerance, IEEE Trans. Comput. 62 (1) (2011) 16–30.
[24] Z. Zheng, S. Xie, H.N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: A survey, Int. J. Web Grid Serv. 14 (4) (2018) 352–375.
[25] R. Gasmi, M. Aliouat, Vehicular Adhoc networks versus Internet of Vehicles - A Comparative View, in: International Conference on Networking and Advanced Systems (ICNAS'19), Annaba, Algeria, 2019, pp. 1–6.
[26] M. Castro, B. Liskov, Practical byzantine fault tolerance and proactive recovery, ACM Trans. Comput. Syst. 20 (4) (2002) 398–461.
[27] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.
[28] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: Advances in Cryptology – EUROCRYPT. Innsbruck (Tyrol), Springer Berlin, Austria, Heidelberg, 2001, pp. 453–474.
[29] M. Abdalla, P.A. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, in: 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), in: Lecture Notes in Computer Science, vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.
[30] Automated validation of internet security protocols and applications, 2019, URL: http://www.avispa-project.org/ Accessed on 2020.
[31] MIRACL cryptographic SDK: Multiprecision integer and rational arithmetic cryptographic library, 2020, URL: https://github.com/miracl/MIRACL Accessed on 2020.
[32] Hyperledger sawtooth architecture guide, intel corporation, 2020, URL: https://sawtooth.hyperledger.org/docs/core/releases/1.1/architecture.html Accessed on 2020.
[33] S. Tzeng, S. Horng, T. Li, X. Wang, P. Huang, M.K. Khan, Enhancing security and privacy for identity-based batch verification scheme in VANETs, IEEE Trans. Veh. Technol. 66 (4) (2017) 3235–3248.
[34] C.C. Lee, Y.M. Lai, Toward a secure batch verification with group testing for VANET, Wirel. Netw. (2013) 19.
[35] H. Zhong, J. Wen, J. Cui, S. Zhang, Efficient conditional privacy-preserving and authentication scheme for secure service provision in VANET, Tsinghua Sci. Technol. 21 (6) (2016) 620–629.
[36] D. He, S. Zeadally, B. Xu, X. Huang, An efficient identity-based conditional privacy-preserving authentication scheme for vehicular Ad Hoc networks, IEEE Trans. Inf. Forensics Secur. 10 (12) (2015) 2681–2691.
[37] S. Jiang, X. Zhu, L. Wang, An efficient anonymous batch authentication scheme based on HMAC for VANETs, IEEE Trans. Intell. Transp. Syst. 17 (8) (2016) 2193–2204.
[38] J. Zhang, H. Zhong, J. Cui, Y. Xu, L. Liu, An extensible and effective anonymous batch authentication scheme for smart vehicular networksluo, IEEE Internet Things J. 7 (4) (2020) 3462–3473.
[39] Bachira Guehguih, Hongwei Lu, Blockchain-based Privacy-Preserving Authentication and Message Dissemination Scheme for VANET. Wuhan, China, 2019, pp. 16–21.
[40] Z. Lu, Q. Wang, G. Qu, H. Zhang, Z. Liu, A blockchain-based privacy-preserving authentication scheme for VANETs, IEEE Trans. Very Large Scale Integr. (VLSI) Systems 27 (12) (2019) 2792–2801.
[41] X. Wang, P. Zeng, N. Patterson, F. Jiang, R. Doss, An improved authentication scheme for internet of vehicles based on blockchain technology, IEEE Access 7 (2019) 45061–45072.
[42] D. Zheng, C. Jing, R. Guo, S. Gao, L. Wang, A traceable blockchain-based access authentication system with privacy preservation in VANETs, IEEE Access 7 (2019) 117716–117726.
[43] Q. Feng, D. He, S. Zeadally, K. Liang, BPAS: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks, IEEE Trans. Ind. Inf. 16 (6) (2020) 4146–4155.
[44] B. Luo, X. Li, J. Weng, J. Guo, J. Ma, Blockchain enabled trust-based location privacy protection scheme in VANET, IEEE Trans. Veh. Technol. 69 (2) (2020) 2034–2048.
[45] H. Tan, I. Chung, Secure authentication and key management with blockchain in VANETs, IEEE Access 8 (2020) 2482–2498.

