Connection Science

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ccos20

A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for Internet of vehicles

Hua Yi Lin & Meng-Yen Hsieh

To cite this article: Hua Yi Lin & Meng-Yen Hsieh (2022) A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for Internet of vehicles, Connection Science, 34:1, 1089-1118, DOI: 10.1080/09540091.2022.2045254

To link to this article: https://doi.org/10.1080/09540091.2022.2045254

Published online: 09 Mar 2022.

CONNECTION SCIENCE
2022, VOL. 34, NO. 1, 1089–1118
https://doi.org/10.1080/09540091.2022.2045254

A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for Internet of vehicles

Hua Yi Lin[a] and Meng-Yen Hsieh[b]*

[a]Department of Information Management, China University of Technology, Taipei, Taiwan; [b]Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan

ABSTRACT

Nowadays, broadband wireless networks and 5G have improvements in the quality and dependability of network services for high-speed mobile vehicles. Within an open communication environment on the Internet of vehicles named IoV, personal details will be revealed on the Wi-Fi network. Considering information security and privacy, this research concentrates on how to protect secure information transmission in real time through IoV, and also wants to avoid illegal vehicles, passers-by or drivers from joining connection and planting or altering significant data. This research provides a multilevel security infrastructure with M-tree based elliptic curve digital signature algorithm (ECDSA) for securing data transmission in both IoV and cloud environments. While transmitting, this study provides several flexible and scalable schemes to manage security issues, which dynamically adjust the framework depending on the rapidly changing IoV topology and speed up the time to synchronise the reconstruction of the system key, and decrease the quantity of phases to resynchronise the system key. Moreover, most research of key management without secure data transmission, we integrate the M-tree key management with secure data transmission to achieve the secure IoV. Simultaneously, the provided key management for the multi-level security is quite appropriate to adaptable and expandable IoV.

ARTICLE HISTORY
Received 9 January 2022
Accepted 17 February 2022

KEYWORDS
5G; Internet of vehicle; IoV; M-tree; ECDSA

1. Introduction

As the Internet of vehicles becomes popular and intelligent, vehicles can establish a connection with more open devices, and thereby the security issues are created. Nowadays, IoV attracts great attention to researchers. Because the IoV may communicate with vehicles, roadside equipment and even cloud services mutually. For the foreseeable future, IoV is going to carry out a big data application and analysis on cloud computing. Simultaneously, the V2X technologies will be employed to strengthen driving safety. Those are vehicles to infrastructure named V2I, vehicles to roadside equipment named V2R, vehicles to vehicles

CONTACT Hua Yi Lin calvan.linmsa@gmail.com Department of Information Management, China University of Technology, Taipei, 116 Taiwan
*Present Address: Department of Information Management, China University of Technology, Taipei, 116 Taiwan

named V2V, vehicles to pedestrians named V2P, vehicles to transportation named V2T, and vehicles to groups named V2G and vehicles to networks named V2N. In particular, V2V is gradually becoming more and more popular. Currently, IEEE would like to propose the security and communication framework named IEEE1609/802.11P.

Nowadays, many countries are continually investing in autopilot and driverless vehicle research. The 5G is going to encourage us to evolve into the leading driving technology for autopilot and driverless vehicles. IoV is a mass-connected network, which integrates the necessary information like vehicle speed, route and location. The vehicle can supplement the collection of its own information about the status and environmental information through sensors, GPS, radars, cameras and image processing devices. Thanks to cloud computing technology, diverse IoV information transmissions are combined with each computer in a vehicle to measure, analyze and process a vast quantity of data. Afterwards, the vehicles find the best path or the possible path, and report the status of the road condition to the control centre immediately. The monitoring system may programme the traffic signal cycle, depending on the real-time traffic conditions to reduce the traffic bottleneck.

The premature phase of vehicle to everything (V2X) counts on typical wireless transmission and adopts a similar technique with Wi-Fi networks, yet its dependability is weak and undefended to intrusion and interference offences. So far, with the progressive popularity of 5G being developed, lots of countries are working towards the development of 5G technology. Thus, the original IoV concept is gradually being changed to the IoV application. In the foreseeable future, 5G is going to definitely offer reliable and efficient V2X services. Meanwhile, except for communicating with one another, the vehicle may also deliver messages via roadside devices and adopt IoV to identify unusual car accidents. In the event of an accident, short-range communication may be initiated. It stores information about accidents or scene information and delivers the videos or accident details to the platform of cloud services. Afterwards, V2X may assist in the completion of accident summary for the vehicle and in restoring the location of the traffic accident at which the accident occurred, such as the precise GPS particulars of the vehicle, the position of the roadside devices, and travel routes and journey information of the vehicle with pictures to examine the behaviour of drivers before car accidents.

Nevertheless, since the IoV open communication surroundings, within an open Internet of vehicle communication environment, personal details are going to be revealed on the Wi-Fi network. Taking into account information security, privacy, the security of information as well as the protection of personal data, the purpose of this study is to safeguard transmitted information in real time and to avoid unauthorised vehicles, drivers or passers-by from being involved in the communication and implementation or modification of critical transmitted information. This study develops a virtual machine on the cloud service platform that comprises services such as key management, identity verification, security certification, the public key infrastructure PKI and the certification authority (CA) server. At the border entrance, the cloud service category manager (CSCM) takes charge of transmitting the type of service required from the vehicles to the end of the cloud service platform, and thoroughly analyzing and dealing with a variety of messages coming from vehicles and RSU.

Moreover, this research must examine how to guarantee security between the vehicle with vehicle (V2V), the vehicle with roadside equipment (V2R), the vehicle with group

(V2G) and the vehicle messages with the cloud service platform, as the vehicle moves and communicates with the RSU using the sensor of the onboard OBU. In short, IoV requires a secure data communication framework to guarantee the information security delivered among OBU, RSU and the cloud service platform. This research also takes into consideration that the IoV infrastructure and topology are changing so rapidly, when the vehicle is involved or deviating from the secure transmission of the IoV. Besides, the connected route can change, and vehicles can be connected to various base stations. As a result, this study needs to offer a secure key management mechanism that can address dynamic change in topology. Moreover, the proposed scheme involves multiple paths and can be made quickly and effectively to process the secure transmission of data in IoV, when there is a proactive change in the IoV topology. As a result, based on dynamic M-tree structures, this research provides a multi-level security key management protocol for IoV with secure data transmission to address IoV weaknesses in key management and security problems. This will improve the information security of IoV.

The rest of this paper is structured as follows. Section 2 introduces the related work. Section 3 presents the process of ECDSA digital signature of IoV. Section 4 introduces the framework of multi-level security. Section 5 describes the security key management agreement based on M-tree and secure data transmission modes. Section 6 is the analysis of security and performance. Eventually, conclusions and further work are derived from Section 7.

2. Related work

Over the past few years, lightweight security mechanisms, including PUF-based authentication (Zhang et al., 2021) and elliptic curve Diffie–Hellman protocol, have been adopted to various application domains, such as medical information systems (Xiao et al., 2021), wireless sensor networks (Hua, 2021; Lin et al., 2016) and smart environments (Xia et al., 2021).

Due to the unique characteristics of vehicular networks, restricted computing resources and unstable communication, security issues in IoV applications tend to be performed by lightweight cryptographic schemes. Not only authentication, but also key operations possibly related to key agreement (Xu et al., 2021), key management (Hua et al., 2021; Zhang et al., 2021) and key distribution (Vijayakumar et al., 2017) protocols are concerned in these issues. Further, wireless high-frequency transmission by 5G (Hu et al., 2021) or 6G technology (Pandi et al., 2022) will support higher capacity and much lower latency for data transmission of IoV and VoX applications.

IoV is considered a mission-critical infrastructure. Vehicles aggregate real-time data with other vehicles, RSU and cloud servers, and transmit aggregated data to OBU. IoV data can be analyzed either locally or on a cloud server, and actions are taken depending on the request type. Because IoV has wide-scale interconnecting networks with a large number of vehicles. Obviously, there exists a great risk of privacy and security issues. Through the encryption mechanism and digital signature cryptographic protocols can address those concerns. Additionally, the simultaneous need for the encryption mechanism and the digital signature, these mechanisms are amalgamated named signcryption. In order to eliminate the issue of key escrow related to the signcryption, a certificateless approach is generally considered.

In 2007, Xiaodong et al. exploited a group signature to inspect multiple signatures (Xiaodong et al., 2007; Yong et al., 2013), and thus improved the efficiency of authentication. The proposed scheme is based on bilinear paring. However, this mechanism exists a serious defect that the paring operations at least need two members to complete and cannot be extended, and therefore this proposed mechanism is not scalable for IoV. Additionally, this mechanism exploits lots of exponential operations that consume computing power and increase the complexity of operation.

In 2007, Raya et al. proposed the LAB scheme (Maxim & Jean, 2007). Initially, this mechanism places every OBU in abundance of private keys and their correlative anonymous certifications within a pool with 43,800 certificates. Then, a vehicle arbitrarily chooses one of its anonymous certifications and adopts its corresponding private key to sign the transmitted data. The receiver adopts the public key of the sender with the anonymous certificate to verify the source data. Those anonymous certificates are used to replace the identity of the vehicles. Every certificate only exists for a short period of time to match the privacy requirement of the vehicle. Although, the proposed LAB scheme achieves the purpose of the supposed privacy requirement, this mechanism needs to maintain quite a huge number of certificates and the identities of authority for all vehicles, and therefore becomes inefficient and not scalable for a large number of IoVs.

In 2013, Hu et al. provided a privacy-preserving scheme for vehicle Ad Hoc networks using group signature protocol (2013). In this study, each vehicle keeps secret keys and public keys to avoid disclosing its identity. Nevertheless, as the increasing quantity of vehicles in the revocation list increases, the consuming time of completing group signature also increases linearly. In other words, this proposed scheme consumes lots of extra time.

In 2018, Nisha et al. provided a scheme to ensure the security of vehicles using the blockchain mechanism (Ali et al., 2019; Kulathunge & Dayarathna, 2019; Nisha et al., 2018). At the same time, Ali et al. exploited a certificateless signature to prove the integrity of transmitted data in IoV, and used a blockchain to store the identity of authorised vehicles and also adopted other blockchain to store unauthorised or revoked vehicles.

In 2015, Xiaodong et al. proposed a security protocol named GSIS for VANETs, which combines group signature with identity-based signature to achieve privacy-preserving (Hong et al., 2016; Xiaodong & Rongxing, 2015). This study exploits group signature to protect the communication between multiple OBUs, where the transmitted data are anonymously signed by the sender, meanwhile, the authorities can recover the identities of senders. Additionally, RSUs adopt identity-based cryptography to sign the transmitted data from RSUs, and make sure the authenticity of RSUs. Since the identity is implied in RSUs, and therefore can reduce the great overhead of signatures.

2020 Joon et al. proposed a secure key and authentication agreement to conquer the security issue in IoV (2020). This study only exploits XOR operations and hash functions to achieve this agreement. In this study, vehicles and RSUs register at the TA (Trusted Authentication) server and perform the secure key protocol instead of a tamper-proof device TPD. Afterwards, vehicles use the session key to encode and decode the transmitted data to RSUs. After receiving the transmitted data, RSUs inspect the received data and forward them to the cloud platform.

In 2021, Insaf et al. proposed an anonymous certificate encryption scheme for IoV on the basis of the Hyper elliptic Curve (HEC) (2021). This given system ensures an official analysis

of security as part of the Random Oracle Model (ROM) for confidence, affordability, and anonymous receiver. Additionally, the proposed certificateless cryptography mechanism avoids the issue of key escrow problem and guarantees the anonymity of the receiver in open wireless channels. However, this scheme without the capability to allocate partial private key on an open channel, and thus needs a secure channel to share partial private key with vehicles in IoV environments.

In 2021, Mu et al. proposed an efficient secure attribute-based encryption system for IoV using association rules (2021). This study adopts the features of frequency between vehicle nodes through the max-miner association rules algorithm, and aims to construct frequent item sets and carry out secure communication in vehicle nodes belonging to the same classification set. In this scheme, only ECUs that meet the ciphertext requirements can decipher the data, ensuring data confidentiality while reducing the risk of attacks on the in-vehicle network. However, the access frequency features cannot report the real situation, if an outsider vehicle joins this network and would like to communicate with the group will be turned down.

