CVICA: Coordinated Vehicle Infrastructure Cryptography Architecture with Fine-Grained Access Control

Shiqi Ou, Xiangyu Liu

***Abstract*—The development of Coordinated Vehicle Infrastructure System (CVIS) provides more efficient and safer traffic services. However, the frequent communications among cloud center, connected vehicles, and connected infrastructure bring risks in privacy preservation and information confidentiality. To largely apply coordinated vehicle infrastructure technologies, an appropriate cryptography system needs to be designed to ensure security issues. This paper dedicates to establishing a cryptography architecture which fits in CVIS well, namely coordinated vehicle infrastructure cryptography architecture (CVICA), to guarantee message confidentiality and authenticity. Attributed-based encryption (ABE) is used to send confidential data to users based on their attributes. Besides, sensitive personal data is hidden to protect users' privacy. Identity-based signature and double-layer signature are used to maintain message authenticity and prevent the system from being invaded by fake data or malicious messages. Rigorous security proofs and efficiency analyses of CVICA are provided. Furthermore, the proposed CVICA is compared with other secure schemes in CVIS. Simulation results show that the proposed CVICA has high efficiency, high practicability, and low latency.**

***Index Terms*—Coordinated vehicle infrastructure system, Cryptography architecture, Fine-grained access control, Privacy protection, Authenticity, Traceability.**

I. Introduction

CYBER security plays a vital role in traffic systems, especially in a system with connected vehicle technologies. To improve system connectivity, system data including user information, travel trajectories, detecting data, and so on is received, used, and sent in the system, and the protection of the data is important to protect privacy and ensure security. (some data to support they are important and the result will be very bad)

The development of connected vehicle technologies enables the prosperous development of coordinated vehicle infrastructure system (CVIS). In CVIS, vehicles, infrastructure, and other traffic participants are connected. Real-time data and historical data of traffic participants are available [1] to allow advanced traffic management and control, which cater to time-varying traffic conditions to provide safer, more efficient, and more environmentally friendly traffic services.

Security of CVIS is one of the biggest challenges to its implementation in the real world. Connectivity and data-availability of CVIS require parties in the system to exchange information frequently, which brings security risks to the system. The control units can obtain real-time data and historical data, and then input to traffic control algorithms. Afterwards, the algorithms output management strategies and send back to the user ends, infrastructure ends, or device ends for the sake of traffic efficiency and safety improvement. In the process of communications among these parties, private information, confidential algorithms, and other sensitive data are exposed in the environment where malicious users and devices exist [2]. Thus, privacy protection and information confidentiality are vital to the realization of CVIS.

There are two main types of security and privacy issues in CVIS, namely, confidentiality and authenticity [3]. Roughly speaking, message confidentiality means that sensitive information, such as personally identifiable data and user travel records, should be hidden from an eavesdropper/attacker in the network. Authenticity requires that each message received should be verified, preventing malicious users from spreading fake data or malicious messages and disturbing the reliability of CVIS (in which case, of course, they should be traced for a punishment) [4]. Therefore, we need to guarantee message confidentiality and authenticity simultaneously to design an appropriate cryptographic system that fits in CVIS well. However, most existing schemes in CVIS considered either confidentiality or authentication, rather than both. Note that communication issues are not the focus of this study, and thus the security and privacy issues discussed in this paper can be applied to all kinds of CVIS communications networks (i.e., vehicular ad-hoc networks).

Digital signatures [5] are commonly used cryptographic tools to maintain message authenticity in CVIS. Namely, a sender signs messages using its secret signing key, and other devices can check the validity of the message-signature pair with the sender's public verification key. One drawback of the traditional signature scheme is the management burden of the verification keys, because a device needs to maintain a verification key list whose size is linear in the number of devices in the system. In CVIS, there are thousands of vehicles, detectors, road-side units (RSUs), multi-access edge computing (MEC), and other devices. This would cause a

This work was primarily completed in 2022, when Shiqi Ou was a Ph.D. candidate at Tongji University focusing on transportation, and Xiangyu Liu was a Ph.D. candidate at Shanghai Jiao Tong University focusing on cryptography. Both authors shared the belief that collaborating on something meaningful before graduation would be worthwhile. The technical novelty in this paper is limited—particularly in cryptography, and it is not intended for publication. Rather, it serves as a personal memento of their Ph.D. journey and a witness for Hoco.

Shiqi Ou: samkie@foxmail.com.

Xiangyu Liu: xiangyu1994liu@gmail.com.

great memory cost as the number of devices in the system expands. To deal with this problem, identity-based signature (IBS) [6] was introduced.[1] In an IBS system, verification can be processed with a master public key (shared among all devices in CVIS) and a unique identity. The widely-used certification chain on the Internet (e.g., X.509 [7]), can be viewed as a generic construction of IBS, from regular signature schemes.

To guarantee confidentiality of data (e.g., destinations of the travels in the network [8] [9]), symmetric and asymmetric encryption schemes are widely used in CVIS. However, traditional encryption schemes have an "all-or-nothing" access control to an encrypted message (i.e., recovering the entire message with a secret key, or revealing nothing). While in CVIS, a fine-grained access control to an encrypted message is desirable. Concretely, sensitive data can only be accessed by the vehicles with certain attributes, while other vehicles do not have the access. For example, in a traffic accident, only the authority vehicles (e.g., ambulances and police vehicles) have access to the identity and personal information of the drivers. Therefore, users are with different attributes, which decide the type of data that they have access to. It is necessary to provide data encryption with fine-grained access in CVIS.

Attribute-based encryption (ABE) [10] [11], is introduced to provides more fine-grained access control in such a setting. ABE is an extension of identity-based encryption (IBE) [6] [12], where a message is encrypted under an access policy. A device's private key is associated with a set of attributes (we mainly focus on ciphertext-policy ABE [11] in this paper). A device can decrypt an encrypted message only if its attributes match the access policy. Usually the policy is an access tree with leaf nodes representing attributes and non-leaf nodes representing threshold gates (e.g., AND, OR, and k of n). As shown in Section IV, almost all access policies in CVIS can be well expressed in this form.

We propose a coordinated vehicle infrastructure cryptography architecture (CVICA), to solve the aforementioned problems. More precisely, CVICA has the following attractive properties.

- **Attribute-based confidentiality**. Fine-grained access control are provided for users/devices in CVICA. Namely, messages are transmitted after encryption, and only users with attributes satisfying the access policy can decrypt and have the access to the messages.
- **Privacy**. Sensitive data (e.g., personally identifiable data and users' locations) for users are hidden from eavesdroppers/attackers in CVICA.
- **Traceability**. If a malicious uploaded data is detected, the manager of CVICA can trace the real identity of the sender.
- **Authenticity**. The validity of each message received by a device can be verified, preventing the system from being invaded by fake data or malicious instructions/messages.

[1]Another potential solution is to download the verification key from PKI each time. However, This interaction will cost a great amount of round travel time and hence is very impractical.

We analyze the security of CVICA via rigorous proofs, and simulate it to test the efficiency. Comparisons among other secure CVIS schemes and experiment results show that the proposed CVICA has high efficiency, high security, and high practicability.

This paper is organized as follows. Related works are presented in Section II, and some relevant cryptographic preliminaries are deferred to Appendix A. The scenario of CVIS and the security requirements are provided in Section III, followed by the detailed introduction of the algorithms designed for CVICA in Section IV. The analysis of CVICA is given in Section V with detailed security proofs of authenticity and attribute-based confidentiality in Appendixes C and D, respectively. In Section VI, experimental results are presented to show the performance of the proposed encryption algorithms in the CVICA. Finally, the conclusion is presented in Section VII.

II. Related Works

Pseudonyms mechanism [13] is a common method for privacy preservation. By using a pseudonym, an eavesdropper/attacker in CVIS cannot link a package of sensitive information to the pseudonym holder's real identity. Privacy in this setting is also referred as anonymity. Nevertheless, message authentication is still performed on every received wireless message by conducting verification for a valid signature [14], and only messages that have been successfully verified are processed further. There are plenty of standardized signature schemes over the past decades, e.g., (EC)DSA [15], SM2 [16], GOST [17], to name a few. The concept of IBS was first introduced by Shamir in [6], to relieve the key management burden. Kiltz and Neven [18] concluded three generic constructions for IBE, based on certification paradigm [19], identification protocol, and hierarchical IBE, respectively.

Lots of studies emerging in recent years proposed a variety of schemes/protocols for authentication. Huang et al. [20] proposed PACP, an authentication protocol using the pseudonyms mechanism. Chim et al. [21] developed a group communication protocol to allow vehicles to authenticate and securely communicate with others in a group. Adaptive privacy-preserving authentication in vehicular networks was first considered by Sha et al. in [22], where a trade-off between privacy and efficiency was provided to achieve better performance. In [4], Wang and Li established an attack model for actual vehicles in multi-network, and designed NOTSA, a three-level architecture to provide authentication. All schemes above took use of signatures (or IBS) for authenticity. There were some other protocols that introduced group signatures to make an authentication, e.g., [23]. However, most group signature schemes depend on heavy cryptographic constructions and non-standard hardness assumptions, hence not reliable in real world.

Besides privacy, traceability [24], [25], key management [26], and other cryptographic issues [27] also need to be considered in the design of a secure CVIS. Sun et al. [28] proposed a security system to achieve both privacy (which is desired by vehicles) and traceability (which is required by

law enforcement authorities), using IBE, IBS, and Shamir's polynomial secret sharing scheme [29]. Brecht et al. [27] took signatures, key expansion algorithms and other typical cryptographic technologies together and designed the system SCMS, in which a series of optional parameters were provided to balance efficiency, security, and privacy. Recently, some techniques in block chain were introduced [30], to get rid of the dependence on trusted certificate authorities.

In terms of flexible data-sharing mechanism in CVIS, few works can be found over the past decades. Ulybyshev et al. [31] considered secure data communication in autonomous V2X systems, where both role-based and attribute-based access control were provided to deploy a more fine-grained management on sensitive data. However, the method in [31] relied on (partially) homomorphic encryption [32], a cryptographic tool which was powerful enough to support operations over encrypted data, but still exhibited prohibitive overhead.

