Attribute-Based Access Control for Vehicular Edge Cloud Computing

Cheng-Yu Cheng*, Hang Liu*, Li-Tse Hsieh*, Edward Colbert*, Jin-Hee Cho‡
*Dept. of Electrical Engineering & Computer Science
The Catholic University of America
Washington, DC, USA
{chengc, liuh, 84hsieh, colberte}@cua.edu
‡Dept. of Computer Science
Virginia Polytechnic Institute and State University
Falls Church, VA, USA
jicho@vt.edu

***Abstract*—Edge computing provides a new platform to support real-time connected vehicle applications. One of the challenges is to ensure communication security among vehicles, roadside devices, and edge servers, while addressing group communication nature and fast changing group membership. This paper presents the design and evaluation of a secure communication scheme for vehicular edge computing applications based on decentralized attribute-based encryption (ABE), which provides flexible data encryption and access control according to attribute-based policies, without requirement for knowing the recipient identities and establishment of a secure communication channel between the sender and recipient. Key-policy ABE and ciphertext-policy ABE are employed to collect sensed data for process and disseminate processed results to the involved entities, respectively. The communication protocols are designed and an experimental prototype considering an edge cloud-assisted collision warning application is implemented for proof of the concept. The evaluation results show that the proposed ABE-based scheme is efficient and feasible.**

***Index Terms*—Attribute Based Encryption, Edge Computing, Connected Vehicle**

I. Introduction

Over the past few years, advances in sensing, information and communication technologies have enabled many new applications for connected vehicles and intelligent transportation, such as collision avoidance, cooperative perception, advanced driver-assistance systems (ADAS), and others. Connected and fully autonomous vehicles have become closer to being a reality [1][2]. More recently, edge computing has been proposed as a new platform to support real-time connected vehicle and autonomous driving applications [3][4][5]. Edge servers with abundant computing and storage resources can be co-located or integrated with base stations (BSs), gateways, and roadside units (RSU) in vehicle networks to perform intensive data processing tasks at the edge of the network to improve vehicular services. Vehicles can also join the edge cloud and contribute their unused resources to assist in workload processing. For example, edge servers can collect sensor data from nearby autonomous vehicles (AVs) and roadside infrastructure, detect the objects that may be in the blind spots of the sensors on an AV, and then send warning messages to the AV to avoid the collision. Compared to traditional cloud services, edge computing can significantly reduce data transfer time, conserve communication bandwidth, and provide local context-aware services required by these real-time vehicular applications.

Security for vehicular edge computing is critical because false information may cause the safety of life. The communications among vehicles, roadside infrastructure, edge servers are often group oriented, e.g. collecting sensed information from and sending warning messages to a group of cars in an area. However, traditional security mechanisms are mainly user-based, employing asymmetric or symmetric cryptographic mechanisms. Such a user-based approach is neither flexible nor scalable in group establishment and management, and cannot support diverse secure group communications well. For example, with asymmetric public/private key schemes, if a data source wants to send encrypted data to a group, it will need to get the public key for each of the entities in the group, and encrypt and send the messages to each of the group members individually, which will cost a lot of communication and computation resources. With symmetric group key schemes, the group membership should be managed and the group key should be distributed to all the group members, which is challenging and incurs extra latency, especially for vehicular networks in which the group members frequently change, some group members leave and new members join due to high mobility of vehicles. Furthermore, there may be multiple communications groups with different sets of attributes at the same time. In addition, the communications between vehicles and edge servers are delay-sensitive. An efficient and scalable security scheme is needed for real-time vehicular edge applications.

Attribute-based encryption (ABE) [6][7] has been proposed to realize flexible data encryption and access control based on attribute policies, instead of users. It enables secure information dissemination to the entities having certain attributes, with no requirement for prior knowledge of who will receive the data and no need to establish the secure channel between the sender and recipient. Encrypted messages can

978-1-7281-8266-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IEEECloudSummit48914.2020.00009

be decrypted only when the attributes/policies associated with the recipient's secret key matches the policies/attributes of the ciphertext. ABE can seamlessly support communications of multiple groups with different attributes or levels of trust. Such a cryptographic technique provides a promising data access control mechanism for vehicular edge computing applications in terms of flexibility and scalability. For example, an edge server may send a notification to a group of vehicles with attributes "autonomous driving capability" and "subscribed to a particular service", or collect the information from a group of "vehicle with LiDAR" or "vehicle with stereo camera".

In this paper, we design, build, and evaluate an ABE-based secure communication scheme for edge computing. Our design is driven by the requirements of deploying and accelerating this new class of vehicular edge computing applications, e.g. securely collecting and processing data generated by vehicles and roadside infrastructure sensors in real time. Specifically, we designed a scheme that employs two types of ABE methods, key-policy ABE and ciphertext-policy ABE, to collect sensed data for process and send processed results to the involved entities, respectively. This enables flexible data encryption and access control to meet diverse group communication requirements. We have implemented an example edge application, i.e. collision warning system on our testbed and have shown how to secure data communications with attribute-based policies. Note that we only use this application as an example to drive the discussion and evaluation. Our scheme can easily support other secure edge computing applications. We have evaluated the performance of the proposed ABE-based scheme on the testbed and demonstrated that it is a viable solution.