The IoV system is susceptible to various security and privacy hazards. Although the above methods provide signcryption, authentication or encryption of data for IoVs networks, all of the aforementioned methodologies are based on complex cryptographic mechanisms and bilinear pairing, and all of them have high calculation and communication costs and are incompatible with IoV environments. In other words, they also suffer from the problems of complicated operations, scalability and consume lots of computing time. Therefore, a light security system is needed to ensure protection from a variety of known and unknown threats. This study would like to propose a high efficiency, flexible and scalable mechanism to deal with the security transmissions in IoV.

3. The process of ECDSA digital signature of IoV

As the IoV computer has limited computing resources, this research surveys numerous conventional signature algorithms. We find that the elliptic curve signature (ECDSA) possesses the benefits of rapid rate, strong and small signature. Besides, the ECDSA usage has become increasingly widespread. The functionality of the ECDSA resembles many algorithms of signature (Yasin & Erkan, 2021). It exploits a private key to sign the message and using a public key to inspect the signature. Moreover, the ECDSA operation mode is similar to the other cryptographic schemes, but the key of ECDSA is retained in CNG (cryptography API: next generation). Through CNG, this study is able to securely maintain key pairs, public keys and refer to them utilising simple string names.

Moreover, compared ECDSA to RSA (Ron Rivest, Shamir, Adleman), this study observes that ECDSA utilises a smaller key length to reach a similar level of RSA security, even though ECDSA requires complex implementing methods. The theoretical foundation for ECDSA is to resolve the problem of discrete logarithm of the elliptic curve named ECDLP (Shoma et al., 2015). Nevertheless, the benefits of RSA are convenient to implement and also can be used for encrypting and deciphering simultaneously. Moreover, RSA is slower and the length of the signature is significantly longer, and the theoretical foundation for RSA is to handle the factoring of very large integers. A comparison between ECDSA and RSA can be found in Table 1 (Dhanashree et al., 2018; Nils et al., 2004).

Taking into account the current phase, the sensing device took along by the vehicle's OBU generally has no strong computer power. This research primarily concentrates on the IoV surroundings and presents a cryptographic signature protocol using an elliptic curve and a mechanism for transmitting information security. Therefore, we adopt the ECDH (elliptic curve Diffie-Hellman) key exchange protocol to carry out our key management mechanism and secure data transmission among V2V, V2R and V2G. As ECDH replaces exponential with multiplication and addition (Ram & Manoj, 2013), it uses less necessary memory to the OBU sensors, minimises intricate encrypting and deciphering calculations for the CPU, and ensures a rapid and effective transmission of information security. In addition, the study suggests that vehicles must also communicate with at the end of cloud service platforms, and as a result, this study utilises ECDSA to guarantee the secure data transmission of IoV to cloud platforms, and thus obtain the fully secure IoV architecture.

Furthermore, in practical environments, this study should combine the elliptic curve cryptographic system with the OBU sensors and RSU roadside equipment. Taking into account the need for deployment, we first choose a $GF(p)$ finite field, and estimate the quantity of OBUs and the roadside device. Therefore, this study selects a proper elliptic curve $y^2 = x^3 + ax + b \bmod p$, where $a = 1$, $b = 1$, $p = 307$ in F_{307}, for instance, the following elliptic curve $y^2 = x^3 + x + 1 \bmod 307$, where p is a prime and G is the elliptic curve Ep (a, b) generator point. The selected elliptic curve conforms to the next qualifications (Daisy et al., 2015): (1) n is a prime and the G order; (2) ${y_G}^2 = {x_G}^3 + ax_G + b \bmod p$; (3) $nG = 0$ and $n < p$. Thanks to this elliptical curve, this survey generates a total of 307 points, as illustrated by Figure 1. Moreover, taking into consideration the quantity of vehicle OBUs and roadside devices, this study allows the elliptic curve and coefficients to be flexibly adjusted and expanded to generate the deployed points required to fit the needs of vehicles and roadside devices.

ECDSA is comprised of ECC (Amit et al., 2016; Zhe et al., 2017) and DSA. The whole process of signing, as shown in Table 2 and Figure 2, is comparable to DSA. However, the obvious distinction during the signing is the algorithm, which utilises ECC to perform the signature. Afterwards, this study obtains the final result of the signature being divided into (r, s). The process for signing is the following.

4. The framework of a multi-level security

As IoV needs to integrate vehicles, roadside devices and cloud service servers. As a result, each security level is considered similar to the hierarchical authorisation of companies, and

Table 1. Comparing ECDSA to RSA.

Cryptosystem	ECDSA Algorithm (Elliptic Curve Digital Signature Algorithm)			RSA Algorithm (Rivest, Shamir and Adleman)		
Secure degree	2^{128}	2^{192}	2^{256}	2^{128}	2^{192}	2^{256}
Signature key Length	64	96	132	384	960	1920
Merits	Rapid speed, short signature length.			Use concise, encrypted and deciphered at the same time.		
Defects	Complicated implementation technology.			With lower speed and longer signature length.		
Security theoretical basis	ECDLP: Elliptic Curve Discrete Logarithm Problem.			The factorisation of extremely large integers.		

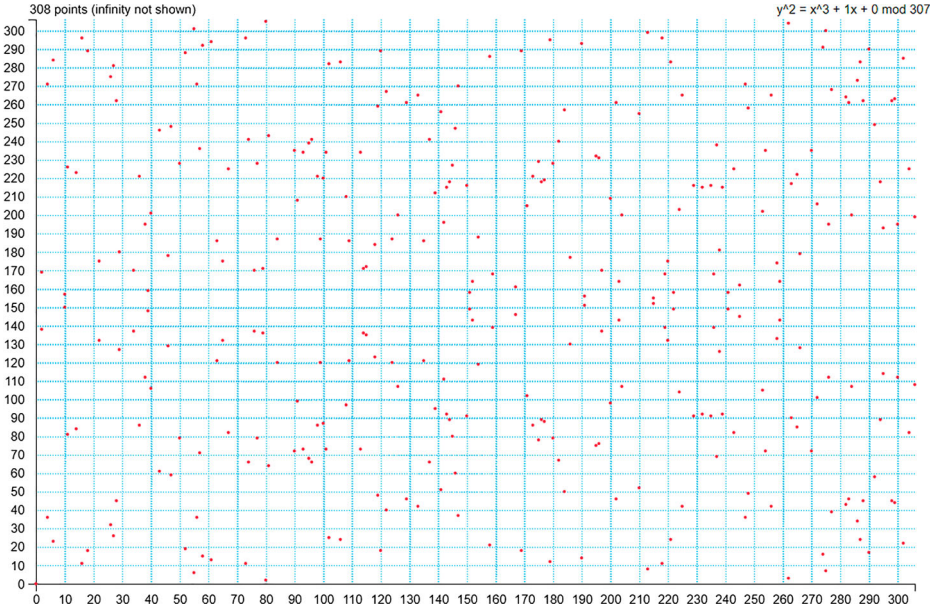

Figure 1. The elliptic curve point generated by $y^2 = 1x^3 + 1x + 1 \bmod 307$.

each security level has individual duties and is responsible for its own security task. The main purpose of this study is to propose a mechanism of the key management of the multi-level security for IoV via an M-tree framework. Because each vehicle is deployed at the lowest level of the key management scheme, then this study maps the leaf node to a vehicle on layer 4 on the security level. Additionally, RSUs are in charge of gathering the data delivered from the OBUs of the vehicle. This study places the RSUs in layer 3 of the security level, and the LCA (local certificate authority CA) in layer 2. The GCA (global CA) and cloud server correspond to the top level on layer 1 of the security level.

This research also takes into account detection devices of OBU that the vehicles carried at this stage generally have no sufficient computing resources. Here, many vehicles are split into multiple groups, and a group is considered a key management unit. Consequently, we can effectively lower the time complexity of updating and synchronising keys of the entire system, and exploit the ECDH (Elliptic Curve Diffie-Hellman) key interchange protocol to exchange keys, manage and en/decipher data to guarantee the secure data transmission between V2V, V2R and V2G. Besides, because the topology of the IoV is changing quite fast, we must also take into account the management of updating and synchronising keys for IoV.

Consequently, the main purpose of this study is to provide a key management agreement based on a multi-level security framework to combine the elliptic curve cryptosystem with IoV and cloud service platforms (Sathishkumar & Rajakumar, 2017). As V2V or V2R equipment executes secure data transmission, this study employs the ECDH session key to perform the secure data transmission for point-to-point. While the communication is between a vehicle and the group vehicle V2G, this research employs the group key management mechanism to compute a common group key for protecting the security of transmitted data. Additionally, this study deploys a VM virtual machine server at the end

Table 2. Sender and receiver perform ECDSA signature process.

Vehicles/OBU/LCA	Signature	Vehicles/OBU/LCA
1. Select an elliptic curve $E_{GF(p)}$ (d, e), and a generator G. Elliptic curve order $= n = \mid E_{GF(p)}$ $(d, e)\mid + 1$, and n is the number of points on the curve that include infinitely far points. 2. Select a point $G = (x_G, y_G)$ and k private key, where $k \in [1, n-1]$ and the G order is n, and then the usage of the generator G calculates $P_K = k$ x $G = k$ x (x_G, y_G) public key and it is denoted as (d, e, p, n, G, P_k).	3. Select a random integer $i \in [1, n-1]$, and calculate the point $R = (x_R, y_R) = i$ x $G = i$ x (x_G, y_G). 4.Use the coordinate values (x, y) and the original message M from point R as parameters, and then take SHA1 as the hash function for computing $e =$ HASH(M). 5. $r = x_R$ mod n; 6. $s = (k$ x $r + e)$ x i^{-1} (mod n). 7. The signature result is (M, r, s). When r or s is 0, go back to duplicate step 3 to regenerate a random integer r until finished from step1 to step7.	8. The recipient accepts the M message along with the value of the signature (M, r, s). 9. Calculate $e =$ HASH(M), $w = s^{-1}$ mod n, $u_1 = e$ x w mode n, $u_2 = r$ x w mode n. $C = (x_c, y_c) = u_1$ x $G + u_2$ x P_k. 10. Verification of the equation $r = x_c$. 11. In case the mentioned equation $r = x_c$ is accurate, and then takes the signature, else turns down the signature.

of the cloud to serve as the root base station GCA, which is an administrator of the entire system and is responsible for the key management agreement.

Moreover, this study deploys a secure LCA in the V2G communication environment to server as the local group leader. LCA is responsible for the key exchange protocol of ECDH, and updating and synchronising the group key. In other words, LCA is the leader of local RSUs, and takes charge of the local group key management and synchronisation.

Figure 3 depicts the authority framework. Initially, a deployed trusted and secure GCA server is located in this entire system. GCA takes in charge of the issuance of key pairs and the digital certificate. If a vehicle wants to take part in this system, the aforementioned information needs to be applied to GCA. Additionally, OBU, RSU or related devices have to be embedded in the ECC cryptosystem.

Afterwards, this study proposes a M-tree framework, as illustrated by Figures 4 and 5, to map IoV equipment to the related security level. The lowest level represents the vehicle. Because the vehicle serves as the role of sender and receiver during the secure data

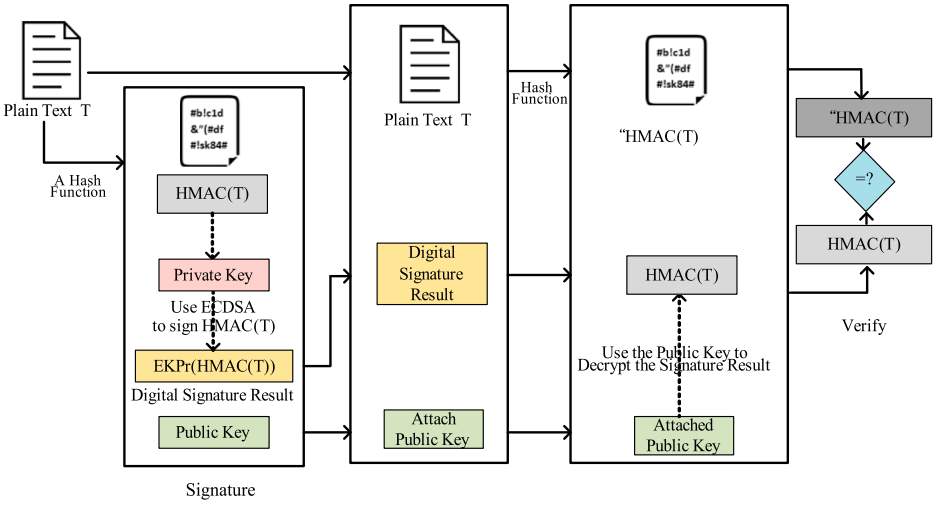

Figure 2. The whole ECDSA signature process.