ABE is a lightweight cryptographic tool that supports flexible access control with a low cost on computation and communication complexities. There are two categories of ABE, key-policy (KP) ABE [33], where the access policy is related to a user's secret key, and ciphertext-policy (CP) ABE [11], where the access policy is related to a ciphertext. The concept of ABE was first introduced by Sahai and Waters [10]. After that, a series of variant ABE schemes came up in need of different applications [34], [35].

III. Problem Description

Figure 1 shows a coordinated vehicle infrastructure system (CVIS) environment including cloud center (CC), detective devices, road-side infrastructure, and connected (automated) vehicles. Note that the urban traffic network with four intersections in Figure 1 serves as an example, but scenarios in both urban areas and highways are considered. Not only connected (automated) vehicles are on the road, but also ordinary vehicles exist. CC serves as the centered controller. It has full information over the system from weather to real-time traffic condition and has the highest level of controlling priority on all the devices. It does not need to be at a physical location, but on the cloud. Detectors perceive traffic conditions, and they play the roles of information detection, including cameras, radars, weather detectors, and so on. The detected traffic conditions include individual data (e.g., individual locations, speeds, and accelerations) and network information (e.g., average speed, volume, and density of segments). The road-side infrastructure includes road-side units (RSU), multi-access edge computing (MEC), and signal controllers. RSU serves as an information transaction unit, exchanging data from and to MEC and the vehicle ends. MEC is a road-side device where most of the traffic management and control algorithms are stored and computed. It receives data from detectors and vehicles. With inputs, the traffic algorithms generate traffic management instructions and messages. These messages are further sent to the vehicles by broadcasting. OBUs are equipped in moving vehicles to receive real-time information or instructions via RSUs. Thus, traffic management and control are more convenient to be implemented with real-time and accurate data. The task is to develop a security system in this environment to ensure data confidentiality and message authenticity. Vehicles, as roles of information reception, include on-board units (OBU) in vehicles. For simplicity, we make the following two reasonable assumptions in this paper.

- **Reliable network.** Transmitting messages between different parties, despite the delay, will eventually be received.
- **Trusted authority.** There exists a trusted authority, the cloud center (CC), that is reliable and would never be compromised by the attackers on the network.

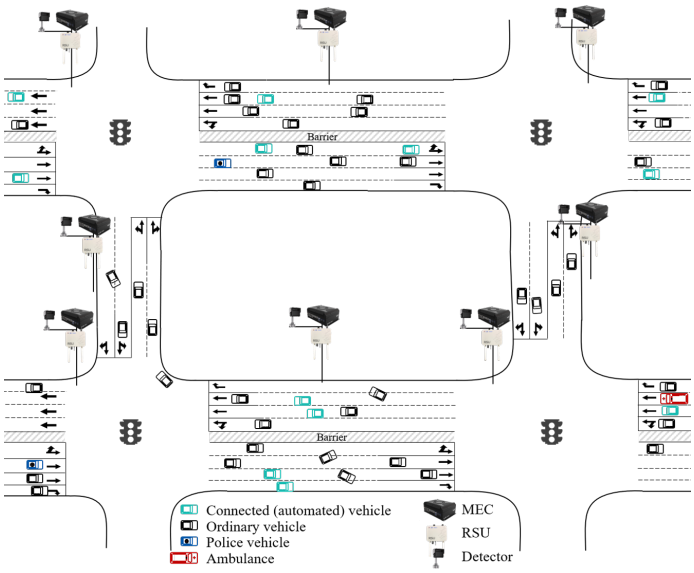

Fig. 1: Problem scenario.

Take traffic accident alert as an example, as shown in Figure 2. A traffic accident occurs and is detected by the detector. The sensitive information of the vehicles and drivers, including their identities, personal data, and health status, is sent to MEC. MEC then calculates the affecting range of the accident and broadcasts the accident-related data to the RSUs within the affecting area. There are regular vehicles, ambulances, police vehicles, and rescue vehicles on the road, but only the ambulances, police vehicles, and small size rescue vehicles have access to the sensitive information of the accident. With the accident-related data, medical treatment can be done, and policemen and rescuers can deal with the accident immediately. Simultaneously, the sensitive information can be protected from other regular vehicles.

Notations used in this paper are listed in Table I. Mathematical assumptions and definitions are also presented as follows. Let $\kappa \in \mathbb{N}$ denote the security parameter. For $\mu \in \mathbb{N}$, define $[\mu] := \{1, 2, ..., \mu\}$. Denote by $x := y$ the operation of assigning y to x. Denote by $x \xleftarrow{\$} \mathcal{X}$ the operation of sampling x uniformly at random from a set $\mathcal{X}$. For an algorithm $\mathcal{A}$, denote by $y \leftarrow \mathcal{A}(x; r)$, or simply $y \leftarrow \mathcal{A}(x)$, the operation of running $\mathcal{A}$ with input x and randomness r and assigning the output to y. The symbol $||$ represent concatenate, and PPT is short for probabilistic polynomial-time.

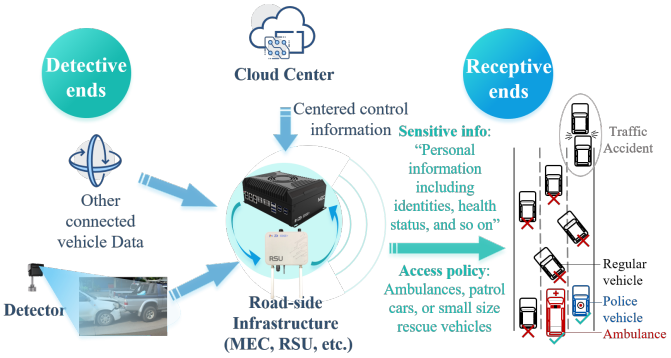

Fig. 2: Coordinated vehicle infrastructure system architecture.

TABLE I: Notations used in this paper.

Notations	Descriptions
CC	cloud center
MEC	multi-access edge computing
D	detector
U	user (on-board units in vehicles)
$\widetilde{\mathsf{U}}$	pseudonym of user U
SIG	signature
ABE	attribute-based encryption
msg	message that needs to be broadcasted
$\mathbb{A}$	access structure (policy) of message msg
ct	ciphertext
msk	master secret key
pk	public key
sk	secret key
$\perp$	failure symbol, which denotes a failed decryption
att	attribute
γ	attribute set
rt	root note of $\mathbb{A}$
δ	node in $\mathbb{A}$
parent(δ)	parent node of δ
index(δ)	index number of δ
Y	set of leaf nodes in $\mathbb{A}$
$cert$	certificate

A. Security Requirements

We require CVICA to have the following security properties.

- **Anonymity.** MECs and eavesdroppers in CVICA cannot recognize the real identity from the uploaded traffic data.
- **Authenticity.** Only registered users and detectors can upload traffic data to MECs.
- **Traceability.** CC can trace the real identity of the sender in case a malicious data is detected.
- **Attribute-based confidentiality.** Only users with attributes satisfying the access policy can decrypt a ciphertext and read the message.

IV. Coordinated Vehicle Infrastructure Cryptography Architecture

As mentioned above, there are four parties in CVICA, namely, cloud center CC, multi-access edge computing MEC, user U, and detector D.

- CC is the trusted authority in CVICA. It initializes the architecture, generates the main secret key, processes registration from users U, derives secret keys for them, and traces the real identity from a malicious data if necessary.
- MEC MEC is the road-side computing device. It stores traffic control and management algorithms, and works as an independent computation unit. MEC receives real-time traffic condition data from detector D and anonymous user $\widetilde{\mathsf{U}}$, generates instructions/messages msg accordingly, and broadcasts them after encryption.
- D is a detector that detects traffic conditions and incidents, and uploads the information to MEC. Note that we have no anonymity requirement for detectors.
- User U is a vehicle (driver) in the architecture. It samples a pseudonym $\widetilde{\mathsf{U}}$, registers it to CC and gets the related secret key back. Furthermore, it uploads real-time driving data in $\widetilde{\mathsf{U}}$'s name, and receives the encrypted broadcast data from MEC.

Now we describe the algorithms of CVICA in details. For a high-level understanding, we note that the system consists of three parts, the attribute-based encryption (ABE) part that is used for access control, the signature (SIG) part that is used for authenticity, and the symmetric encryption (SE) part that is used for enhance the encryption efficiency. See Appendix A for their formal definitions. Moreover, the ABE algorithm used in this paper is a modified version of [11], and the signature algorithm is the GenDSA signature [36]. We do not specify SE algorithm in this section because it is succinct enough in hybrid encryption compared with ABE and SIG. The structural flowchart is shown in Figure 3.

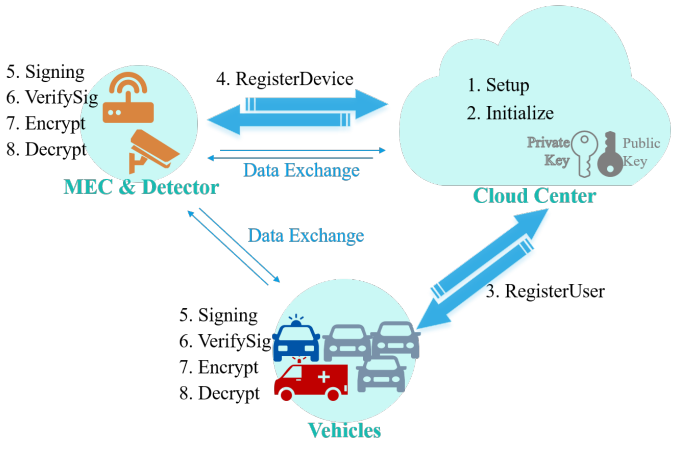

Fig. 3: Algorithm structure in CVICA.