[46] R. Sharma, S. Chakraborty, Blockapp: Using blockchain for authentication and privacy preservation in IoV, in: IEEE Global Communications Conference (IEEE GLOBECOM) Workshops, United Arab Emirates, Abu Dhabi, 2018, pp. 1–6.
[47] D. He, Y. Zhang, D. Wang, K.K.R. Choo, Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE p1363 standard for public key cryptography, IEEE Trans. Dependable Secure Comput. (2018) http://dx.doi.org/10.1109/TDSC.2018.2857775.
[48] Q. Feng, D. He, Z. Liu, D. Wang, K.K.R. Choo, Multi-party signing protocol for the identity-based signature scheme in IEEE P1363 standard, IET Inf. Secur. (2020) http://dx.doi.org/10.1049/iet-ifs.2019.0559.
[49] M. Wazid, P. Bagga, A.K. Das, S. Shetty, J.J.P.C. Rodrigues, Y. Park, AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment, IEEE Internet Things J. 6 (5) (2019) 8804–8817.
[50] A. Dua, N. Kumar, A.K. Das, W. Susilo, Secure message communication protocol among vehicles in smart city, IEEE Trans. Veh. Technol. 67 (5) (2018) 4359–4373.
[51] C.C. Chang, H.D. Le, A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks, IEEE Trans. Wireless Commun. 15 (1) (2016) 357–366.
[52] N. Kumar, G.S. Aujla, A.K. Das, M. Conti, Eccauth: A secure authentication protocol for demand response management in a smart grid system, IEEE Trans. Ind. Inf. 15 (12) (2019) 6572–6582.
[53] J. Srinivas, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, TCALAS: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment, IEEE Trans. Veh. Technol. 68 (7) (2019) 6903–6916.
[54] S. Challa, A.K. Das, V. Odelu, N. Kumar, S. Kumari, M.K. Khan, et al., An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Comput. Electr. Eng. 69 (2018) 534–554.
[55] M. Wazid, A.K. Das, S. Kumari, X. Li, F. Wu, Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS, Secur. Commun. Netw. 9 (13) (2016) 1983–2001.
[56] M. Wazid, A.K. Das, N. Kumar, V. Odelu, A. Goutham Reddy, Park K., et al., Design of lightweight authentication and key agreement protocol for vehicular Ad Hoc networks, IEEE Access 5 (2017) 14966–14980.
[57] A.K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, X. Huang, Provably secure user authentication and key agreement scheme for wireless sensor networks, Secur. Commun. Netw. 9 (16) (2016a) 3670–3687.
[58] A.K. Das, A.K. Sutrala, S. Kumari, V. Odelu, M. Wazid, X. Li, An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks, Secur. Commun. Netw. 9 (13) (2016) 2070–2092.
[59] M. Wazid, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, Secure three-factor user authentication scheme for renewable-energy-based smart grid environment, IEEE Trans. Ind. Inf. 13 (6) (2017) 3144–3153.
[60] M.S. Kakkasageri, S.S. Manvi, Multiagent driven dynamic clustering of vehicles in VANETs, J. Netw. Comput. Appl. 35 (6) (2012) 1771–1780.
[61] A. Menezes, An introduction to pairing-based cryptography, 2013, URL: https://www.math.uwaterloo.ca/ajmeneze/publications/pairings.pdf Accessed on 2020.
[62] D. Boneh, Pairing-based cryptography: past, present, and future, in: International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'12). Beijing, China, 2012, pp. 1–1.
[63] H. Zhang, J. Wang, Y. Ding, Blockchain-based decentralized and secure keyless signature scheme for smart grid, Energy 180 (2019) 955–967.
[64] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, 2017, CoRR;abs/1712.05526. URL http://arxiv.org/abs/1712.05526.
[65] S. Jangirala, A.K. Das, A.V. Vasilakos, Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment, IEEE Trans. Ind. Inf. 16 (11) (2020) 7081–7093.
[66] Y. Duan, J.S. Edwards, Y.K. Dwivedi, Artificial intelligence for decision making in the era of big data – evolution, challenges and research agenda, Int. J. Inf. Manage. 48 (2019) 63–71.
[67] S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P.K. Singh, W. Hong, Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward, IEEE Access 8 (2020) 474–488.
[68] Y. Liu, F.R. Yu, X. Li, H. Ji, V.C.M. Leung, Blockchain and machine learning for communications and networking systems, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1392–1431.
[69] A.K. Das, M. Wazid, N. Kumar, A.V. Vasilakos, J.J.P.C. Rodrigues, Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment, IEEE Internet Things J. 5 (6) (2018) 4900–4913.
[70] S. Malani, J. Srinivas, A.K. Das, K. Srinathan, M. Jo, Certificate-based anonymous device access control scheme for IoT environment, IEEE Internet Things J. 6 (6) (2019) 9762–9773.
[71] M. Wazid, A.K. Das, V. Odelu, N. Kumar, W. Susilo, Secure remote user authenticated key establishment protocol for smart home environment, IEEE Trans. Dependable Secure Comput. 17 (2) (2020) 391–406.
[72] Span, the security protocol aNimator for AVISPA, 2019, URL: http://www.avispa-project.org/ Accessed on 2020.
[73] W.E. May, Secure Hash Standard, FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, 2015, URL: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. Accessed on 2019.
[74] D.E. Knuth, Seminumerical algorithms, in: The Art of Computer Programming, Volume II, third ed., Addison-Wesley, USA, ISBN: 0201896842, 1998.
[75] B. Curran, What is proof of elapsed time consensus? (PoET) complete beginner's guide, 2018, URL: https://blockonomi.com/proof-of-elapsed-time-consensus/ Accessed on 2020.
[76] D. Ongaro, J. Ousterhout, In search of an understandable consensus algoritm, in: USENIX Annual Technical Conference (ATC'14). Philadelphia, PA, USA, 2014, pp. 305–319.