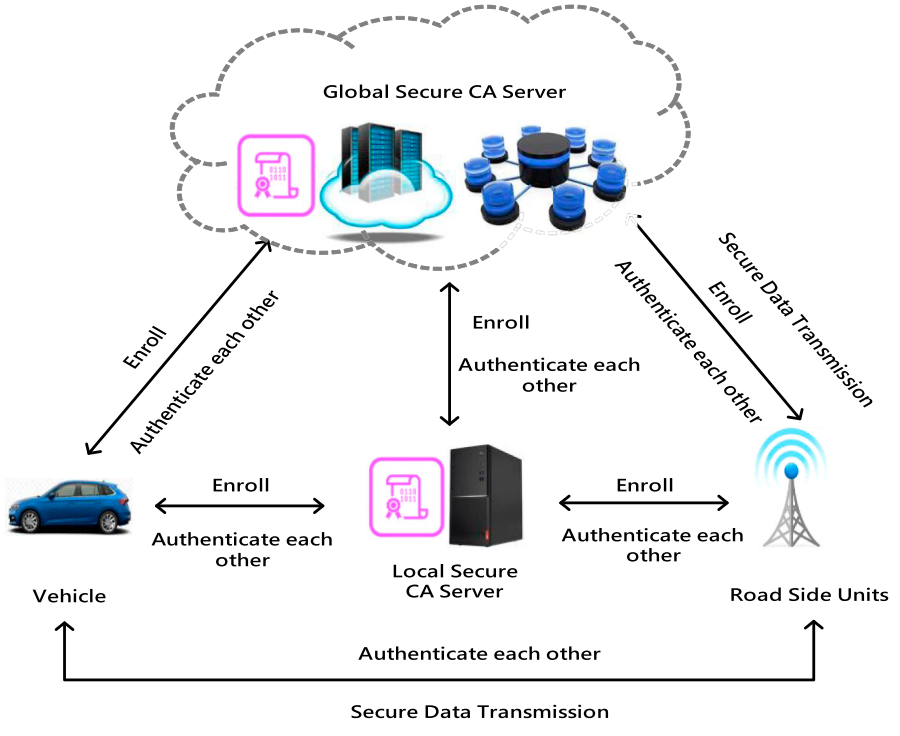

Figure 3. The multi-level security system framework.

transmission. Additionally, the roadside device RSU serves as a mediator role, and RSUs are located on level 3 of M-tree, since the vehicle transmits data through the RSU to other vehicles or to the cloud service control manager CSCM and GCA, which gathers all delivered data from vehicles, and this study arranges the CSCM and GCA on level 1. Eventually, we

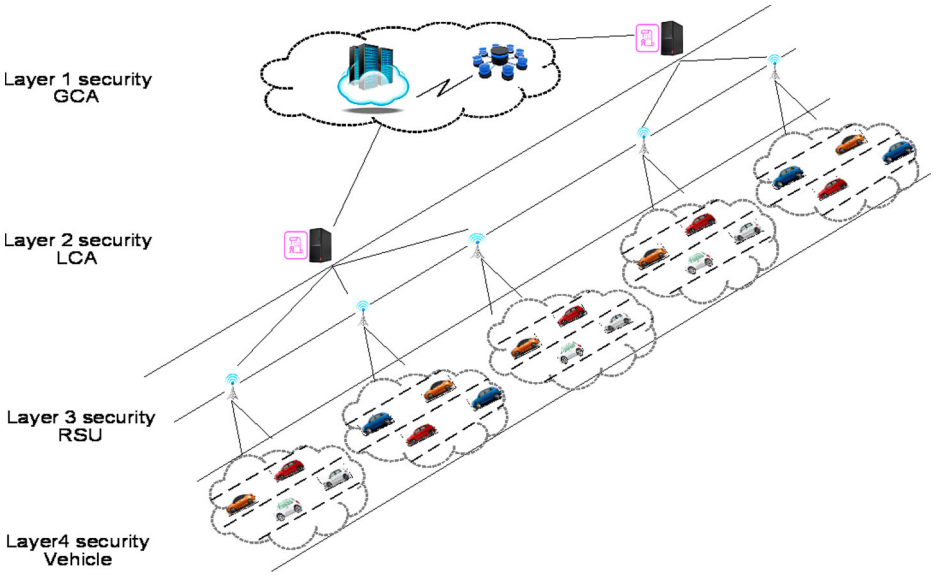

Figure 4. The proposed multi-level security infrastructure maps to corresponding devices.

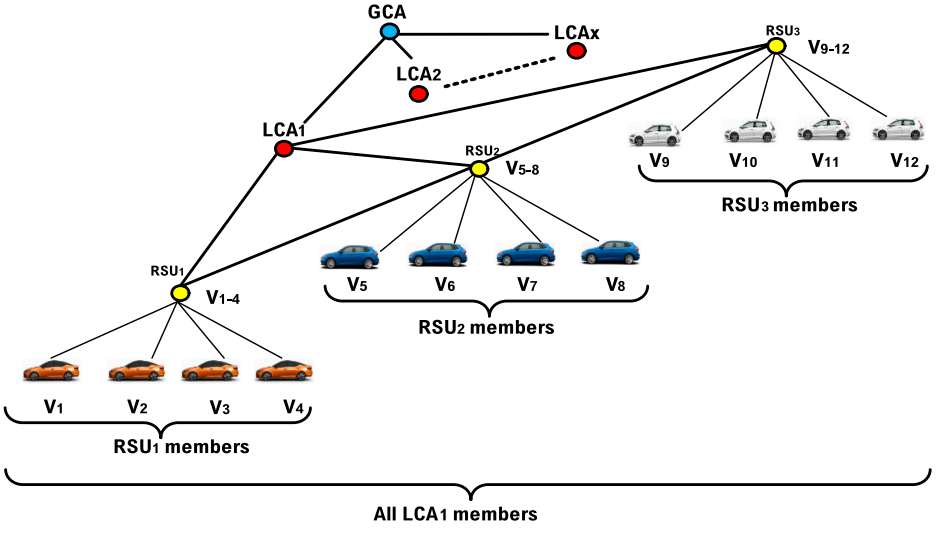

Figure 5. The proposed M-tree structure correlates to the mapped multi-level security authority.

employ an M-tree with multiple degrees to correlate the whole system of the key management of the multi-level security, and thus our presented scheme can efficiently address the dynamic key management.

In this presented research, this study divides M-tree into several groups as the group key management unit. Then, we compute the session key of every leaf node located at the lowest layer, and upward deliver the session key along the route of its ancestor node. Afterwards, the computation of the ancestor node RSU, we can gain the partial group key of level

3 and subsequently forwards it to upper LCA. Eventually, all LCAs perform a similar process and upload the partial group key result to GCA. Consequently, GCA gathers all coming from partial group keys, and eventually calculates the whole IoV system key. In the same way, when an RSU located in level 3 departs, only the replaced RSU passes upwards along its ancestor node to recompute the newer partial group key, subsequently transmits it to LCA and LCA can instantly figure out the newer partial group key, and then forward the newer result to GCA located on the top level. Finally, the whole system comes back to regularity and gains the new system key. Therefore, the proposed scheme indeed accelerates the performance of key resynchronisation. When GCA at the top level collapses, the substituted GCA just needs to gather each LCA's key to quickly restore the whole system key. Figure 5 depicts the M-tree framework of LCA. The entire system consists of the following elements.

(1) Certificate Authority: CA has constant power and plenty of computing and storage resources. It is the secure trusted centre of IoV and is responsible for issuing digital certificates for RSUs, OBUs and vehicles. In this study, we divide CA into GCA (Global CA) and LCA (Local CA). GCA is in charge of the secure trusted centre of the entire IoV, and LCA is in charge of the secure trusted centre of the local area. They serve registration and revocation of certification, and also key management in IoV. Additionally, LCAs also issue the public and private key pairs of OBUs and RSUs, and maintain the identity of OBUs and RSUs.
(2) Road Side Unit: RSU is an equipment located along the roadside, and serves as the middle device to forward information from OBUs to LCA. Therefore, while transmitting information to RSU, we should consider the security issue. The main tasks of RSUs are to (1) forward an encrypted message form OBUs, and (2) assist LCA to efficiently verify the real identity of OBU. Additionally, RSUs need the authorisation of LCA and GCA. Besides, RSUs will not reveal any transmitted messages. Here, this study assumes that RSU is strongly secure and cannot be compromised. At the same time, RSU cannot sign delivered messages in the name of either OBU or LCA. Unlike vehicles, this study assumes that RSUs have no computational, energy and capacity constraints. Therefore, RSUs are able to serve as mediators to transfer messages from OBUs.
(3) On-Board Unit: OBU is a device installed on vehicles, which has to be registered to GCA with certificates, key pairs and public system parameters before joining IoV. Additionally, OBU can communicate with RSUs via wireless communications and generates the necessity for a tamper-proof device in each vehicle. OBU can perform signature, encryption, decryption, and security operations. When the OBUs of vehicles are on the road, they steadily broadcast information periodically, including position, current time, traffic, speed, direction, accident events, etc. The above information will be processed, aggregated and delivered to the drivers and cloud platform via RSUs.

5. The M-tree based key management with multi-modes secure data transmission

This section utterly discusses the main purpose of our presented the key management agreement of the multi-level security. First of all, GLA uses the elliptic curve cryptosystem to produce several key pair sets. Subsequently, this study embeds the key pair into the OBU of

the joining vehicle, and divides vehicles into several groups. Moreover, this study employs the ECDH key agreement to produce session keys between the vehicles depending on the M-tree degree. Afterwards, vehicles deliver the produced session key to its own upper RSU located at level 3. Similarly, the RSU produces the session key between RSUs utilising the ECDH key agreement depending on the M-tree degree, and then delivers the produced session key to its upper LCA. The following describes the process in detail.

In the beginning, the whole system had a prime generator known as the base point P of the Diffie–Hellman agreement. Additionally, each vehicle randomly chooses its individual private key V_x utilising the ECC cryptosystem to produce a correlated public key P_x, and $P_x = V_xP$. Consequently, we combine the private key V_x with the public key P_x as a pair (V_x, P_x) for vehicle X. In the same way, (V_y, P_y) represents the key pair of the vehicle Y. Within the proposed multilevel security scheme, the leaf node is the same as the vehicle owning a key pair. In the presented system, this research supposes the neighbouring vehicle V_1 and the vehicle V_2 are located at level 4 with key pairs (V_1, P_1) and (V_2, P_2), respectively. Then V_1 and V_2 employ ECDH to compute the common session key $S = V_1V_2P$ for both vehicles, and P is the ECDH generator. Afterwards, both vehicles upward send S to their same ancestor node *RSU* located at level 3. For briefness, this study illustrates a binary tree. V_1 and V_2 are located at the lowest level as leaf nodes, and both vehicles can infer the same session key V_1V_2P for V_1 and V_2. Similarly, V_3 and V_4 gain the same session key V_3V_4P. Subsequently, V_1 and V_2 upward send V_1V_2P upward to their common ancestor node RSU_1. Afterwards, V_3 and V_4 upward send the session key V_3V_4P to their common ancestor nodes RSU_2. Since RSU_1, RSU_2, RSU_3 … RSU_z are all located at level 3, after similar processes referred to above, then RSU_1 and RSU_2 together compute the common session key $RSU_1RSU_2P = V_1V_2V_3V_4P$. Subsequently, RSU_1 and RSU_2 respectively upward deliver their common session key to their ancestor node LCA. After receiving it, LCA obtains $V_1V_2V_3V_4P$. In the same way, if there exist other LCAs, they perform the above process and upward to their parent node GCA. Consequently, GCA figures out the system key $V_1V_2V_3V_4 \ldots VzP$, and Z indicates the final vehicle. Because this scheme just utilises addition and multiplication instead of exponential operations, therefore this mechanism saves a significant amount of computing resources.