A. Setup

The cloud center CC setups the system via algorithm $Setup(\cdot)$, and publishes the public information used in

CVICA. $Setup(\cdot)$ inputs the security parameter 1^κ, and outputs public parameter pp, key pairs $(vk^{\mathsf{SIG}}, msk^{\mathsf{SIG}})$ (for the signature part) and $(pk^{\mathsf{ABE}}, msk^{\mathsf{ABE}})$ (for the ABE part). Then, CC publishes pp, vk^{SIG} and pk^{ABE}.

The detailed algorithm $Setup(\cdot)$ works as follows.

$(\mathsf{pp}, vk^{\mathsf{SIG}}, msk^{\mathsf{SIG}}, pk^{\mathsf{ABE}}, msk^{\mathsf{ABE}}) \leftarrow Setup(1^\kappa)$: CC samples a group $G = (\mathbb{G}_0, q, g_0)$ as well as a pairing group $G_{PG} := (\mathbb{G}_1, \mathbb{G}_2, q, e, g)$, where $\mathbb{G}_0, \mathbb{G}_1$ and $\mathbb{G}_2$ are both cyclic groups of order q, g_0 is a generator of $\mathbb{G}_0$, g_1 is a generator of $\mathbb{G}_1$, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ is a bilinear map with bilinearity and non-degeneracy [37]. Then it chooses four hash functions f, H_1, H_2, and H_3, where $f : \mathbb{G}_0 \mapsto \mathbb{Z}_q$, $H_1 : \{0,1\}^* \mapsto \mathbb{Z}_q$, $H_2 : \{0,1\}^* \mapsto \mathbb{G}_1$, $H_3 : \mathbb{G}_2 \mapsto \{0,1\}^\ell$ (ℓ is the bit-length of symmetric key in SE). The public parameter is set to be $\mathsf{pp} := (G, G_{PG}, f, H_1, H_2, H_3)$.[2][3]

Next, for the signature part, CC samples $x \xleftarrow{\$} \mathbb{Z}_q$, and sets $msk^{\mathsf{SIG}} := x$, $vk^{\mathsf{SIG}} := X = g_0^x$. For the ABE part, CC samples $\alpha \xleftarrow{\$} \mathbb{Z}_q$, and sets $msk^{\mathsf{ABE}} := g_1^\alpha$, $pk^{\mathsf{ABE}} := e(g_1, g_1)^\alpha$.

B. Register

CC deals with the register request from a user or a device via algorithm $Register(\cdot)$, and returns the secret key(s). For a user's request, $Register(\cdot)$ inputs msk^{SIG}, msk^{ABE}, user U's random pseudonym $\widetilde{\mathsf{U}}$[4] and attribute set γ, and outputs secret keys sk^{SIG} and sk^{ABE}. For a device's request (a detector or an MEC), $Register(\cdot)$ inputs msk^{SIG} and the device's identity id, and output the signing key sk^{SIG}. During this process, CC records $(\mathsf{U}, \widetilde{\mathsf{U}})$ in the user list.

We introduce attributes (of users) and access structures to achieve attribute-base confidentiality. We refer to Appendix B for a toy example of the access structure. The attribute set $\gamma = \{\mathsf{att}_1, ..., \mathsf{att}_j, ...\}$ is related to a specific a user, where the binary string att_j denotes a specific attribute hold by the user. Sometimes we may abbreviate att_j as j without ambiguity. When a user registers to CVICA, CC will define the attribute set γ for this user according to its real identity and functionality. For example, a small police vehicle may hold an attribute set $\gamma = \{\text{“}police\text{”}, \text{“}small\text{”}, \text{“}flashing\ lights\text{”}, \text{“}dispatch\ systems\text{”}\}$.

The detailed algorithm $Register(\cdot)$ works as follows.

Denote by $(sk^{\mathsf{SIG}}, sk^{\mathsf{ABE}}) \leftarrow Register(msk^{\mathsf{SIG}}, msk^{\mathsf{ABE}}, \mathsf{U}^{(i)}, \widetilde{\mathsf{U}}^{(i)}, \gamma)$ the registration from a user. For the signature part, CC samples $x^{(i)}$ and computes $X^{(i)} := g_0^{x^{(i)}}$, generates a signature $cert$ on message $\widetilde{\mathsf{U}}^{(i)} || X^{(i)}$ as follows: $w \xleftarrow{\$} \mathbb{Z}_q$, $c := f(g_0^w)$, $h := H_1(\widetilde{\mathsf{U}}^{(i)} || X^{(i)})$, $z := (h + x \cdot c)/w$, and $cert := (c, z)$. The signing key is set as $sk^{\mathsf{SIG}} := (x^{(i)}, X^{(i)}, cert)$.

Then for the ABE part, CC samples $r \xleftarrow{\$} \mathbb{Z}_q$, $D := g_1^{\alpha + r}$, for $\forall j \in \gamma$, samples $r_j \xleftarrow{\$} \mathbb{Z}_q$ and computes $D_j = g_1^r \cdot H_2(j)^{r_j}$, $D'_j := g_1^{r_j}$. The decryption secret key is set as $sk^{\mathsf{ABE}} := (D, \{D_j, D'_j\}_{j \in \gamma})$.

Denote by $sk^{\mathsf{SIG}} \leftarrow Register(msk^{\mathsf{SIG}}, id)$ the registration from a device. CC samples $x^{(i)}$ and computes $X^{(id)} := g_0^{x^{(id)}}$, and then it generates a signature $cert$ on message $\widetilde{\mathsf{U}}^{(id)} || X^{(id)}$ as follows: $w \xleftarrow{\$} \mathbb{Z}_q$, $c := f(g_0^w)$, $h := H_1(\widetilde{\mathsf{U}}^{(id)} || X^{(id)})$, $z := (h + x \cdot c)/w$, and $cert := (c, z)$. The signing key is set as $sk^{\mathsf{SIG}} := (x^{(id)}, X^{(id)}, cert)$.

C. Sign

Before sending a message out, a user, an MEC, or a detector with identity id will sign it via algorithm $Sign(\cdot)$. For user $\mathsf{U}^{(i)}$, id denotes its pseudonym $\widetilde{\mathsf{U}}^{(i)}$, and for a MEC or RSU, id denotes its real identity. $Sign(\cdot)$ inputs the singing key sk^{SIG}, message m and identity id, and outputs a signature σ. After that, the user/device sends (m, σ) out.

The detailed algorithm $Sign(\cdot)$ works as follows.

$\sigma \leftarrow Sign(sk^{\mathsf{SIG}}, m, id)$: The user/device parses $sk^{\mathsf{SIG}} = (x^{(id)}, X^{(id)}, cert)$. Then it samples $\hat{w} \xleftarrow{\$} \mathbb{Z}_q$, $\hat{c} := f(g_0^{\hat{w}})$, $\hat{h} := H_1(m)$, $\hat{z} := (\hat{h} + x^{(id)}\hat{c})/w$, and $\hat{\sigma} := (\hat{c}, \hat{z})$. Finally, it outputs $\sigma := (X^{(id)}, cert, \hat{\sigma})$.

D. Verify

After receiving a message-signature pair (m, σ), the user/device invokes $Verify(\cdot)$ to check the validity of the message. $Verify(\cdot)$ inputs the verification key vk^{SIG}, message m, the identity id of the "claimed" sender and a signature σ, and outputs a bit, where 1 indicates that the message is valid and 0 otherwise. Here id may be a pseudonym $\widetilde{\mathsf{U}}^{(i)}$, or a real identity of MEC/RSU.

The detailed algorithm $Verify$ works as follows.

$0/1 \leftarrow Verify(vk^{\mathsf{SIG}}, m, id, \sigma)$: The user/device parses $\sigma = (X^{(id)}, (c, z), (\hat{c}, \hat{z}))$, and $vk^{\mathsf{SIG}} = X$. If $f((g_0^{H_1(id || X^{(id)})} \cdot X^c)^{1/z}) = c$ and $f((g_0^{H_1(m)} \cdot (X^{(id)})^{\hat{c}})^{1/\hat{z}}) = \hat{c}$, then it outputs 1; otherwise it outputs 0.

E. Encrypt

After generating a/an message/instruction, MEC encrypts it via $Encrypt(\cdot)$ and then broadcasts the ciphertext. The algorithm $Encrypt(\cdot)$ inputs the public key pk^{ABE}, a message msg and a proper access structure $\mathbb{A}$, and outputs a ciphertext ct.

An access structure $\mathbb{A}$ is presented as a tree with root node rt. Each non-leaf node δ in the tree is a threshold gate (k_δ / num_δ), where num_δ is the number of its child nodes, and k_δ is a threshold value s.t. $0 < k_\delta \leq num_x$. For example, $k_\delta = num_\delta$ denotes an AND gate, and $k_\delta = 1$ denotes an OR gate. All child nodes are indexed from 1 to num_δ, and we use function $\mathsf{index}(\cdot)$ to present the corresponding index number for a specific child node. Each leaf node δ is related to an attribute $\mathsf{att}(\delta)$, and we define $k_\delta = 1$ in this case.

[2]One can also set $\mathbb{G}_0$ (used in the signature part) the same as $\mathbb{G}_1$ or $\mathbb{G}_2$ (used in the ABE part). However, it may decrease the efficiency of signature to some extent, since the exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$ is more expensive that in $\mathbb{G}_0$ w.r.t. the same security level.

[3]The hash function H_2 maps an attribute j to a group element $h_j \in \mathbb{G}_1$. One can also replace H_2 by publishing h_j for all j in the attribute universe as the public parameter. However, it requires the size of attribute universe to be bounded in advance, and results in a pp of large size.

[4]Without loss of generality, we assume no collision happens on pseudonyms.

We use $\mathsf{parent}(\delta)$ to denote the parent node of δ. In $Encrypt(\cdot)$, each non-root node δ establishes a polynomial q_δ of degree $k_\delta - 1$, with constraint that $q_\delta(0) = q_{\mathsf{parent}(\delta)}(\mathsf{index}(\delta))$. Therefore, the polynomial of δ (hence the value $q_\delta(0)$) can be reconstructed from any k_δ constant coefficient values of its child nodes, using Lagrange polynomial interpolation.