Palak Bagga is currently a Ph.D. candidate in Computer Science and Engineering at the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India. She received her M.Tech. degree in Computer Science and Engineering from Uttar Pradesh Technical University, India. She was a gold medalist in academics and also awarded by a Certificate of Merit. Her research interests include network security, and security in Internet of Things and Internet of Vehicles. She has published two journal articles in her research areas.

Anil Kumar Sutrala received his Ph.D. degree in computer science and engineering from the International Institute of Information Technology (IIIT), Hyderabad, India, in 2018 and also M.C.A. from the University of Hyderabad, India, in 2006. He is currently working as a principal software engineer with the CA Technologies — A Broadcom Company, Hyderabad 500 032, India. His research interests include cryptography and network security. He has published several journal articles in his research areas.

Ashok Kumar Das received a Ph.D. degree in computer science and engineering, an M.Tech. degree in computer science and data processing, and an M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, network security, blockchain, security in Internet of Things (IoT), Internet of Vehicles (IoV), Internet of Drones (IoD), smart grids, smart city, cloud/fog computing and industrial wireless sensor networks, and intrusion detection. He has authored over 235 papers in international journals and conferences in the above areas, including over 200 reputed journal papers. Some of his research findings are published in top cited journals, such as the IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Smart Grid, IEEE Internet of Things Journal, IEEE Transactions on Industrial Informatics, IEEE Transactions on Vehicular Technology, IEEE Transactions on Consumer Electronics, IEEE Journal of Biomedical and Health Informatics (formerly IEEE Transactions on Information Technology in Biomedicine), IEEE Consumer Electronics Magazine, IEEE Access, IEEE Communications Magazine, IEEE Systems Journal, Future Generation Computer Systems, Computers & Electrical Engineering, Computer Methods and Programs in Biomedicine, Computer Standards & Interfaces, Computer Networks, Expert Systems with Applications, and Journal of Network and Computer Applications. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of IEEE Systems Journal, Computer Communications (Elsevier), IET Communications, KSII Transactions on Internet and Information Systems, and International Journal of Internet Technology and Secured Transactions (Inderscience), is a Guest Editor for Computers & Electrical Engineering (Elsevier) for the special issue on Big data and IoT in e-healthcare, ICT Express (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT and Wireless Communications and Mobile Computing (Wiley/Hindawi) for the special issue on Security and Privacy for Smart Mobile Devices: Attacks, Challenges, and New Designs, and has served as a Program Committee Member in many international conferences. He also severed as one of the Technical Program Committee Chairs of the International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, and 2nd International Congress on Blockchain and Applications (BLOCKCHAIN 2020), L'Aquila, Italy, October 2020.

Pandi Vijayakumar received the B.E. degree in computer science and engineering from Madurai Kamaraj University, Madurai, India, in 2002, the M.E. degree in computer science and engineering from the Karunya Institute of Technology, Coimbatore, India, in 2005, and the Ph.D. degree in computer science and engineering from Anna University, Chennai, India, in 2013. He is the former Dean and presently working as Assistant Professor in the Department of Computer Science and Engineering at University College of Engineering Tindivanam, which is a constituent college of Anna University, Chennai, India. His current research interests include key management in network security, VANET security, and multicasting in computer networks. He has produced four Ph.D. candidates successfully. He has published various quality papers in reputed journals like IEEE Transactions/Journals, ACM transaction, Elsevier, Springer, IET, Taylor & Francis, Wiley, etc.