Moreover, this study takes into account when the vehicle or RSU takes apart from IoV. We have to recalculate the whole system key. However, this study provides a very efficient way to deal with the reorganisation of the system key. This system only needs to recalculate from the level of the departing node along the upward route to the GCA node. At last, the IoV network gains the newer system key. Besides, the necessity of the whole process takes just Log_mN stages, where m stands for the M-tree degree and $\mathrm{Log_m}N$ stands for the M-tree height. Generally, since this study employs ECDH, the benefit is that we almost ignore the M-tree degree, the vehicle just transmits the V_iP value to its ancestor RSU depending on the route from its parent node LCA to GCA, and this mechanism rapidly and efficiently gains the system key. Additionally, our multi-level security mechanism may also offer various managements of various levels and protect devices using different security levels. For instance, before deployment, RSUs need more verification and security. However, we cannot request to equip with ECC key pairs in each vehicle's OBUs, and therefore give more loose restrictions on vehicles. In addition, LCA, GCA, CSCM and cloud servers must pass multiple strict authentication to take part in IoV. Consequently, the presented system is exactly like the hierarchical concept in an enterprise or in an army, and there are various levels with various security permissions.

Moreover, taking into account the rapid changing of topology and the quantity of vehicles in IoV, the proposed mechanism quickly transforms the multi-level security framework into a dynamic framework of M-tree. These include an RSU that can be connected to at most M-1 vehicles. It is known as an M-tree with *M* order, and 2 $\leq$ the degree of nodes $\leq M$. The M-tree possesses the below features.

(1) The degree of the root node is at least 2 degrees.
(2) The M-tree conforms to an M-way search tree.
(3) The bottom level places the leaf node.
(4) The M-tree satisfies $\lceil M/2 \rceil \leq$ the degree of other nodes $\leq$ M, with the exception of root and leaf nodes.

Consequently, the following is that we infer the quantity of members for an M-tree. The full M-tree has the maximum members $M^0 + M^1 + M^2 + \ldots M^{h-1}$, and the M-tree height is *h*.

$$M^0 + M^1 + M^2 + \ldots\ldots M^{h-1} = \frac{M^h - 1}{M - 1} \tag{1}$$

The binary tree named B-tree with degree two is the M-tree's special case, which has the minimum members $1 + 2(r^0 + r^1 + r^2 + \ldots r^{h-2})$, and *r* equals $\lceil M/2 \rceil$, and the B-tree height is *h*.

$1 + 2(r^0 + r^1 + r^2 + \ldots r^{h-2})$

$$= 1 + 2\left(\frac{r(r^{h-2}) - r^0}{r - 1}\right)$$

$$= 1 + 2\left(\frac{(r^{h-1}) - 1}{r - 1}\right) \tag{2}$$

According to the aforementioned inference, we can immediately change the M-tree degree dynamically, depending on the quantity of vehicles to meet the scalability of the practical vehicle environment. Because this research employs the more flexible, scalable and expandable ECC cryptosystem (Jia et al., 2019; Kuljeet et al., 2019), which is also more efficient. The study describes the steps of the proposed multi-level security as follows.

Step1. The identity of every vehicle is V_i with a key pair represented (V_i, P_i). Subsequently, V_i figures out V_iP as its public key, where *P* is a generator of ECC.

Step2. Depending on the algorithm of M-tree, we suppose that the M-tree degree is 3. Figure 6 and Table 3 describe the M-tree structure and notation, respectively. Here, this research illustrates a 3-way tree to present V_{xy1}, V_{xy2} and V_{xy3} cooperatively computing the common session key $V_{xy1}V_{xy2}V_{xy3}P$. For simplicity, this study denotes $V_{xy1}V_{xy2}V_{xy3}P$ as $V_{xy1-3}P$, and subsequently delivers RSU_y with the computed session key. Similarly, the following result indicates that this study repeats the alike procedure and infers the session key.

$$RSU_{y+1} \text{ obtains } V_{x[y+1]1}V_{x[y+1]2}V_{x[y+1]3}P = V_{x[y+1]1-3}P \tag{3}$$

$$RSU_{y+2} \text{ obtains } V_{x[y+2]1}V_{x[y+2]2}V_{x[y+2]3}P = V_{x[y+2]1-3}P \tag{4}$$

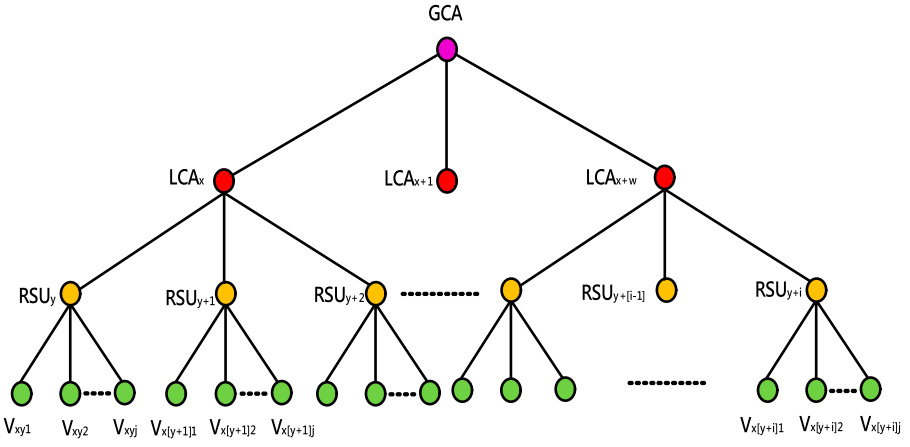

Figure 6. Construct the system key through the mapped M-tree structure.

Table 3. The usage of notations in secure data transmission.

Name	Description
GCA	The global certificate authorisation server
LCA_x	The local certificate authorisation server, *x* denotes the LCA number
RSU_y	The road side unit, *y* indicates the RSU number
V_{xyj}	A vehicle, *x* indicates the belonging of GCA, and *y* indicates the belonging of RSU.
ECDSA	Elliptic curve digital signature algorithm.
HMAC	Hash function code
TS	Time stamp
ToS	Type of service
D_{Vxyj}	Delivered data of V_{xyj}
HMAC(D_{Vxyj})	Hash function code of data D_{Vxyj}
ID_{Vxyj}	Identity of V_{xyj}
||	Data merge
&	And operation
EK_{GK_RSUy}	Encipher/Encipher data using the group key of RSU_y

$$RSU_{y+3} \text{ obtains } V_{x[y+3]1}V_{x[y+3]2}V_{x[y+3]3}P = V_{x[y+3]1-3}P \tag{5}$$

$$RSU_{y+i} \text{ obtains } V_{x[y+i]1}V_{x[y+i]1}V_{x[y+i]1}P = V_{x[y+i]1-3}P \tag{6}$$

Then, RSU_y, $RSU_{y+1}, \ldots, RSU_{y+j}$ independently and securely communicate their session key to the parent node called LCA_x, LCA_{x+1}, LCA_{x+w}, respectively. Upon receipt of the transferred session key, from LCA_x to LCA_{x+w} duplicate the same process and securely transmit individual session keys to the GCA located at the cloud service platform.

Step3. Finally, on the cloud service platform, GCA is going to gain all the session keys coming from LCA_x, $LCA_{x+1}, \ldots, LCA_{x+w}$. Subsequently, GCA computes the whole system key $= V_{xy1}V_{xy2}V_{xy3} \ldots V_{x[y+i]j}P$ through ECDH operations, according to the equations (1) $\sim$ (6).

Following these steps above, then every vehicle, RSU, LCA and GCA can utilise the same system key to encipher and decipher the transmitteVSPd data, and thus complete the secure data transmission of IoV.

5.1. The algorithm of secure data transmission

Initial phase:
Each RSU computes the local group key according to the M-tree structure of ECCDH.
Execution phase:
At the beginning, the source vehicle V_{xyi} determines whether the destination vehicle V_{wzj} belongs to the same LCA_x group.
Perform x & w operations.
If $x = w$, **then** the source vehicle V_{xyi} and the destination vehicle V_{wzj} belong to the same LCA_x authority, and then perform y & z operation.

If $y = z$, **then** the source vehicle V_{xyi} and the destination vehicle V_{wzj} belong to the same RSU_y authority. Subsequently, V_{xyi} and V_{wzj} perform secure data transmission using the group key of RSU_x to ensure security.

Else V_{xyi} and V_{wzj} belong to the same LCA_x but belong to the different RSU. Then RSU_y and RSU_z exploit LCA_x to perform the group exchange agreement via ECDH to achieve the same group key of LCA_x for the future secure data transmission of V_{xyi} and V_{wzj}.

Else the source vehicle V_{xyi} and the destination vehicle V_{wzj} belong to the different LCA_x authority and RSU authority, then RSU_y and RSU_z exploit ECDSA signature agreement to perform the secure data transmission.

5.2. Secure data transmission modes

For increasing the efficiency of secure data transmission, this study proposes the group key mode and the ECDSA mode to deal with different scenarios. The group key mode accelerates the speed of secure communication once the members obtain the same group key. Only the vehicle owns the same group key that can take part in the communication. Others cannot take part in secure communication without the group key. Nevertheless, the ECDSA mode gives an optional way to achieve secure data transmission. When the members of IoV are abundant and huge, the time complexity of obtaining the system key takes longer time. The member can exploit the ECDSA mode to deal with secure data transmission, since the member has already owned the certificate, and thus the vehicle can sign the transmitted data to protect the transmitted data. The detailed presentation is as follows.

(1) The secure data transmission of IoV belongs to the same RSU authority: the group key agreement via RSU

When the source vehicle V_{xy1} and destination V_{xyi} are located in the same RSU_y, as shown in the red dot line of Figure 7, and therefore they adopt the group key of RSU_y for the secure communication of all members. Initially, both vehicles perform ECDH cryptography agreement. For simplicity, here illustrates a B-tree structure. V_{xy1} and V_{xy2} execute the ECDH agreement and obtain the common session key $V_{xy1}V_{xy2}{*}P$, V_{xy3} and V_{xy4} obtain the common session key $V_{xy3}V_{xy4}{*}P$, the remaining vehicle V_{xyi} execute the same operation and forwards its key to RSU_y, and then RSU_y calculates the common group key for $GK_{RSUy} = V_{xy1}V_{xy2}V_{xy3}V_{xy4} \ldots V_{xyj}{*}P$. Subsequently, if V_{xy1} would like to deliver the data to

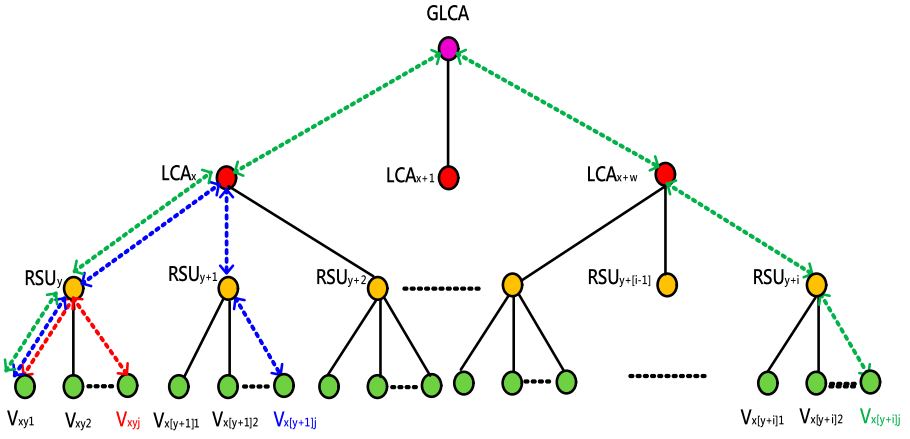

Figure 7. Construct the system key through the mapped M-tree structure.