Let γ be an attribute set and $\mathbb{A}$ be an access tree with root rt. For a leaf node δ, we say δ is satisfied if $\mathsf{att}(\delta) \in \gamma$, i.e., γ contains the attribute of δ. For a non-leaf node δ (including the root node), we say δ is satisfied if at least k_δ child nodes of δ are satisfied. In the scenario of ABE in this paper, we say the attribute set γ matches the access structure $\mathbb{A}$, if there is a subset of γ that has the root node rt satisfied. We refer Figure 9 for a toy example.

The detailed algorithm $Encrypt(\cdot)$ works as follows.

$\mathsf{ct} \leftarrow Encrypt(pk^{\mathsf{ABE}}, \mathsf{msg}, \mathbb{A})$: Let access structure $\mathbb{A}$ be a tree with root rt. MEC first samples $s \xleftarrow{\$} \mathbb{Z}_q$ and sets $q_{\mathsf{rt}}(0) := s$, and then it completes the polynomial q_{rt} by sampling other coefficients at random. Started from rt to the leaf layer, for every child node δ, it constructs polynomial q_δ by choosing random coefficients with constraint that $q_\delta(0) = q_{\mathsf{parent}(\delta)}(\mathsf{index}(x))$. Let Y be the set of leaf nodes in $\mathbb{A}$. For $\forall y \in Y$, it sets $C_y := g_1^{q_y(0)}$, $C'_y := H_2(\mathsf{att}(y))^{q_y(0)}$. At last, it outputs the ciphertext $\mathsf{ct} := (C := g_1^s, \widetilde{C} = Enc(H_3(e(g_1,g_1)^{\alpha s}), \mathsf{msg}), \{C_y, C'_y\}_{y\in Y})$, where the plain message msg is masked via symmetric encryption (e.g., AES) with $H_3(e(g_1,g_1)^{\alpha s})$ as the symmetric key.

F. Decrypt

After receiving a broadcasted ciphertext from MEC, a user invokes $Decrypt(\cdot)$ to decrypt, according to its attributes. $Decrypt(\cdot)$ inputs the secret key sk^{ABE} and the ciphertext ct, and outputs message msg if the attributes of the user satisfies the access structure contained in ct, or outputs a failure symbol $\perp$ otherwise.

The detailed algorithm $Decrypt(\cdot)$ works as follows.

$\mathsf{msg}/\perp \leftarrow Decrypt(sk^{\mathsf{ABE}}, \mathsf{ct})$: The decryption is a bottom-to-up algorithm. Let γ be the corresponding attribute set w.r.t. sk^{ABE}, and y is a leaf node in $\mathbb{A}$. If $j = \mathsf{att}(y) \in \gamma$,

$$\frac{e(D_j, C_y)}{e(D'_j, C'_y)} = \frac{e(g_1^r \cdot H_2(j)^{r_j}, g_1^{q_y(0)})}{e(g_1^{r_j}, H_2(j)^{q_y(0)})} = e(g_1,g_1)^{rq_y(0)}.$$

If γ contains a subset S of the attributes that satisfy $\mathbb{A}$ and y' is a parent node of some leaf nodes in Y, the information of $\{e(g_1,g_1)^{rq_y(0)}\}_{y\in S}$ can be used to obtain $e(g_1,g_1)^{rq_{y'}(0)}$ using Lagrange polynomial interpolation. Via recursion, one can recover $e(g_1,g_1)^{r\cdot q_{\mathsf{rt}}(0)}$ for the root node rt, which is exactly $e(g_1,g_1)^{r\cdot s}$. Therefore, $e(g_1,g_1)^{\alpha\cdot s} = \frac{e(C,D)}{e(g_1,g_1)^{r\cdot s}} = \frac{e(g_1,g_1)^{s\cdot(\alpha+r)}}{e(g_1,g_1)^{r\cdot s}}$ is obtained and the decryption finally returns $\mathsf{msg} \leftarrow Dec(H_3(e(g_1,g_1)^{\alpha s}), \widetilde{C})$, where $Dec(\cdot)$ is the decryption algorithm for the symmetric encryption scheme.

Remark 1: Our System can be further extended to support adaptive revocations for users. For the signature part, a native approach is to maintain a revocation list by CC. Each time an MEC verifies a message-signature pair for a certain user $\mathsf{U}^{(i)}$, it queries CC to check whether $id^{(i)}$ is currently in the revocation list. However, such method would make the verification an interactive algorithm and cost much time in querying CC, hence inefficient. Another approach is to add a time stamp in users' certificates, i.e., $cert$ now is a signature for $\widetilde{\mathsf{U}}^{(i)}||X^{(i)}||t^{(i)}$, where $t^{(i)}$ is a label indicating the expiration point of sk^{SIG}.

The same method can be used for the ABE part. That is, $\mathsf{U}^{(i)}$'s attribute set γ now contains a special attribute called "expiration", which is a number within $\log N$ bits. Also, a special sub-tree indicating current time is added into the access policy. Only when the current time is smaller than the expiration can γ satisfy the access policy. Note that such comparison functionality can be expressed as a $\log N$-deep sub-tree efficiently using AND and OR gates, see [11].

V. Analysis of CVICA

In this section we analyze the security and efficiency of CVICA, and compare it with other schemes.

A. Security

We first analyze the security of CVICA. Namely, it achieves anonymity, authenticity, traceability, and attribute-based confidentiality, as claimed in Section III.

1) Anonymity: Users' data are sent to MEC in the name of pseudoname $\widetilde{\mathsf{U}}$, and the correlationship between U and $\widetilde{\mathsf{U}}$ is exposed to CC only (in the registration phase). Therefore, users' anonymity is guaranteed from MEC and other eavesdroppers in CVICA.

2) Authenticity: Authenticity requires that only registered users and detectors can upload traffic data to MEC, preventing malicious users from uploading fake data to disturb the system. In CVICA, whenever receiving a message, MEC will first check the validity by verifying the signature under the claimed identity of the sender. Therefore, a malicious user cannot upload a traffic data to MEC, unless it forges a valid signature for its fake data.

CVICA takes a two-layer signature architecture with GenDSA signature scheme [36] as the underlying building blocks, and the security of GenDSA is based on the hardness of the discrete logarithm problem. Therefore, the authenticity of CVICA is guaranteed. We refer Appendix C for a formal security proof.

3) Traceability: As shown in Introduction and Section III, users' anonymity is guaranteed against MEC and eavesdroppers in CVICA, but not CC. Each user needs to apply a registration for CC using its real identity, and CC maintains a list storing all identities of users and their pseudonames. That is, once an MEC detects that a certain user $\widetilde{\mathsf{U}}$ uploads malicious data to disturb the CVIS system (of course it should pass the verification algorithm of SIG first), MEC can ask CC to dispose the real identity U under $\widetilde{\mathsf{U}}$ and make a punishment (e.g., a fine). As a result, the traceability of our system achieves.

4) Attribute-based Confidentiality: Attribute-based confidentiality requires that only users with proper attributes can decrypt a ciphertext and read the message. In CVICA, the plain message msg is masked by symmetric encryption under key $H_3(e(g_1, g_1)^{\alpha s})$. However, if the attribute set does not satisfy the access policy, then for any adversary, it cannot obtain enough information to compute $e(g_1, g_1)^{\alpha s}$ and hence the decrypted message. As a result, the attribute-based confidentiality of CVICA is guaranteed. We refer Appendix D for a formal security proof.

B. Efficiency

The space complexity is shown in Table II. Here $|\mathbb{G}_i|$ ($i \in \{0, 1, 2\}$) and $|\mathbb{Z}_q|$ denote the bit-length of an element in $\mathbb{G}_i$ and $\mathbb{Z}_q$, respectively. $|\gamma|$ denotes the number of attributes hold by a user, $|Y|$ denotes the number of leaf nodes in the access structure, and $|\mathsf{msg}|$ denotes the bit-length of the message to be encrypted. The actual sizes of $|\mathbb{G}_i|$ and $|\mathbb{Z}_q|$ depend on the security parameter 1^λ. The lager 1^λ is, the larger the sizes of $|\mathbb{G}_i|$ and $|\mathbb{Z}_q|$ are, and simultaneously the safer CVICA becomes. For example, if we take 256-bit elliptic curve secp256k1 [38] for signatures, then $|\mathbb{Z}_q| = 256$ and $|\mathbb{G}_0| \approx 257$ (after compression).

The time complexity is shown in Table III. Here d denotes the depth of tree access $\mathbb{A}$, and w.l.o.g., we assume the width of $\mathbb{A}$ (maximum number of child nodes per non-leaf node) is a constant. $\mathsf{exp}_i (i \in \{0, 1, 2\})$ and pair denote the time cost per exponentiation in $\mathbb{G}_i$ and per pairing operation, respectively. Similarly, the actual time of $\mathsf{exp}_i (i \in \{0, 1, 2\})$ and pair depend on the security parameter (see also the experiment results in Section VI). We omit the time of hash functions and symmetric encryption since it is modest compared to exp_i and pair.

Remark 2: The space cost of σ and time cost of $Verify$ can be reduced to $2|\mathbb{Z}_q|$ and $2\mathsf{exp}_0$ respectively, if MEC records a list of valid identity-certificate pairs. In this case, a trusted user can upload traffic data with a shorter signature $\hat{\sigma}$. In $Verify(\cdot)$, MEC retrieves the user's certificate $cert = (X^{(id)}, (c, z))$ from the list and checks the validity of $\hat{\sigma}$.

C. Comparison

We compare CVICA with other cryptography schemes used in CVIS from seven perspectives, i.e., authenticity, authentication pattern, anonymity (privacy), traceability, revocation, access control, and storage requirement, see Table IV. Here "authentication pattern" means whether the sender needs to interact with the receiver in the authentication process (obviously non-interactive pattern is superior to interactive pattern), and "revocation" means whether the system supports adaptive revocation for users.

From Table IV we can see, our CVICA meets almost all security properties like privacy and access control, and has a smaller (constant) storage cost. One shortcoming of CVICA is that it does not support adaptive revocation. However, all existing schemes with active revocation have a more expensive storage cost that is linear in the the number of devices. We leave it as an open problem as to construct CVICA with adaptive revocation and a constant storage requirement.

VI. Experiment Results

The experimental efficiency of our system is demonstrated in this section. As for the ABE part, we adapt the cpabe toolkit developed by Bethencourt et al. [11], and the symmetric encryption is implemented with AES in CBC model. For the signature part, standard ECDSA is used. The experiments are run in Ubuntu 20.04, 64-bit, core i5-8250 CPU. More details about the parameters and experiment results are shown as follows.

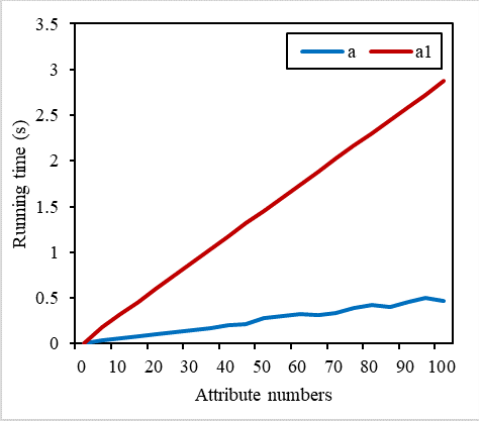

Fig. 4: Running time for Register (KeyGen).

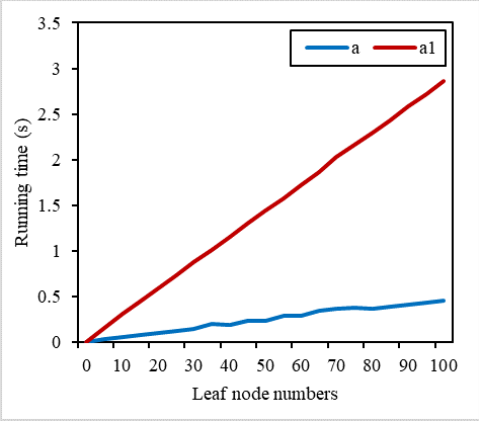

Fig. 5: Running time for ABE encryption.

The parameters "a" and "a1" denote two different recommended pairing types (with base field sizes 512 bits and 1024 bits, respectively) in [41]. As we can see from Figures 4, 5, and 6, the running time of $Register$ (KeyGen) and $Encrypt$ are expected to be linear in the number of attributes or leaf nodes. The decryption time is slightly more difficult to analyze because it depends on the specific attribute access in application. We encrypt a message under randomly generated access tree s.t. the number of leaf nodes is fixed. Then, we decrypt it using one particular key randomly selected from all secret keys satisfying the access tree. Experiment results show that, except for some fluctuations, the running time is still linear in the number of leaf nodes.

TABLE II: Space complexity of CVICA.

pk^{SIG}	sk^{SIG}	σ	msk^{ABE}	pk^{ABE}	sk^{ABE}	ct
$\lvert\mathbb{G}_0\rvert$	$3\lvert\mathbb{Z}_q\rvert$	$4\lvert\mathbb{Z}_q\rvert + \lvert\mathbb{G}_0\rvert$	$\lvert\mathbb{Z}_q\rvert$	$\lvert\mathbb{G}_2\rvert$	$(2\lvert\gamma\rvert+1)\lvert\mathbb{G}_1\rvert$	$(2\lvert Y\rvert+1)\lvert\mathbb{G}_1\rvert + \lvert\mathsf{msg}\rvert$

TABLE III: Time complexity of CVICA.

$Register(User)$	$Regisger(Divice)$	$Sign$	$Verify$	$Encrypt$	$Decrypt$
$2\mathsf{exp}_0$ $+(2\lvert\gamma\rvert+1)\mathsf{exp}_1$	$2\mathsf{exp}_0$	exp_0	$4\mathsf{exp}_0$ or $2\mathsf{exp}_0$	$(2\lvert Y\rvert+1)\mathsf{exp}_1$ $+\mathsf{exp}_2$	$(2\lvert Y\rvert+1)\mathsf{pair}$ $+O(d)\mathsf{exp}_2$

TABLE IV: Comparison among other schemes.

Scheme	Authenticity	Authentication Pattern	Privacy: Anonymity	Traceability	Revocation	Access Control	Storage Requirement w.r.t. number of devices
SXSS06 [22]	✓	interactive	✓	×	✓	×	linear
AMOEBA [25]	✓	non-interactive	✓	✓	×	×	linear
ECPP [24]	✓	interactive	✓	✓	✓	×	linear
SPECS [21]	✓	interactive	✓	✓	✓	×	linear
SZZF10 [28]	✓	interactive	✓	✓	✓	×	linear
PACP [20]	✓	non-interactive	✓	×	×	×	constant
WDG10 [39]	✓	non-interactive	✓	✓	✓	×	linear
XBQR10 [23]	✓	non-interactive	✓	✓	✓	×	linear
SCMS [27], [40]	✓	interactive	✓	✓	✓	×	linear
UABS18 [31]	✓	non-interactive	✓	×	×	✓	constant
NOTSA [4]	✓	interactive	×	×	×	×	constant
FKKB21 [30]	✓	interactive	✓	×	×	×	constant
Ours: CVICA	✓	non-interactive	✓	✓	×	✓	constant

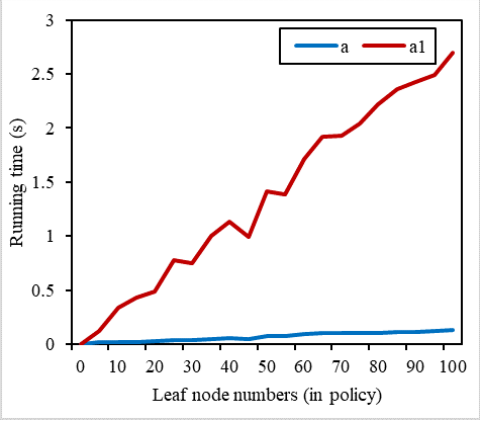

Fig. 6: Running time for ABE decryption.

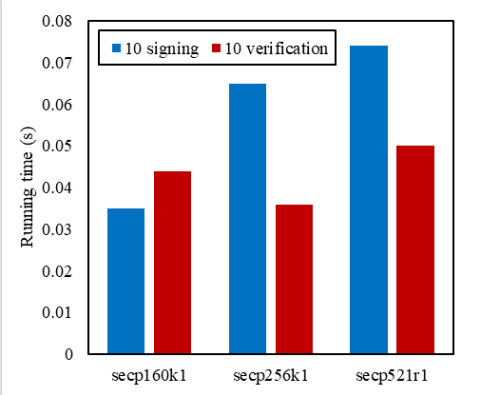

Fig. 7: Running time for 10 signing and 10 verification.

We use openssl cryptography toolkit [42] to test the performance of ECDSA and AES-CBC. They are both lighter building blocks compared with ABE and show high efficiency in practice. The detailed results are shown in Figure 7 and 8.

VII. Conclusion

This study proposes CVICA, a coordinated vehicle infrastructure cryptography architecture to deal with the confidentiality and authenticity issues in CVIS. Concretely, attribute-based encryption, ABE, is used to send confidential data to target users with the attributes that satisfying the access policy. Anonymous identity is also used to ensure privacy protection. Message authenticity is ensured at the same time by using identity-based signature and double-layer mechanism. Traceability is also guaranteed to trace malicious data. Algorithms concerning privacy preservation, message authenticity, and flexible access control are proposed. Security proofs and efficiency analyses are provided to prove practicability and security of CVICA. It is also compared with existing security schemes in terms of cryptographic properties. Besides, the efficiency of the proposed CVICA is tested, and results show that CVICA has high efficiency, high practicability, and low latency.

As we can see, a trusted authority (CC) is required in CVICA. However, totally trusting the authority is too ideal and risky, making the system unreliable. In later works, we will consider designing a distributed system, to enhance the

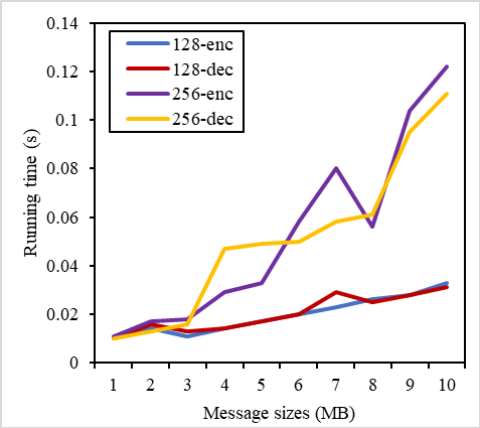

Fig. 8: Running time for AES encryption and decryption.

reliability of CVICA. Another further direction is to support adaptive revocation in out system, without sacrificing much efficiency and practicability.

REFERENCES

[1] S. I. Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," Transportation Research Part C: Emerging Technologies, vol. 46, pp. 121–131, Sep. 2014.

[2] M. Amoozadeh, A. Raghuramu, C.-n. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," IEEE Communications Magazine, vol. 53, no. 6, pp. 126–132, Jun. 2015, conference Name: IEEE Communications Magazine.

[3] D. Hahn, A. Munir, and V. Behzadan, "Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges," IEEE Intelligent Transportation Systems Magazine, vol. 13, no. 1, pp. 181–196, 2021, conference Name: IEEE Intelligent Transportation Systems Magazine.

[4] L. Wang and X. Liu, "NOTSA: Novel OBU With Three-Level Security Architecture for Internet of Vehicles," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3548–3558, Oct. 2018, conference Name: IEEE Internet of Things Journal.

[5] A. Sadiq, N. Javaid, O. Samuel, A. Khalid, N. Haider, and M. Imran, "Efficient Data Trading and Storage in Internet of Vehicles using Consortium Blockchain," in 2020 International Wireless Communications and Mobile Computing (IWCMC), Jun. 2020, pp. 2143–2148, iSSN: 2376-6506.