V_{xyj}, the detailed secure procedure is as follows.

$$\#\boldsymbol{V_{xy_1}} \rightarrow \boldsymbol{RSU_y}$$

$$\#\mathbf{EK}_{\boldsymbol{GKRSUy}}[\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{Vxy_1}}|\boldsymbol{D_{Vxy_1}}||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}})]$$

Initially, V_{xy1} adopts the hash function to calculate HMAC(D_{Vxy1}), and then enciphers the aggregations of *ToS*, *TS*, the routing path, original data and HMAC(D_{Vxy1}) via the group key RSU_y. Subsequently, V_{xy1} sends the enciphered result to RSU_y.

$$\#\boldsymbol{RSU_y} \rightarrow \boldsymbol{V_{xy_i}}$$

$$\#\mathbf{EK}_{\boldsymbol{GKRSUy}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}ID_{RSUy}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}})]$$

Once the enciphered data are received, RSU_y deciphers the received data utilising the group key RSU_y. Subsequently, RSU_y performs the hash function on the original data D_{Vxy1}, and verifies if it equals to the received HMAC(D_{Vxy1}). If there is a difference, it means during the transmission, the data are tampered with other vehicles. Otherwise, RSU_y puts its *ID* into the routing path ($ID_{Vxy1}ID_{RSUy}$), and then enciphers the merged data of *ToS*, *TS*, the rouging path, the original data ($D_{Vxy1}|D_{RSUy}$) and its HMAC($D_{Vxy1}|D_{RSUy}$) via the group key of RSU_y. Eventually, RSU_y sends the enciphered result to V_{xyj}.

After receiving the enciphered data, V_{xyj} deciphers them using the group key of RSU_y, and verifies the integrity of HMAC($D_{Vxy1}|D_{RSUy}$). If the HMAC is correct, then the secure data transmission is complete.

(2) The secure data transmission of IoV belongs to the same LCA and different RSU: the group key agreement via LCA

When the source vehicle V_{xy1} and destination $V_{x(y+1)j}$ are located in the different *RSU*, but in the same LCA_x, as shown in the blue dot line of Figure 7. In this situation, this study has to perform the group key exchange operation to achieve the new group key for *LCA*. In the beginning, RSU_y obtains its group key $GK_{RSUy} = V_{xy1}V_{xy2} \ldots V_{xyj} * P$ using the ECDH agreement, and RSU_{y+1} obtains its group key $GK_{RSUy+1} = V_{x(y+1)1}V_{x(y+1)2} \ldots V_{x(y+1)j} * P$.

Subsequently, RSU_y and RSU_{y+1} individually forward their group key to LCA_x. After receiving, LCA_x performs the group key exchange operations and obtains the newer group key $GK_{LCAx} = GK_{RSUy} * GK_{RSUy+1} = V_{xy1}V_{xy2} \ldots V_{xyj}V_{x(y+1)1}V_{x(y+1)2} \ldots V_{x(y+1)j} * P$. Additionally, this study employs the common group key GK_{LCAx} to execute the secure data transmission from V_{xy1} to $V_{x(y+1)j}$ as follows.

$$\#\boldsymbol{V_{xy_1} \rightarrow RSU_y}$$
$$\#\ \mathbf{EK}_{\boldsymbol{GKLCAx}}[\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{Vxy_1}}|\boldsymbol{D_{Vxy_1}}||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}})]$$

At the beginning, V_{xy1} adopts the hash function to calculate HMAC(D_{Vxy1}), and then enciphers the merged data of *ToS*, *TS*, the rouging path, the original data and HMAC via the group key GK_{LCAx}. Subsequently, V_{xy1} sends the enciphered result to RSU_y.

$$\#\boldsymbol{RSU_y \rightarrow LCA_x}$$
$$\#\mathbf{EK}_{\boldsymbol{GKLCAx}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}}|\boldsymbol{ID_{RSUy}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}})]$$

Once the enciphered data are received, RSU_y deciphers the received data with the group key GK_{LCAx}. Subsequently, RSU_y performs the hash function on the original data D_{Vxy1}, and verifies if it equals to the received HMAC(D_{Vxy1}). If there is a difference, it means during the transmission, the data are tampered with other vehicles. Otherwise, RSU_y put its *ID* into the routing path ($ID_{Vxy1}ID_{RSUy}$), and then enciphers the merged data of *ToS*, *TS*, the rouging path, the original data ($D_{Vxy1}|D_{RSUy}$) and its HMAC($D_{Vxy1}|D_{RSUy}$) via the group key GK_{LCAx}. Eventually, RSU_y forwards the enciphered result to LCA_x.

$$\#\boldsymbol{LCA_x \rightarrow RSU_{y+1}}$$
$$\#\mathbf{EK}_{\boldsymbol{GKLCAx}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}}|\boldsymbol{ID_{RSUy}}|\boldsymbol{ID_{LCAx}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}})]$$

Once the enciphered data are received, LCA_x deciphers the received data with the group key GK_{LCAx}. Subsequently, LCA_x performs the hash function on the original data ($D_{Vxy1}|D_{RSUy}$), and verifies if it equals to the received HMAC($D_{Vxy1}|D_{RSUy}$). If there is a difference, it means during the transmission, the data are tampered with other vehicles. Otherwise, LCA_x puts its *ID* into the routing path ($ID_{Vxy1}|ID_{RSUy}|ID_{LCAx}$), and then enciphers the merged data of *ToS*, *TS*, the rouging path, the original data ($D_{Vxy1}|D_{RSUy}|D_{LCAx}$) and its HMAC($D_{Vxy1}|D_{RSUy}|D_{LCAx}$) via the group key GK_{LCAx}. Eventually, LCA_x forwards the enciphered result to RSU_{y+1}.

$$\#\boldsymbol{RSU_{y+1} \rightarrow V_{x(y+1)j}}$$
$$\#\mathbf{EK}_{\boldsymbol{GKLCAx}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}}|\boldsymbol{ID_{RSUy}}|\boldsymbol{ID_{LCAx}}|\boldsymbol{ID_{RSUY+1}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{RSUY+1}})|$$
$$|\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{RSUY+1}})]$$

Similar to the above procedure, RSU_{y+1} deciphers the enciphered data and verifies the integrity of the received data. Afterwards, RSU_{y+1} puts the related data into the corresponding column of the combination. Finally, RSU_{y+1} forwards the enciphered result to $V_{x(y+1)j}$.

After receiving the enciphered data, $V_{x(y+1)j}$ performs the above similar procedure, and verifies the integrity of HMAC($D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{RSUY+1}$), and completes the secure data transmission from V_{xy1} to $V_{x(y+1)j}$.

(3) The secure data transmission of IoV belongs to the different LCA: the ECDSA mechanism

When the source vehicle V_{xy1} and destination $V_{x(y+i)j}$ are located in the different LCAs, as shown in the green dot line of Figure 7. For reducing the computing time of the group key exchange, this study adopts ECDSA mechanism to secure data transmission. The detailed procedure is as follows.

$$\#\boldsymbol{V_{xy_1}} \rightarrow \boldsymbol{RSU_y}$$
$$\#\mathbf{EK}_{\boldsymbol{Sign\_Vxy_1}}[\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{Vxy_1}}|\boldsymbol{D_{Vxy_1}}||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}})]$$

First, V_{xy1} signs the transmitted data $[ToS|TS|ID_{Vxy1}|D_{Vxy1}||\text{HMAC}(D_{Vxy1})]$ with its own private key using ECDSA, and then forwards the signature result to RSU_y.

$$\#\boldsymbol{RSU_y} \rightarrow \boldsymbol{LCA_x}$$
$$\#\mathbf{EK}_{\boldsymbol{Sign\_RSUy}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}ID_{RSUy}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}})]$$

Upon receipt of the transmitted data, RSU_y deciphers the signature result utilising the public key of V_{xy1}, and verifies the original data $[ToS|TS|ID_{Vxy1}|D_{Vxy1}||\text{HMAC}(D_{Vxy1})]$ whether they are correct. If the transmitted data are tampered with, then discard this data, otherwise RSU_y adds its ID_{RSUy} into the routing path and computes $\text{HMAC}(D_{Vxy1}|D_{RSUy})$ using the hash function. Then, RSU_ysigns the entire data using its private key, and delivers the signature result to LCA_x.

$$\#\boldsymbol{LCA_x} \rightarrow \boldsymbol{GCA}$$
$$\#\mathbf{EK}_{\boldsymbol{Sign\_LCAx}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}ID_{RSUy}ID_{LCAx}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}})]$$

After receiving, LCA_x adopts the RSU_y's public key to decipher the enciphered data, and then verifies the original data of $[ToS|TS|(ID_{Vxy1}ID_{RSUy})|(D_{Vxy1}|D_{RSUy})||\text{HMAC}(D_{Vxy1}|D_{RSUy})]$ whether it is correct. Subsequently, LCA_x adds its identity ID_{LCAx}into the routing path and computes $\text{HMC}(D_{Vxy1}|D_{RSUy}|D_{LCAx})$. Consequently, LCA_x signs the entire data using its private key, and then sends the signature result to GCA.

$$\#\boldsymbol{GCA} \rightarrow \boldsymbol{LCA_{x+w}}$$
$$\#\mathbf{EK}_{\boldsymbol{Sign\_GLCA}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}ID_{RSUy}ID_{LCAx}ID_{GLCA}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{GLCA}})|$$
$$|\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{GLCA}})]$$

When GCA obtains the transmitted data, it performs the above similar procedure and delivers the signature result to LCA_{x+w}.

$$\#\boldsymbol{LCA_{x+w}} \rightarrow \boldsymbol{RSU_{y+i}}$$
$$\#\mathbf{EK}_{\boldsymbol{Sign\_LCAy}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}ID_{RSUy}ID_{LCAx}ID_{GLCA}ID_{LCAx+w}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{GLCA}}$$
$$|\boldsymbol{D_{LCAx+w}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{GLCAx}}|\boldsymbol{D_{LCAx+w}})]$$

After receiving, LCA_y utilises the public of GCA to decipher the enciphered data, and then verifies the original data of $[ToS|TS|(ID_{Vxy1}ID_{RSUy}ID_{LCAx}ID_{GLCA})|(D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{GLCA})|$

$|\text{HMAC}(D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{GLCA})]$ whether they are tamped with, and repeats the similar procedures, adds its identity ID_{LCAx+w} into the routing path and computes $\text{HMAC}(D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{GLCA}|D_{LCAx+w})$. Eventually, LCA_{x+w} signs the entire data using its private key, and then sends the signature result to RSU_{y+i}.

$$\#\boldsymbol{RSU_{y+i}} \rightarrow \boldsymbol{V_{x(y+j)j}}$$

$$\#\mathbf{EK}_{\boldsymbol{Sign\_RSUy+i}}[\boldsymbol{ToS}|\boldsymbol{TS}|(\boldsymbol{ID_{Vxy_1}ID_{RSUy}ID_{LCAx}ID_{GLCA}ID_{LCAx+w}ID_{RSUy_{+i}}})|(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{GLCA}}|\boldsymbol{D_{LCAx+w}}|\boldsymbol{D_{RSUy+i}})||\mathbf{HMAC}(\boldsymbol{D_{Vxy_1}}|\boldsymbol{D_{RSUy}}|\boldsymbol{D_{LCAx}}|\boldsymbol{D_{GLCA}}|\boldsymbol{D_{LCAx+w}}|\boldsymbol{D_{RSUy+i}})]$$

When RSU_{y+i} receives the transmitted data, it adopts the public key of LCA_y to decipher the enciphered data, after that verifies the original data whether they are tamped with. Subsequently, RSU_{y+i} adds its identity ID_{RSUy+i} into the routing path and computes $\text{HMAC}(D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{GLCA}|D_{LCAx+w}|D_{RSUy+i})$. Consequently, RSU_{y+i} signs the entire data using its private key, and then sends the signature result to $V_{x(y+j)j}$.

Finally, when $V_{x(y+j)j}$ receives the data coming from RSU_{y+i}, it utilises the public key of RSU_{y+i} to decipher the enciphered data, and performs the hash function on the source data $(D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{GLCAx}|D_{LCAx+w}|D_{RSUy+i})$, and verifies if they equal to the received $\text{HMAC}(D_{Vxy1}|D_{RSUy}|D_{LCAx}|D_{GLCA}|D_{LCAx+w}|D_{RSUy+i})$. If they are different, it means during the transmission, the data are tampered with other vehicles. Otherwise, the original data from each vehicle are correct. According to the proposed mechanism, this study achieves the secure data transmission of IoV belonging to the different *LCAs*.