[6] A. Shamir, "Identity-based cryptosystems and signature schemes," in CRYPTO 1984, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds., vol. 196. Springer, 1984, pp. 47–53. [Online]. Available: https://doi.org/10.1007/3-540-39568-7_5

[7] "Itu-t x.509 information technology – open systems interconnection – the directory: Public-key and attribute certificate frameworks," 2019.

[8] J. M. d. Fuentes, L. González-Manzano, J. Serna-Olvera, and F. Veseli, "Assessment of attribute-based credentials for privacy-preserving road traffic services in smart cities," Personal and Ubiquitous Computing, vol. 21, no. 5, pp. 869–891, Oct. 2017. [Online]. Available: https://doi.org/10.1007/s00779-017-1057-6

[9] A. Oracevic, S. Dilek, and S. Ozdemir, "Security in internet of things: A survey," in 2017 International Symposium on Networks, Computers and Communications (ISNCC), May 2017, pp. 1–6.

[10] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in EUROCRYPT 2005, ser. Lecture Notes in Computer Science, R. Cramer, Ed., vol. 3494. Springer, 2005, pp. 457–473. [Online]. Available: https://doi.org/10.1007/11426639_27

[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in S&P 2007. IEEE Computer Society, 2007, pp. 321–334. [Online]. Available: https://doi.org/10.1109/SP.2007.11

[12] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in CRYPTO 2001, ser. Lecture Notes in Computer Science, J. Kilian, Ed., vol. 2139. Springer, 2001, pp. 213–229. [Online]. Available: https://doi.org/10.1007/3-540-44647-8_13

[13] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," IEEE Commun. Surv. Tutorials, vol. 17, no. 1, pp. 228–255, 2015. [Online]. Available: https://doi.org/10.1109/COMST.2014.2345420

[14] H. Krishnan and A. Weimerskirch, "Verify-on-demand - a practical and scalable approach for broadcast authentication in vehicle-to-vehicle communication," 2011.

[15] C. F. Kerry and C. R. Director, "Fips pub 186-4 federal information processing standards publication digital signature standard (dss)," 2013.

[16] "Public key cryptographic algorithm sm2 based on elliptic curves part 2: Digital signature algorithm," 2010. [Online]. Available: http://www.gmbz.org.cn/upload/2018-07-24/1532401673138056311.pdf

[17] V. Dolmatov and A. Degtyarev, "GOST R 34.10-2012: Digital signature algorithm," RFC, vol. 7091, pp. 1–21, 2013. [Online]. Available: https://doi.org/10.17487/RFC7091

[18] E. Kiltz and G. Neven, "Identity-based signatures," in Identity-Based Cryptography, ser. Cryptology and Information Security Series, M. Joye and G. Neven, Eds. IOS Press, 2009, vol. 2, pp. 31–44. [Online]. Available: https://doi.org/10.3233/978-1-58603-947-9-31

[19] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," J. Cryptol., vol. 22, no. 1, pp. 1–61, 2009. [Online]. Available: https://doi.org/10.1007/s00145-008-9028-8

[20] D. Huang, S. Misra, M. Verma, and G. Xue, "PACP: an efficient pseudonymous authentication-based conditional privacy protocol for vanets," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 3, pp. 736–746, 2011. [Online]. Available: https://doi.org/10.1109/TITS.2011.2156790

[21] T. W. Chim, S. Yiu, L. C. K. Hui, and V. O. K. Li, "SPECS: secure and privacy enhancing communications schemes for vanets," Ad Hoc Networks, vol. 9, no. 2, pp. 189–203, 2011. [Online]. Available: https://doi.org/10.1016/j.adhoc.2010.05.005

[22] K. Sha, Y. Xi, W. Shi, L. Schwiebert, and T. Zhang, "Adaptive privacy-preserving authentication in vehicular networks," in 2006 First International Conference on Communications and Networking in China, 2006, pp. 1–8.

[23] H. Xiong, K. Beznosov, Z. Qin, and M. Ripeanu, "Efficient and spontaneous privacy-preserving protocol for secure vehicular communication," in ICC 2010. IEEE, 2010, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICC.2010.5502673

[24] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," in INFOCOM 2008. IEEE, 2008, pp. 1229–1237. [Online]. Available: https://doi.org/10.1109/INFOCOM.2008.179

[25] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: robust location privacy scheme for VANET," IEEE J. Sel. Areas Commun., vol. 25, no. 8, pp. 1569–1589, 2007. [Online]. Available: https://doi.org/10.1109/JSAC.2007.071007

[26] A. Studer, E. Shi, F. Bai, and A. Perrig, "Tacking together efficient authentication, revocation, and privacy in vanets," in SECON 2009. IEEE, 2009, pp. 1–9. [Online]. Available: https://doi.org/10.1109/SAHCN.2009.5168976

[27] B. Brecht and T. Hehn, "A Security Credential Management System for V2X Communications," in Connected Vehicles: Intelligent Transportation Systems, ser. Wireless Networks, R. Miucic, Ed. Cham: Springer International Publishing, 2019, pp. 83–115. [Online]. Available: https://doi.org/10.1007/978-3-319-94785-3_4

[28] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," IEEE Trans. Parallel Distributed Syst., vol. 21, no. 9, pp. 1227–1239, 2010. [Online]. Available: https://doi.org/10.1109/TPDS.2010.14

[29] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: http://doi.acm.org/10.1145/359168.359176

[30] H. Farran, D. J. Khoury, E. F. Kfoury, and L. Bokor, "A blockchain-based V2X communication system," in TSP 2021. IEEE, 2021, pp. 208–213. [Online]. Available: https://doi.org/10.1109/TSP52935.2021.9522599

[31] D. A. Ulybyshev, A. O. Alsalem, B. K. Bhargava, S. Savvides, G. Mani, and L. B. Othmane, "Secure data communication in autonomous V2X systems," in ICIOT 2018. IEEE Computer Society, 2018, pp. 156–163. [Online]. Available: https://doi.org/10.1109/ICIOT.2018.00029

[32] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: protecting confidentiality with encrypted query processing," in SOSP 2011, T. Wobber and P. Druschel, Eds. ACM, 2011, pp. 85–100. [Online]. Available: https://doi.org/10.1145/2043556.2043566

[33] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted

data," in CCS 2006, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds. ACM, 2006, pp. 89–98. [Online]. Available: https://doi.org/10.1145/1180405.1180418
[34] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in EUROCRYPT 2011, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer, 2011, pp. 568–588. [Online]. Available: https://doi.org/10.1007/978-3-642-20465-4_31
[35] Y. Michalevsky and M. Joye, "Decentralized policy-hiding ABE with receiver privacy," in ESORICS 2018, ser. Lecture Notes in Computer Science, J. López, J. Zhou, and M. Soriano, Eds., vol. 11099. Springer, 2018, pp. 548–567. [Online]. Available: https://doi.org/10.1007/978-3-319-98989-1_27
[36] M. Fersch, E. Kiltz, and B. Poettering, "On the provable security of (EC)DSA signatures," in ACM SIGSAC 2016, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 1651–1662. [Online]. Available: https://doi.org/10.1145/2976749.2978413
[37] D. Jao, "Elliptic curve cryptography," in Handbook of Information and Communication Security, P. P. Stavroulakis and M. Stamp, Eds. Springer, 2010, pp. 35–57. [Online]. Available: https://doi.org/10.1007/978-3-642-04117-4_3
[38] Ecdsa: Elliptic curve signatures. [Online]. Available: https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages
[39] Q. Wu, J. Domingo-Ferrer, and Ú. González-Nicolás, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," IEEE Trans. Veh. Technol., vol. 59, no. 2, pp. 559–573, 2010. [Online]. Available: https://doi.org/10.1109/TVT.2009.2034669
[40] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, "A security credential management system for V2V communications," in 2013 IEEE Vehicular Networking Conference. IEEE, 2013, pp. 1–8. [Online]. Available: https://doi.org/10.1109/VNC.2013.6737583
[41] B. Lynn. (2022) Pbc: The pairing-based cryptography library. [Online]. Available: https://crypto.stanford.edu/pbc/
[42] (2022) Openssl: Cryptography and ssl/tls toolkit. [Online]. Available: https://www.openssl.org/
[43] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in EUROCRYPT 1996, ser. Lecture Notes in Computer Science, U. M. Maurer, Ed., vol. 1070. Springer, 1996, pp. 387–398. [Online]. Available: https://doi.org/10.1007/3-540-68339-9_33
[44] G. Fuchsbauer, E. Kiltz, and J. Loss, "The algebraic group model and its applications," in CRYPTO 2018, ser. Lecture Notes in Computer Science, H. Shacham and A. Boldyreva, Eds., vol. 10992. Springer, 2018, pp. 33–62. [Online]. Available: https://doi.org/10.1007/978-3-319-96881-0_2
[45] G. Fuchsbauer, A. Plouviez, and Y. Seurin, "Blind schnorr signatures and signed elgamal encryption in the algebraic group model," in EUROCRYPT 2020, ser. Lecture Notes in Computer Science, A. Canteaut and Y. Ishai, Eds., vol. 12106. Springer, 2020, pp. 63–95. [Online]. Available: https://doi.org/10.1007/978-3-030-45724-2_3
[46] V. Shoup, "Lower bounds for discrete logarithms and related problems," in EUROCRYPT 1997, ser. Lecture Notes in Computer Science, W. Fumy, Ed., vol. 1233. Springer, 1997, pp. 256–266. [Online]. Available: https://doi.org/10.1007/3-540-69053-0_18
[47] B. Bauer, G. Fuchsbauer, and A. Plouviez, "The one-more discrete logarithm assumption in the generic group model," in ASIACRYPT 2021, ser. Lecture Notes in Computer Science, M. Tibouchi and H. Wang, Eds., vol. 13093. Springer, 2021, pp. 587–617. [Online]. Available: https://doi.org/10.1007/978-3-030-92068-5_20

APPENDIX

A. Cryptographic Tools

In this section we define the semantics of cryptographic building blocks, and their security requirements.

1) Signatures:

Definition 1 (Signatures): A signature (SIG) scheme SIG consists of the following three algorithms.

- $(vk, sk) \leftarrow Setup(1^\kappa)$: The key generation algorithm takes as input the security parameter κ, and outputs a (public) verification key vk and a (secret) signing key sk.
- $\sigma \leftarrow Sign(sk, m)$: The signing algorithm takes as input sk and a message m, and outputs a signature σ.
- $0/1 \leftarrow Ver(vk, m, \sigma)$: The verification algorithm takes as input vk, m, σ, and outputs a bit $0/1$, where 1 indicates that σ is valid for m and 0 otherwise.

We require SIG to have the following correctness and security.

- **Correctness**. For any $(vk, sk) \leftarrow Setup(1^\kappa)$, any message m and $\sigma \leftarrow Sign(sk, m)$, it holds that $Ver(vk, m, \sigma) = 1$.
- **Security**. SIG is said to have existential unforgeability against chosen message attacks (EUF-CMA), if for any PPT adversary $\mathcal{A}$, it is hard to forge a valid signature σ^* for a new message m^*, after seeing multiple valid message-signature pairs (m, σ) with m chosen adaptively by $\mathcal{A}$.

2) Attribute-Based Encryption:

Definition 2 (Attribute-Based Encryption): A (ciphertext-policy) attrubute-based encryption ((CP)-ABE) scheme ABE consists of the following algorithms.

- $(pk, msk) \leftarrow Setup(1^\kappa)$: The setup algorithm takes as input the security parameter κ, and outputs a public key pk and a master secret key msk of the system.
- $c \leftarrow Encrypt(pk, m, \mathbb{A})$: The encryption algorithm takes as input pk, a message m, and an access structure $\mathbb{A}$, and outputs a ciphertext c.
- $sk_\gamma \leftarrow KeyGen(msk, \gamma)$: The key generation algorithm takes as input msk, a set of attributes γ (for a certain user), and outputs a secret key sk_γ.
- $m/\perp \leftarrow Decrypt(pk, c, sk_\gamma)$: The decryption algorithm takes as input pk, c, and sk_γ, and outputs a message m, or a failure symbol $\perp$.

We require ABE to have the following correctness and security.

- **Correctness**. For any $(pk, msk) \leftarrow Setup(1^\kappa)$, any message m, access structure $\mathbb{A}$ and $c \leftarrow Encrypt(pk, m, \mathbb{A})$, any attribute set γ and $sk_\gamma \leftarrow KeyGen(msk, \gamma)$, it holds that $Decrypt(pk, c, sk_\gamma) = m$ if γ satisfies $\mathbb{A}$.
- **Security**. ABE is said to be secure, if for any PPT adversary $\mathcal{A}$, it is hard to distinguish an encryption of m_0 from an encryption of m_1 under access structure $\mathbb{A}^*$, after seeing multiple secret keys sk_γ with γ chosen adaptively by $\mathcal{A}$ (with restriction that γ does not satisfy $\mathbb{A}^*$). Here $m_0, m_1, \mathbb{A}^*$ are sampled by $\mathcal{A}$ itself.

3) Symmetric Encryption:

Definition 3 (Symmetric Encryption): A symmetric encryption (SE) scheme SE consists of the following algorithms.

- $c \leftarrow Enc(k, m)$: The encryption algorithm takes as input the symmetric key k and the message m, and outputs a ciphertext c.
- $m \leftarrow Dec(k, c)$: The decryption algorithm takes as input k, c, and outputs a message m.

We require SE to have the following correctness and security.

- **Correctness**. For any symmetric key k, any message m and $c \leftarrow Enc(k, m)$, it holds that $Dec(k, c) = m$.
- **Security**. SE is said to have semantic security, if for any PPT adversary $\mathcal{A}$, it is hard to distinguish an encryption

of m_0 from an encryption of m_1 under a random key k. Here m_0 and m_1 are sampled by $\mathcal{A}$ itself.

B. A Toy Example for The Access Structure

A toy example for access structure is shown in Figure 9. Now assume unfortunately there is a traffic accident on the road. CC receives the detected accident information from the detector and broadcasts the information to the MECs within the affecting area. Further, the accident-related sensitive information is broadcasted to RSUs and the vehicles passing by. But the sensitive information (e.g., personal data and driver health status) are encrypted so that only the police vehicle, the ambulance, or the small size rescue vehicle can access to it. A rescue vehicle with big size does not has the corresponding attributes satisfying the first "AND" gate, and hence cannot decrypt the encrypted message.

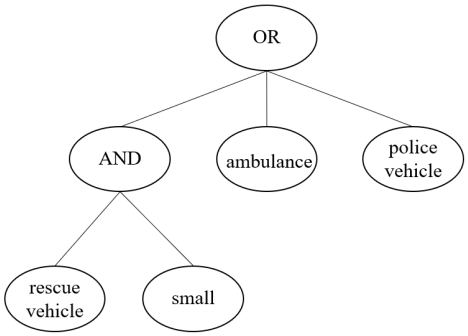

Fig. 9: A toy example for the access policy structure.

C. Proof of Authenticity

Our construction mainly follows the certification paradigm of IBS scheme due to Bellare et al. [19], which is actually a two-layer signature architecture. Namely, let $\widehat{\mathsf{SIG}}$ be a signature scheme with EUF-CMA security. The verification (resp. signing) key of CC servers as the main public (resp. secret) key of the signature part. For each party id, its signing key consists of a key-pair of $\widehat{\mathsf{SIG}}$ independently generated by CC, and a signature for id and the public key, which is served as the certificate. Parties' signature for message m contains its public key, the certificate, and a signature for m.

In our CVICA, we adopt the GenDSA signature scheme [36] as the underlying $\widehat{\mathsf{SIG}}$. Recall that in $\widehat{\mathsf{SIG}}$, the verification key is in the form of $(g_0, X = g_0^x)$. To sign a message m, the signer first generates a commitment $W = g_0^w$ with $w \in \mathbb{Z}_q$ chosen at random. Then it computes $z = (h + x \cdot c)/w$ for $c := f(W)$ and $h := H_1(m)$, and the signature is (c, z).

For the security of $\widehat{\mathsf{SIG}}$, we have the following theorem.

Theorem 1 (EUF-CMA Security of GenDSA): If the discrete logarithm (DL) assumption holds in $\mathbb{G}_0$, H_1 and the middle part of f works as random oracles, $\widehat{\mathsf{SIG}}$ is EUF-CMA secure. More precisely, for any PPT adversary $\mathcal{A}$ against the EUF-CMA security of $\widehat{\mathsf{SIG}}$, there exists an algorithm $\mathcal{B}$ with running time roughly the same as $\mathcal{A}$, and

$$\mathsf{Adv}^{dl}_{\mathcal{B},\mathbb{G}_0}(\kappa) + \frac{Q^2_{hash}}{q} \geq \left(\mathsf{Adv}^{euf\text{-}cma}_{\mathcal{A},\widehat{\mathsf{SIG}}}(\kappa) - \frac{6Q_{hash}Q_s + 2Q^2_{hash}}{q}\right)^2 / 2Q_{hash},$$

where Q_s and Q_{hash} denote the total numbers of signing and hash queries (for both f and H_1), respectively.

According to Theorem 1, if $\mathcal{A}$'s advantage is non-negligible, the discrete logarithm problem is not hard, which conflicts with the DL assumption. Theorem 1 is proved using the programmability of random oracles and the standard forking lemma [43], and we refer readers [36] for the details of proof. It is worth mentioning that the reduction can be made tighter if we are working in the algebraic group model (AGM) [44] [45] or the generic group model (GGM).

Further, we prove the following theorem.

Theorem 2 (Authenticity of CVICA): If $\widehat{\mathsf{SIG}}$ is EUF-CMA secure, our two-layer signature architecture in CVICA satisfies authenticity. More precisely, for any PPT forger $\mathcal{A}$ against the authenticity of CVICA, there exists algorithm $\mathcal{B}$ with running time roughly the same as $\mathcal{A}$, and

$$\mathsf{Adv}^{euf\text{-}cma}_{\mathcal{B},\widehat{\mathsf{SIG}}}(\kappa) \geq \frac{Adv^{auth}_{\mathcal{A},CVICA}(\kappa)}{(\mu_{\mathsf{U}} + \mu_{\mathsf{D}} + \mu_{\mathsf{MEC}} + 1)},$$

where $(\mu_{\mathsf{U}} + \mu_{\mathsf{D}} + \mu_{\mathsf{MEC}})$ is the total number of signing identities in the system.

Recall that a forger $\mathcal{A}$ breaks the authenticity of CVICA, if it generates a valid message-signature pair (m^*, σ^*) under some identity id^*, while party id^* has not signed on m^* at all. To prove Theorem 2, we define the event of breaking authenticity into two sub-events.

- Event Repeat: party id^* had sent its certificate $cert^*$ out before (therefore $\mathcal{A}$ can reuse $cert^*$ in its forge σ^*).
- Event $\overline{\mathsf{Repeat}}$: party id^* had never sent its certificate $cert^*$ out.