In an actual IoV environment, because V2V, V2I, V2P and V2G usually strongly demand rapidity, in addition to the topological architecture of the IoV is changing quickly. This research employs the proposed dynamic M-tree key management agreement and takes advantage of the ECDH and ECDSA to solve the problems of insufficient computing resources on the vehicle, RSUs and the cryptographic system issues between the vehicles.

(4) The secure data transmission framework for IoV and the cloud server platform

When data goes through RSU and Internet to arrive at the cloud service platform, the submitted data must be protected against modification. This research uses a PKI infrastructure to securely transmit data and perform identity authentication. Furthermore, we exploit ECDSA to finish the secure transmission of data between vehicles, RSUs and cloud services on the cloud platform.

Because PKI utilises the digital signature of EDCSA to verify the participator identity, this scheme is highly appropriate to authenticate the identity of infrastructure devices, roadside devices, intelligent lights, signage and cloud platform servers. In this way, this research exploits digital signatures to guarantee the authenticity of the information source and to prevent impersonation, man-in-the-middle, and information replay attacks. This allows us to improve the reliability of the securely transmitted data from IoV infrastructure to a cloud services platform.

At the deployment stage, each vehicle member initially requests a certificate via the GCA of the cloud platform. The GCA server is in charge of the issuance of certificates to vehicles, RSU devices, vehicle service provider (VSP) and users. At the same time, the GCA server also is responsible for renewing certificates, revoking certificates and maintaining CRL (certificate revocation list).

When vehicles are required to demand a cloud service, this research utilises RSU to reach the cloud platform to perform secure data transmission. Because RSU and VSP are stationary installations and seldom altered, the identity of participants, such as RSU and VSP, may be confirmed by digital signatures. In the meantime, RSU also may help perform secure data transmission. This lowers the calculation load on OBU of vehicle. The framework of the GCA server is shown on the top left corner in Figure 8.

Firstly, at the early stage of this study, a secure virtual machine is chosen as a GCA server from the cloud platform.

Stage 1. The RSU of the road equipment, the vehicles and the VSP first apply to GCA for a certificate.

Stage 2. If a vehicle requires a VSP service, initially, the vehicle delivers the source data, the type of service *ToS* demand, itself identity *ID*, and the time stamp *TS*. This computation uses a hash function to generate the HMAC code of the above data as HMAC(*ToS*|*TS*|*ID*|*source data*). Subsequently, this study merges the plaintext data with the HMAC outcome, and employs ECDSA to sign and encipher the entire data via a private key, then after that delivers the signature result to the cloud platform.

Stage 3. The result of the signature is delivered to the boundary CSCM (cloud service category manager) via Internet routers.

Stage 4. Afterwards, CSCM utilises the vehicle's public key for deciphering enciphered data, then computes the newer HMAC′′ code of the received source data, and then verifies if HMAC′′ equals to the received HMAC. When HMAC′′ equals to HMAC, that indicates the source data are accurate and have not changed as the transmission process progresses.

Stage 5. Depending on the *ToS* from vehicles, CSCM will send the request to the respective application service on the cloud platform.

This research assumes when the M_A vehicle transmits the data and request to CSCM along V2N, RSU_1 and Internet routers. Once the delivered data has been received, subsequently CSCM delivers this request to the relevant cloud service, depending on the request of *ToS* coming from the vehicle. Figure 8 shows the path of the blue colour. The following

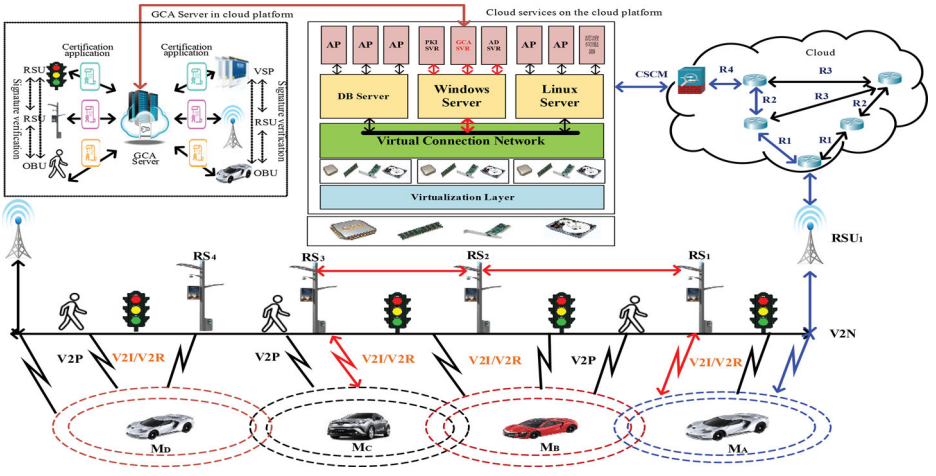

Figure 8. The whole framework of the secure data transmission between IoV and the cloud service platform.

steps describe in detail the secure transmitting of data from the vehicle to the cloud service platform.

If the vehicle M_A would like to deliver the M(DV_A) message to the cloud services platform, M_A initially delivers M(DV_A) to the connected base station RSU_1 by V2N, as indicated by the path $M_A \leftrightarrow$ V2N $\leftrightarrow$ RSU$_1$ $\leftrightarrow$ R$_1$ $\leftrightarrow$ R$_2$ $\leftrightarrow$ R$_4$ $\leftrightarrow$ CSCM $\leftrightarrow$ cloud service platform in Figure 8. Firstly, the vehicle M_A places the required *ToS*, the time stamp, itself *ID* and the transmitted data M(DV_A) in the respective column, subsequently enters the aforementioned data as the hash function input, and produces the HMAC($ToS|TS|ID_{MA}|$M(DV_A)). This research then uses M_A's private key to sign and encipher HMAC($ToS|TS|ID_{MA}|$M(DV_A)) utilising ECDSA and merges the original data with the signature result. Finally, M_A sends all data to RSU_1.

$$\#\boldsymbol{M_A} \rightarrow \boldsymbol{RSU_1}$$

$$\#\mathbf{EK}_{\boldsymbol{SigMA}}[\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{MA}}|\boldsymbol{M}(\boldsymbol{DV_A})||\mathbf{HMAC}(\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{MA}}|\boldsymbol{M}(\boldsymbol{DV_A}))]$$

Once the enciphered data are received, RSU_1 instantly deciphers the enciphered data utilising the M_A's public key, after that utilises the hash function to figure out the obtained source ($ToS|TS|ID_{MA}|$M(DV_A)) to gain a newer HMAC$''$. Afterwards, we check HMAC$''$ with the obtained source HMAC, and validate if HMAC$''$ equals to the source HMAC. When HMAC$''$ does not match HMAC, this represents that the source data are altered while transmitting, else the received data are valid, and then RSU_1 delivers EK$_{SigMA}$[$ToS|TS|ID_{MA}|$M(DV_A)||HMAC($ToS|TS|ID_{MA}|$M(DV_A))] to the R_1 cloud router.

$$\#\boldsymbol{RSU_1} \rightarrow \boldsymbol{R_1} \rightarrow \boldsymbol{R_2} \rightarrow \boldsymbol{R_4} \rightarrow \boldsymbol{CSCM}$$

$$\#\mathbf{EK}_{\boldsymbol{SigMA}}[\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{MA}}|\boldsymbol{M}(\boldsymbol{DV_A})||\mathbf{HMAC}(\boldsymbol{ToS}|\boldsymbol{TS}|\boldsymbol{ID_{MA}}|\boldsymbol{M}(\boldsymbol{DV_A}))]$$

Once the enciphered data are received, router R_1 forwards it to R_4 passing through routers R_2 and R_3 depending on the routing table. During this phase, the router can utilise its security protocol to protect the network packet.

Once R_4 obtains the enciphered data, R_4 forwards it to the border CSCM server. Meanwhile, CSCM uses the M_A's public key to decipher the enciphered data, and subsequently computes the newer HMAC$''$ code of the received source data [$ToS|TS|ID_{MA}|$M(DV_A)] utilising the hash function. Afterwards, this research verifies HMAC$''$ with the received source HMAC, and subsequently determines if HMAC$''$ = HMAC. When it makes any difference, this represents that the source data are altered during the transmitting path of $RSU_1 \rightarrow R_1 \rightarrow R_2 \rightarrow R_4 \rightarrow$CSCM. If not, CSCM sends the request to the relevant cloud application service, depending on the requested *ToS* from M_A.

Generally, the proposed framework using the ECDSA signature protocol can securely transmit data between V2V, V2I, V2P, and V2G in IoV, and also secure the transmitting of data between the OBU of the vehicle and the cloud server platform as well. Finally, this research prevents harmful vehicles or malicious network devices from joining the transmission of data.

(5) The architecture of secure data transmission for the cloud server platform: secure Map Reduce operations

When the CSCM sends the data to the related cloud service server (CSS) to perform Map/Reduce operation according to *ToS*, in order to confirm the identity and security of

Mapper and Reducer for participants deployed in the cloud, this study uses VM Ware to simulate and construct several virtual servers to jointly execute Hadoop and Map/Reduce operations. Because there are so many servers in the cloud, the system must verify the identity of the participants before it cooperatively performs security operations. This study uses Kerberos to authenticate the participating servers before performing secure Map/Reduce operations to keep the data secure (Mukti et al., 2018; Subodh & Rajesh, 2020), as shown in Figure 9.

At the beginning of this system, we selected a secure GCA (as the key distribution centre), Mappers and Reducers from the cloud servers with mutual trust. The account database of GCA DB stores the account secrets of virtual servers such as CSS, Mappers, Reducers and other cloud service servers as follow-up identification operations. When the CSS requests the Map/Reduce operation, at the initial registration stage: CSS, Mappers and Reducers generate a group of personal passwords through the random number generator and store them into the account database of the GCA server.

Step1. Apply for a session key Initially, CSS requests a session key *SK* from the GCA VM server.

Step2. The master key and the enciphered session key: When GCA (a key distribution centre) receives the request, and then calculates a session key *SK*. In order to ensure that only CSS and Mappers/Reducers know *SK*, GCA respectively extracts the password of CSS and the Mappers/Reducers which take part in this operation, from the account DB. Subsequently, GCA performs a hash operation and respectively obtains the master key of CSS and the Mappers and Reducers, such as $MK_{CSS(/Map/Red)} = \text{HASH}(\text{Password}_{CSS(/Map/Red)})$. After that, CSS uses itself master key MK_{CSS} to encipher the *SK* via a symmetric encryption scheme and obtains $\text{EK}(SK)_{MK-of-CSS}$, and then uses MK_{Map} and MK_{red} to perform symmetric encryption operations on *SK*, ID_{CSS}, IP_{CSS}, its *Domain* and *TS* to respectively obtain $\text{EK}[SK|ID_{CSS}|IP_{CSS}|Domain|TS]_{MK-of-Map}$ and $\text{EK}[SK|ID_{CSS}|IP_{CSS}|Domain|TS]_{MK-of-Red}$. For simplicity, we present it as $\text{EK}[SK|ID_{CSS}|IP_{CSS}|Domain|TS]_{MK-of-Map(/Red)}$ also named

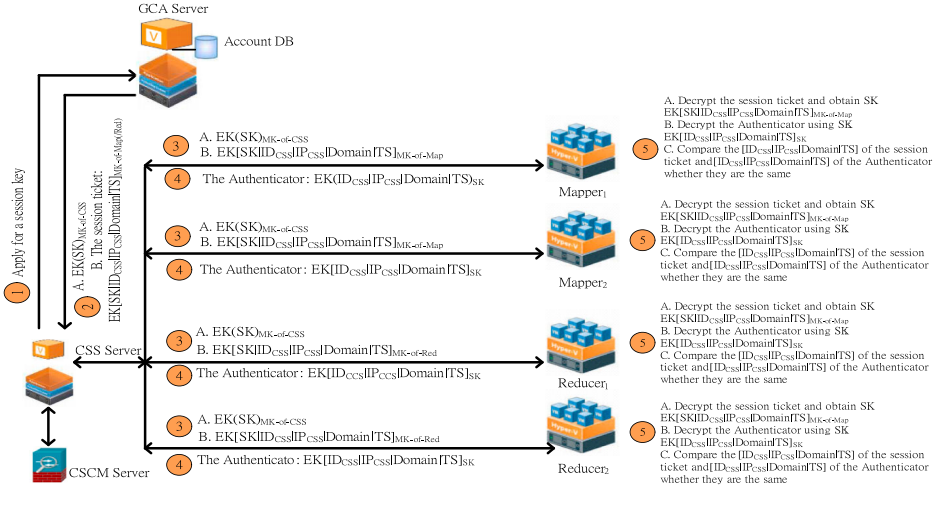

Figure 9. The Kerberos authentication protocol for the Mappers and Reducers of the cloud service platform.

the session ticket. So far, GCA has a session ticket that consists of *SK*, *CSS* and *Mappers*/*Reducers* encrypted by *MK*. Eventually, GCA delivers EK$(SK)_{MK-of-CCS}$ and the session ticket EK$[SK|ID_{CSS}|IP_{CSS}|Domain|TS]_{MK-of-Map(/Red)}$ to CSS.