We analyze $\overline{\mathsf{Repeat}}$ first. Recall that $\sigma^* = (X^*, cert^*, \hat{\sigma})$, where X^* is a verification key of $\widehat{\mathsf{SIG}}$. In the verification algorithm, the verifier checks whether $cert^*$ is a valid signature for message $id^*||X^*$ under public key X of CC, and whether $\hat{\sigma}$ is a valid signature for message m under public key X^* of the claimed party id^*. If the real party id^* had never signed any message and sent out a signature, its real certificate is totally hidden to the forger $\mathcal{A}$. $\overline{\mathsf{Repeat}}$ means that $\mathcal{A}$ successfully generates a valid signature $cert^*$ for a new message $id^*||X^*$, hence breaking the EUF-CMA security of $\widehat{\mathsf{SIG}}$ under CC's public key X. Therefore, we can construct a reduction algorithm $\mathcal{B}_1$ such that

$$Adv^{euf\text{-}cma}_{\mathcal{B}_1,\widehat{\mathsf{SIG}}}(\kappa) \geq \Pr[\overline{\mathsf{Repeat}}].$$

Then, we analyze Repeat. In this case, the real party id^* had signed some other message m' and sent the signature $\sigma' = (X^*, cert^*, \hat{\sigma}')$ out, thus $\mathcal{A}$ can reuse the public key and certificate of id^*. Namely, it is easy for $\mathcal{A}$ to generate a forge $\sigma^* = (X^*, cert^*, \hat{\sigma})$ and pass the first check formula. However, passing the second verification formula means that

$\mathcal{A}$ successfully generates a valid signature $\hat{\sigma}$ for a new message m^*, hence breaking the EUF-CMA security of $\widehat{\mathsf{SIG}}$ under id^*'s public key X^*. There are $(\mu_{\mathsf{U}} + \mu_{\mathsf{D}} + \mu_{\mathsf{MEC}})$ different choices of id^* in total, and hence we can construct a reduction algorithm B_2 such that

$$Adv^{euf\text{-}cma}_{\mathcal{B}_2,\widehat{\mathsf{SIG}}}(\kappa) \geq \frac{\Pr[\overline{\mathsf{Repeat}}]}{(\mu_{\mathsf{U}} + \mu_{\mathsf{D}} + \mu_{\mathsf{MEC}})}.$$

Theorem 2 holds immediately as a result.

D. Proof of Attribute-based Confidentiality

We prove the attribute-based confidentiality of CVICA in the generic group model (GGM) [46] [47]. Recall that in GGM, the are two encoding functions ψ_1, ψ_2 that map $\mathbb{Z}_q$ into $\{0,1\}^\omega$ with $\omega \geq 3|q|$. One is given oracles ψ_1, ψ_2 to encode an element $x \in \mathbb{Z}_q$ to $\psi_1(x) \in \mathbb{G}_1$ or $\psi_2(x) \in \mathbb{G}_2$, oracles to compute multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$, and an oracle to compute bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$.

Theorem 3: If SE is semantically secure, and H_2, H_3 work as random oracles, our system achieves attribute-based confidentiality in the generic group model. More precisely, for any PPT adversary $\mathcal{A}$, there exists an algorithm $\mathcal{B}$ such that

$$\mathsf{Adv}^{semantic}_{\mathcal{B},\mathsf{SE}}(\kappa) \geq \mathsf{Adv}^{att\text{-}conf}_{\mathcal{A},CVICA} - \frac{Q^2_\psi}{q},$$

where Q_ψ is the total number of coding queries (for both ψ_1 and ψ_2) involved in the experiment.

Let $\mathcal{A}$ be an adversary against attribute-based confidentiality. It aims to read the masked message from ct, but $\mathcal{A}$ does not have an admissible secret key for decryption. Here we formally prove the IND-CPA security for ABE. That is, $\mathcal{A}$ cannot distinguish whether ct is an encryption of msg or ct is an encryption of msg', even if $\mathcal{A}$ can conclude many users and collect their secret keys with restriction that all their attribute sets w.r.t. the secret keys do not satisfy $\mathbb{A}$.

As we can see, the message msg is masked by symmetric encryption under key $H_3(e(g_1, g_1)^{\alpha s})$. Since SE has semantic security, and H_3 is a random oracle, $\mathcal{A}$'s attack advantage is bounded by $\mathsf{Adv}^{se}(\kappa)$ if it does not compute $e(g_1, g_1)^{\alpha s}$. Next, we prove that in GGM, given the challenge ciphertext ct and all secret keys $\{sk^{(\mathsf{ABE})(i)}\}_{i\in T}$ for some user set T, the probability that $\mathcal{A}$ obtains $e(g_1, g_1)^{\alpha s}$ (i.e., the encoding $\psi_2(\alpha s)$) is at most Q^2_ψ/q.

Let $j \in \mathcal{U}$ be an attribute. On input j, $H_2(\cdot)$ samples a random t_j and returns $g_1^{t_j}$. For a leaf node y_j, we use λ_j to denote $q_{y_j}(0)$. Therefore, all information $\mathcal{A}$ has obtained includes

- from public key: $e(g_1, g_1)^\alpha$;
- from hash queries: $\{g_1^{t_j}\}_{j\in\mathcal{U}}$;
- from ciphertext: $g_1^s, \{g_1^{\lambda_j}, g_1^{t_j\lambda_j}\}_{j\in Y}$;
- from secret keys: $\{g_1^{\alpha+r^{(i)}}, \{g_1^{r^{(i)}+t_j\cdot r_j^{(i)}}, g_1^{r_j^{(i)}}\}_{j\in\gamma^{(i)}}\}_{i\in T}$.

Since all exponents in $\mathbb{Z}_q$ are chosen independently at random, with probability $1 - Q^2_\psi/q$, there exists no collision on neither $\mathbb{G}_1$ nor $\mathbb{G}_2$.

For simplicity, in the analysis below, we use $a \in \mathbb{Z}_q$ to denote the information $g_1^a \in \mathbb{G}_1$ or $g_1^a \in \mathbb{G}_2$ that $\mathcal{A}$ obtains. Based on the obtained information, $\mathcal{A}$ has access to the following types of queries in $\mathbb{G}_2$, see Table V.

From Table V we can see, the only way $\mathcal{A}$ can obtain a term containing αs is by multiplying s with $\alpha + r^{(i)}$ to get $\alpha s + sr^{(i)}$ for $i \in T$ (here T denotes the colluded set controlled by $\mathcal{A}$). Therefore, $\mathcal{A}$ needs to create some linear combination of existing terms in the table to eliminate the term $\sum_{i\in T} u^{(i)} \cdot sr^{(i)}$.

Recall that $s = q_{\mathsf{rt}}(0)$ in the access tree with root node rt, and s can be reconstructed from $\lambda_{j'}$ for some set $T^{(i)}$. By searching all items in Table V, we find that the only way $\mathcal{A}$ can concrete the term $\sum_{i\in T} u^{(i)} \cdot sr^{(i)}$ is multiplying $(r^i + t_j r_j^{(i)})$ with $\lambda_{j'}$. Now consider the polynomial

$$\sum_{i\in T}\left(\sum_{(j,j')\in T^{(i)}} u^{(i)}_{j,j'} \cdot \left(\lambda_{j'} r^{(i)} + t_j \lambda_{j'} r_j^{(i)}\right)\right).$$

We analysis the formula in two cases.

Case 1. For some $i \in T$, the secret s cannot be reconstructed from $\lambda_{j'}$ s.t. $\lambda_{j'} \in T^{(i)}$ (this means that for party i, its attribute set cannot satisfy the access control $\mathbb{A}$).

In this case, the term $\sum_{i\in T} u^{(i)} \cdot sr^{(i)}$ cannot be canceled. Hence $\mathcal{A}$ cannot compute αs.

Case 2. For all $i \in T$, the secret s can be reconstructed from $\lambda_{j'}$ (this means that $\mathcal{A}$ tries to reconstruct s from Lagrange polynomial interpolation, even it has no admissible secret key).

In this case, $\mathcal{A}$ needs to eliminate the extra term $t_j\lambda_{j'} r_j^{(i)}$ for $i \in \gamma^{(i)}$ (the attribute set of party i). However, according to Table V, there exists no term of the form $t_j\lambda_{j'} r_j^{(i)}$, which indicates that $\mathcal{A}$ cannot cancel it. As a result, αs is totally hidden from $\mathcal{A}$.

Theorem 3 holds immediately from the analysis above.

TABLE V: All types of information on $\mathbb{G}_2$ obtained by $\mathcal{A}$.

	s	t_j	λ_j	$t_j\lambda_j$	$r_j^{(i)}$	$r^{(i)}+t_jr_j^{(i)}$	$\alpha+r^{(i)}$	
s	$s\cdot s$	st_j	$s\lambda_j$	$s\cdot t_j\lambda_j$	$sr_j^{(i)}$	$sr^{(i)}+st_jr_j^{(i)}$	$\alpha s+sr^{(i)}$	
$t_{j'}$		$t_jt_{j'}$	$t_{j'}\lambda_j$	$t_jt_{j'}\lambda_j$	$t_{j'}r_j^{(i)}$	$t_{j'}r^i+t_jt_{j'}r_j^{(i)}$	$\alpha t_{j'}+t_{j'}r^{(i)}$	
$\lambda_{j'}$			$\lambda_j\lambda_{j'}$	$t_j\lambda_j\lambda_{j'}$	$\lambda_{j'}r_j^{(i)}$	$\lambda_{j'}r^{(i)}+t_j\lambda_{j'}r_j^{(i)}$	$\alpha\lambda_{j'}+\lambda_{j'}r^{(i)}$	
$t_{j'}\lambda_{j'}$				$t_jt_{j'}\lambda_j\lambda_{j'}$	$t_{j'}\lambda_{j'}r_j^{(i)}$	$t_{j'}\lambda_{j'}r^{(i)}+t_jt_{j'}\lambda_{j'}r_j^{(i)}$	$\alpha t_{j'}\lambda_{j'}+t_{j'}\lambda_{j'}r^{(i)}$	
$r_{j'}^{(i')}$					$r_j^{(i)}r_{j'}^{(i')}$	$r^{(i)}r_{j'}^{(i')}+t_jr_j^{(i)}r_{j'}^{(i')}$	$\alpha r_{j'}^{(i')}+r^{(i)}r_{j'}^{(i')}$	
$r^{(i')}+t_{j'}r_{j'}^{(i')}$						$(r^{(i)}+t_jr_j^{(i)})(r^{(i')}+t_{j'}r_{j'}^{(i')})$	$(\alpha+r^{(i)})(r^{(i')}+t_{j'}r_{j'}^{(i')})$	
$\alpha+r^{(i')}$							$(\alpha+r^{(i)})(\alpha+r^{(i')})$	
								α