Step3. When CSS receives the above message, then forwards EK$(SK)_{MK-of-CSS}$ and the session ticket EK$[SK|ID_{CSS}|IP_{CSS}|Domain|TS]_{MK-of-Map(/Red)}$ to the Mappers/Reducers, which take part in this operation.

Step4. In order to provide more effective identity authentication, after receiving the encrypted session key *SK*, CSS uses its own MK_{CSS} to decipher the received EK$(SK)_{MK-of-CSS}$ from GCA and obtain *SK*. Then the authenticator is generated, and contains the ID_{CSS}, IP_{CSS}, *Domain* and *TS* to be the duration of the subsequent confirmation request. Subsequently, this study enciphers the authenticator using *SK* and obtains EK$[ID_{CSS}|IP_{CSS}|Domain|TS]_{SK}$. After that, CSS delivers the authenticator EK$[ID_{CSS}|IP_{CSS}|Domain|TS]_{SK}$ to the Mappers/Reducers, which take part in this operation.

Step5. When the Mappers/Reducers receive the above message, then decipher EK$[SK|\ ID_{CSS}|IP_{CSS}|Domain|TS]_{MK-of-Map(/Red)}$ using itself $MK_{Map(/Red)}$ to obtain *SK*, and then can decipher the received authenticator EK$[ID_{CCS}|IP_{CCS}|Domain|TS]_{SK}$ using *SK* to compare the $[ID_{CSS}|IP_{CSS}|Domain|TS]$ of the session ticket and the deciphered result $[ID_{CSS}|IP_{CSS}|Domain|TS]$ of the authenticator whether they are the same to ensure the identity of CSS. After CSS and Mappers/Reducers verify each other's identities, and this study can perform the secure Map/Reduce operations.

The above secure data transmission modes provide a flexible and scalable option to protect data. The proposed mechanisms finish the secure data transmission from vehicles to cloud, and also integrate cloud platform with IoV. When a source vehicle wants to deliver data to a destination vehicle, under the same RSU or LCA, this study adopts a group key mechanism to secure data. Nevertheless, a huge IoV can exploit ECDSA to improve the efficiency of secure data transmission between vehicles. Additionally, this study also considers the secure transmission of data between vehicles and the cloud service platform. We provide a complete secure architecture to achieve this target, and make sure the identity of VM members joining the secure map/reduce operations.

6. The analysis of security and performance

In this section, we analyze and estimate our presented secure data transmission scheme. While transmitting data, the system must guarantee data integrity and avoid modification. Furthermore, we must make sure of the identity of participants in order to avoid impersonation from taking part in the process of data transmission. The following describes the detailed analysis of security.

(1) Signature verification proof

According to the ECDSA algorithm, the receiver has to verify the identity of the sender. This study infers the following operations and ensures the signature of sender.

$$\begin{aligned} C = (x_c, y_c) &= u_1 \times G + u_2 \times P_k \\ &= u_1 \times G + u_2 \times k \times G = (u_1 + u_2 \times k) \times G \end{aligned}$$

$$
\begin{aligned}
&= (e \times s^{-1} + r \times s^{-1} \times k) \times G = (e + r \times k) \times s^{-1} \times G \\
&= (e + r \times k) \times [(k \times r + e) \times i^{-1}]^{-1} \times G \\
&= (e + r \times k) \times (k \times r + e)^{-1} \times (i^{-1})^{-1} \times G \\
&= i \times G = (x_R, y_R) = (r, y_R)
\end{aligned}
$$

From the above inference, this study can prove that indeed $r = x_c$, and thus this study can ensure the signature is correct.

(2) Identity verification

In this study, although there is no authentication process when GCA made a request to CSS. However, if *CSS_A* pretends or claims to be *CSS_B*, then *CSS_A* will obtain the session key *SK* of *CSS_B* and Mapper/Reducers. However, the subsequent GCA uses *CSS_B*'s password to calculate the master key of *CSS_B* and uses the master key of *CSS_B* to encrypt *SK*.

However, only those who really know the *CSS_B*'s password can decipher and obtain the session key *SK*. Therefore, *CSS_A* cannot decipher and obtain the session key *SK* without *CSS_B*'s password, so it can confirm the identity of participants.

(3) Confidentiality of Vehicle-to-Vehicle

Once malicious vehicles are interested in communicating with each other, through the route of transmission, only the two connected vehicles own the same session key via the ECDH key exchange protocol, after that utilise the session key for enciphering and deciphering transmitted data. Just the vehicle or RSU owns the common session key that can encipher or decipher the transmitted data (Zhang et al., 2021). The vehicle with no session key which cannot be included in the process of secure transmission, in spite of the harmful vehicle impersonates others, and thus this study can guarantee confidentiality of information transmission for Vehicle-to-Vehicle.

(4) Confidentiality of Vehicle-to-Group

Just vehicles that have the same group key are allowed to participate in the group to communicate (Dan et al., 2004; Zhang et al., 2011), while vehicles with no group key are not allowed to join the communication. Even though an outside vehicle is listening or receiving a transmission, it is unable to decipher the encrypted message by the group vehicle as it has no group key. This method prevents messages from leaking or altering and ensures the confidentiality of messages between vehicles and groups.

(5) System scalability

In this study, we utilise a dynamic M-tree structure to achieve the group key management. The benefit of M-tree is flexibility and scalability. As this research expands the quantity of vehicles, the whole system constantly retains the $O(\log_M N)$ time complicity, where M stands for the M-tree degree and N stands for the quantity of vehicles, to figure

out the group key. Moreover, even a great number of vehicles join the communication, the system just needs $\log_M N$ stages to deal with the resynchronisation of the group key and obtain the common group key, as shown in Table 4. This mechanism is very well adapted to the wide range of vehicles in the IoV environment. Figure 10 shows the dashed green square as the new connecting vehicles, and the investigation result indicates that the system still needs the same quantity of phases for handling key interchange and computing the group key. Besides, when a vehicle leaves, the whole system only has to compute the subtree of the leaving vehicle, then can reconstruct the newer group key of the vehicles. Therefore, this mechanism has flexibility and scalability.

(6) Efficiency of the key management

In this study, we compare the proposed key management of M-tree with None-tree and B-tree schemes on re-synchronising the system key. From Table 4, if vehicles participate in

Table 4. Resynchronisation cost of the system key for IoV.

Cost	Vehicle/RSU	Employ ECDH to perform M-tree key agreement	Employ ECDH to perform B-tree key agreement	Employ DH to perform none-tree key agreement
Operational costs	Vehicle join	2 unicasts and 1 multiplication	2 unicasts and 1 multiplication	C_2^N
	Vehicle leave	2 unicasts and 1 multiplication	2 unicasts and 1 multiplication	C_2^N
	RSU join	2 unicasts and 1 multiplication	2 unicasts and 1 multiplication	C_2^N
	RSU leave	2 unicasts and 1 multiplication	2 unicasts and 1 multiplication	C_2^N
Resynchronising phases	Vehicle join	$\text{Log}_M(N)$	$\text{Log}_2(N)$	C_2^N
	Vehicle leave	$\text{Log}_M(N)$	$\text{Log}_2(N)$	C_2^N
	RSU join	$\text{Log}_M(N-1)$	$\text{Log}_2(N-1)$	$C_2^{\log_2 N}$
	RSU leave	$\text{Log}_M(N-1)$	$\text{Log}_2(N-1)$	$C_2^{\log_2 N}$

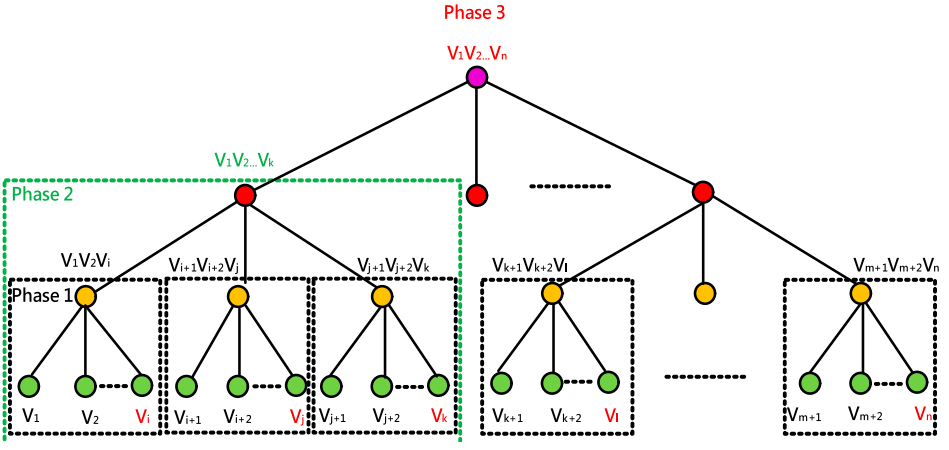

Figure 10. The resynchronisation phase of the group key when vehicles joining or leaving IoV.

or take apart from IoV, the entire system needs to resynchronise the system key. As a result, the ECDH with M-tree and B-tree schemes outperform the None-tree scheme on communication and operational costs, since the M-tree scheme just takes two unicast phases and one multiplication phase. However, the None-tree scheme takes C_2^N phases utilising DH.

(7) Synchronisation time of a vehicle joining or leaving

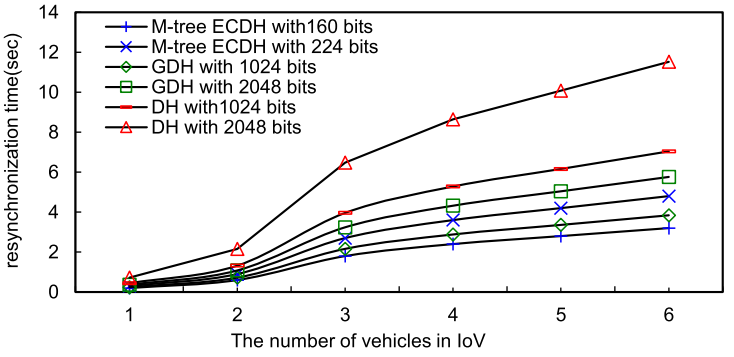

Figure 11. The comparison of constructing the system key on M-tree, GDH and DH schemes.

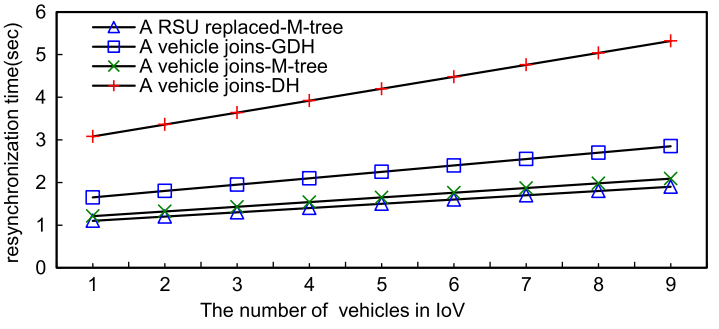

Figure 12. The comparison of reconstructing the system key when a vehicle joins IoV.

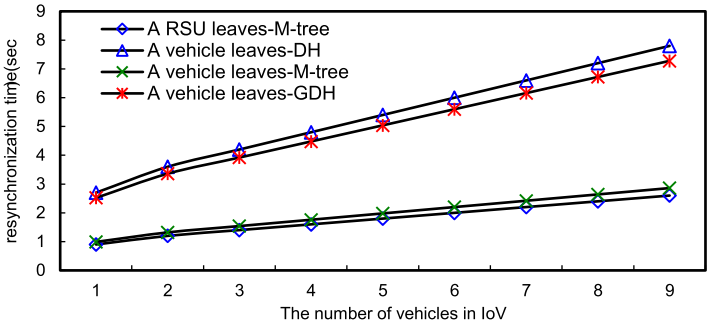

Figure 13. The comparison of reconstructing the system key when a vehicle leaves IoV.

Initially, this system needs to construct the system key. Figure 11 depicts the construction time of the system key on different schemes. As a result, the M-tree based scheme outperforms GDH and DH under the same secure levels. Since the M-tree structure only needs to update the keys along the parent nodes, GDH and DH have to exchange the share key for each other that needs C_2^N phases, and then obtain the system key. Additionally, a vehicle or an RSU joins/leaves the IoV, the entire system needs to reconstruct the entire system key. Figure 12 demonstrates that a vehicle joins IoV. The M-tree based scheme takes less resynchronisation time to reconstruct the system key then GDH and DH, since M-tree only needs to recalculate the joining node and the parent node. Similarly, when a vehicle leaves, the M-tree based scheme also outperforms GDH and DH, as shown in Figure 13.

7. Conclusions

Thanks to the development of 5G, automated driving without a driver will be right around the corner. Because IoV has an open communication environment, personal information is disclosed to the wireless network. Therefore, the problem of information security for IoV is going to be a significant theme. The proposed agreement of the multi-level security key management for IoV relies on the structure of an M-tree with the capacity to dynamically adjust the framework depending on the rapidly changing IoV topology. Additionally, this mechanism speeds up the time to synchronise the reconstruction of the system key, and decreases the quantity of phases to resynchronise the system key. Moreover, the provided key management for the multi-level security is quite appropriate to adaptable and expandable IoV environments. As well, the system effectively lowers operation and communication costs. In comparison to a B-tree, this study employs an M-tree that is more effective than the conventional key management. Moreover, the use of addition and multiplication operations can take the place of complex exponential operations and decrease the computing loading of OBU and RSU, and is better suited for IoV without plenty of calculating resources. Consequently, this study utilises a smaller key length of ECDSA to reach Diffee-Hellman or RSA security level, and secure the transmitted data among IoV devices and cloud service platforms. Thus this study can ensure information security and achieve secure IoV.

Data availability statement

The data that support the findings of this study are available on request.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Ali, I., Gervais, M., Ahene, E., & Li, F. (2019). A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs. *Journal of Systems Architecture*, *99*. https://doi.org/10.1016/j.sysarc.2019.101636

Amit, D., Neeraj, K., Mukesh, S., Mohammad, S. O., & Kuei, F. H. (2016). Secure message communication among vehicles using elliptic curve cryptography in smart cities. *International Conference on Computer, Information and Telecommunication Systems*. https://ieeexplore.ieee.org/document/7546385

Daisy, P. B. T., Albert, R. S., & Vimal, J. A. (2015). Elliptic Curve Cryptography based Security Framework for Internet of Things and Cloud Computing. *International Journal of Computer Science and Technology*, https://www.researchgate.net/publication/305913586

Dan, B. X., Boyen, H., & Shacham. (2004). Short group signatures. *Lecture Notes in Computer Science*, *3152*, 41–55. https://doi.org/10.1007/978-3-540-28628-8_3

Dhanashree, T., Rohan, S., Het, S., Nikita, N., & Vishal, P. (2018). Prominence of ECDSA over RSA Digital Signature Algorithm. *2nd International Conference on IoT in Social, Mobile, Analytics and Cloud*. https://ieeexplore.ieee.org/document/8653689/

Hong, Z., Jingyu, W., Jie, C., & Shun, Z. (2016). Efficient conditional privacy-preserving and authentication scheme for secure service provision in VANET. *Tsinghua Science and Technology*, *21*(6), 620–629. https://ieeexplore.ieee.org/document/7787005 https://doi.org/10.1109/TST.2016.7787005

Hu, J., Liang, W., Hosam, O., Hsieh, M. Y., & Su, X. (2021). 5GSS: A framework for 5G-secure-smart healthcare monitoring. *Connection Science*, 1–23. https://doi.org/10.1080/09540091.2021.1977243

Hu, X., Guobin, Z., Zhong, C., & Fagen, L. (2013). Efficient communication scheme with confidentiality and privacy for vehicular networks. *Computers and Electrical Engineering*, *39*(6), 1717–1725. https://doi.org/10.1016/j.compeleceng.2012.11.009

Hua, Y. L. (2021). Integrate the hierarchical cluster elliptic curve key agreement with multiple secure data transfer modes into wireless sensor networks. *Connection Science*, 1–27. https://doi.org/10.1080/09540091.2021.1990212.

Insaf, U., Muhammad, A. K., & Mohammed, H. A. (2021). Anonymous certificateless Signcryption Scheme for secure and efficient deployment of Internet of vehicles. *Sustainability Journal MDPI*, *13*(19), 10891. https://doi.org/10.3390/su131910891

Jia, W., Jianqiang, L., Hui, H. W., Leo, Y. Z., Lee, M. C., & Qiuzhen, L. (2019). Dynamic scalable elliptic curve cryptographic scheme and its application to in-vehicle security. *IEEE Internet of Things Journal*, *6*(4), 5892–5901. https://ieeexplore.ieee.org/document/8463502

Joon, Y. L., Sung, J. Y., Myeong, H. K., Young, H. P., Sang, W. L., & Bo, H. C. (2020). Secure key agreement and authentication protocol for message confirmation in vehicular cloud computing. *Applied Sciences*, *10*, 18. https://doi.org/10.3390/app10186268

Kulathunge, A., & Dayarathna, H. (2019). Communication framework for vehicular ad-hoc networks using Blockchain: Case study of Metro Manila Electric Shuttle automation project. *In Proceedings of the 2019 International Research Conference on Smart Computing and Systems Engineering*. pp. 85–90. https://ieeexplore.ieee.org/document/8842814

Kuljeet, K., Sahil, G., Georges, K., François, G., Syed Hassan, A., & Mohsen, G. (2019). A secure, lightweight, and privacy preserving authentication scheme for V2G connections in Smart Grid. *IEEE Conference on Computer Communications Workshops*. https://ieeexplore.ieee.org/document/8845140

Lin, H. Y., Hsieh, M. Y., & Li, K. C. (2016). Flexible group key management and secure data transmission in mobile device communications using elliptic curve Diffie-Hellman cryptographic system. *International Journal of Computational Science and Engineering*, *12*(1), 47. https://doi.org/10.1504/IJCSE.2016.074558

Maxim, R., & Jean, P. H. (2007). Securing vehicular ad hoc networks. *Journal of Computer Security. Special Issue on Security of Ad Hoc and Sensor Networks*, *15*(1), 39–68. https://doi.org/10.5555/1370616.1370618

Mu, H., Mengli, Z., Pengzhou, C., Zhikun, Y., & Haixin, Q. (2021). Implementing an efficient secure attribute based encryption system for IoV using association rules. *Symmetry Journal MDPI*, *13*(7), https://doi.org/10.3390/sym13071177

Mukti, R., Nigar, S., Himel, D., & Hossain, A. (2018). A new version of Kerberos authentication protocol using ECC and Threshold Cryptography for Cloud Security. *Joint 7th International Conference*

on Informatics, Electronics & Vision and 2nd International Conference on Imaging, Vision & Pattern Recognition. https://ieeexplore.ieee.org/document/8641010

Nils, G., Arun, P., Avinderpal, W., Hans, E., & Sheueling, C. S. (2004). Comparing elliptic Curve Cryptography and RSA on 8-Bit CPUs. *Lecture Notes in Computer Science*, *3156*, 119–132. https://doi.org/10.1007/978-3-540-28632-5_9

Nisha, M., Priyadarsi, N., Arushi, A., Xiangjian, H., Deepak, P., & Puthal, D. (2018). Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks. In *Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering*. pp. 674–679. https://ieeexplore.ieee.org/document/8455967

Pandi, V., Maria, A., Sergei, A. K., & Joel, J. P. C. R. (2022). An anonymous batch authentication and key exchange protocols for 6G enabled VANETs. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 1630–1638. https://doi.org/10.1109/TITS.2021.3099488

Ram, R. A., & Manoj, A. (2013). Elliptic curve Diffie-Hellman Key Exchange Algorithm for securing hypertext information on Wide Area Network. *International Journal of Computer Science and Information Technologies*, *4*(2), 363–368. http://citeseerx.ist.psu.edu/viewdoc/summary?doi = 10.1.1.299.9663

Sathishkumar, N., & Rajakumar, K. (2017). A study on vehicle to vehicle collision prevention using Fog, Cloud, Big Data and Elliptic Curve Security based on Threshold Energy Efficient Protocol. *Second International Conference on Recent Trends and Challenges in Computational Models*. https://ieeexplore.ieee.org/document/8057548

Shoma, K., Yasuyuki, N., Shunsuke, M., & Thomas, A. (2015). Volunteer computing for solving an Elliptic Curve Discrete Logarithm Problem. *Third International Symposium on Computing and Networking*. https://ieeexplore.ieee.org/document/7424699

Subodh, C., & Rajesh, D. (2020). Secured map building using Elliptic Curve Integrated Encryption Scheme and Kerberos for Cloud-based Robots. *Fourth International Conference on Computing Methodologies and Communication*. https://ieeexplore.ieee.org/document/9076465

Vijayakumar, P., Azees, M., Chang, V., Deborah, J., & Balamurugan, B. (2017). Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks. *Cluster Computing*, *20*(3), 2439–2450. https://doi.org/10.1007/s10586-017-0848-x

Xia, X., Ji, S., Vijayakumar, P., Shen, J., & Rodrigues, J. (2021). An efficient anonymous authentication and key agreement scheme with privacy-preserving for smart cities. *International Journal of Distributed Sensor Networks*, *17*(6). https://doi.org/10.1177/15501477211026804.

Xiao, L., Xie, S., Han, D., Liang, W., Guo, J., & Chou, W. K. (2021). A lightweight authentication scheme for telecare medical information system. *Connection Science*, *33*(1), 1–17. https://doi.org/10.1080/09540091.2021.1889976

Xiaodong, L., & Rongxing, L.. (2015). GSIS: Group signature and ID-based signature-based secure and privacy preserving protocol. In *Vehicular Ad Hoc network security and privacy* (pp. 21–49). Wiley-IEEE Press. https://onlinelibrary.wiley.com/doi/10.1002/9781119082163.ch2

Xiaodong, L., Xiaoting, S., Pin, H. H., & Xuemin, S. (2007). GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on Vehicular Technology*, *56*(6), 3442–3456. https://ieeexplore.ieee.org/document/4357367 https://doi.org/10.1109/TVT.2007.906878

Xu, Z., Liang, W., Li, K. C., Xu, J., & Jin, H. (2021). A blockchain-based roadside unit-assisted authentication and key agreement protocol for Internet of vehicles. *Journal of Parallel and Distributed Computing*, *149*, 29–39. https://doi.org/10.1016/j.jpdc.2020.11.003

Yasin, G., & Erkan, A. (2021). Design and implementation of an efficient Elliptic Curve Digital Signature Algorithm (ECDSA). *IEEE International IOT, Electronics and Mechatronics Conference*. https://ieeexplore.ieee.org/document/9422589

Yong, J. K., Yong, M. K., Yong, J. C., & Hyong, C. O. (2013). An efficient bilinear pairing-free certificateless two-party authenticated Key Agreement Protocol in the eCK model. *Journal of Theoretical Physics and Cryptography*, *3*, 1–10. https://arxiv.org/abs/1304.0383

Zhang, C., Ho, P., & Tapolcai, J. (2011). On batch verification with group testing for vehicular communications. *Wireless Network*, *7*(8), 1851–1865. https://doi.org/10.1007/s11276-011-0383-2

Zhang, L., Xu, J., Obaidat, M., Li, X., & Vijayakumar, P. (2021). A PUF-based lightweight authentication and key agreement protocol for smart UAV networks. *IET Communications*, 1–18. https://doi.org/10.1049/cmu2.12295

Zhe, L., Jian, W., Zhi, H., & Hwajeong, S. (2017). Efficient Elliptic Curve Cryptography for embedded devices. *ACM Transactions on Embedded Computing Systems*, *16*(2). https://doi.org/10.1145/2967103