II. System Architecture

In this section, we introduce the system architecture of our ABE-based secure communication scheme for vehicular edge computing, considering an edge cloud-assisted collision warning service [8] as an application example to prove the concept. Edge servers obtain sensor information, such as camera and LiDAR data, from various vehicles and roadside infrastructure as well as the trajectory of the vehicles at an intersection or in a high-risk area. They aggregate and analyze sensor information obtained to detect obstacles that may obstruct the current path of any vehicle. Information about the detected obstacles is conveyed to the vehicles. The edge servers have access to a pool of sensor data from multiple vehicles and roadside sensors, including the information that may be in blind spots or out of the range of the onboard sensors on a vehicle. The edge cloud-assisted collision warning service can help autonomous vehicles (AVs) and vehicles with advanced driver-assistance systems (ADAS) to make intelligent decisions that foresee and avoid obstacles not directly sensed by the onboard sensors. Further, since the edge servers have longer range information about the traffic patterns and other unexpected events such as roadwork or accidents, they can help AVs to plan more efficient paths. In summary, the edge cloud-assisted collision warning can be an add-on service and the vehicles can subscribe to this service to improve driving safety and efficiency, especially in high-risk areas.

For such vehicular edge computing applications, the privacy of vehicles' passengers as well as the subjects in the camera images should be protected and only the authorized edge servers can access and process the sensor data. Further, only the vehicles with certain connected autonomous driving or ADAS capabilities, and subscription to this collision warning service or its partner services can access to the warning messages. ABE is used to protect data transmission and control data access in our system design. We focus on data confidentiality and access control in this paper. Attribute based signature [9] can be employed to ensure data authenticity and integrity regarding to the claimed attributes in the messages, which is part of our future work.

In general, there are two types of ABEs, key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). With KP-ABE, the plaintext is encrypted to the ciphertext that is labeled with a set of attributes with a public key. The private secret key of an entity is associated with an access policy. A private key can decrypt a particular ciphertext only if there is a match between the attributes of the ciphertext and the policy of the private key. For CP-ABE, a source encrypts the data with the public keys associated with a set of attributes and also includes an access policy. The ciphertext can be decrypted by a recipient or a group of recipients if and only if the attributes or credentials ascribed to the private keys satisfy the policy of the ciphertext. In our system design, vehicles and roadside sensors employ KP-ABE for protection of their sensed information because it is more suitable for them to encrypt and transmit the sensed data that contains the data attributes to the edge servers. For example, a road camera sends its video with the attributes of its location of region and road camera. Only an edge server with a private key associated with the policy matching the data attributes. i.e. that "particular location of region" and "camera video process permission", can decrypt and process the data. Another reason to employ KP-ABE for data collection from the vehicles and roadside sensors is that the computation complexity of KP-ABE encryption is less than that of CP-ABE encryption.

On the other hand, we employ decentralized CP-ABE [10] to provide fine-grained role-based access control to the messages generated by an edge server because CP-ABE allows that a data source, i.e. an edge server to specify the access policy to the data as a flexible Boolean formula structure over a set of attributes when encrypting the data. The access policy determines the attributes that an entity should have in order to decrypt the messages. For example, an edge server may broadcast a collision warning message with the information of detected obstacles, and the messages are encrypted by CP-ABE with a policy that only vehicles with "autonomous driving" or "ADAS" capability and with subscription to "edge cloud-assisted collision warning service" or "one of its partner services" and in "service region" can decrypt the messages.

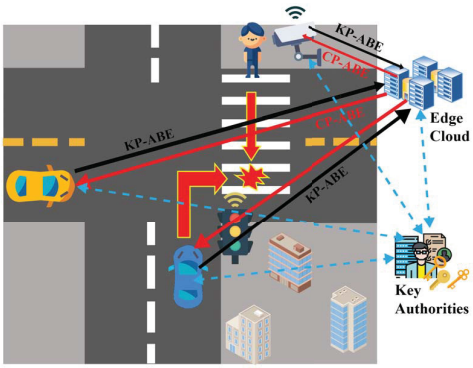


Fig. 1: ABE-based security scheme for vehicular edge computing.

In addition, distributed multiple authorities, i.e. multiple attribute key generation centers in the system, further improve flexibility and scalability.

Fig. 1 shows the system architecture under consideration. There are four types of entities. We briefly describe each of them below.

1) Key Authorities (KAs): KAs are responsible for generating and distributing the private secret keys and publishing the public keys. A KP-ABE KA issues a private key to an user that associated with an access structure of the attributes to control which ciphertexts the user is able to decrypt, and publishes the corresponding public keys. For decentralized CP-ABE, there may be multiple KAs, and each KA is responsible for one or more attributes, issuing private attribute key components and publishing the corresponding public key components. These KAs are not required to have online coordination and may not even know each other.

2) Autonomous Vehicles (AVs): AVs are the primary entities served by the system. They send the sensed information to the servers for process using KP-ABE and receive the processed results and warning messages encrypted by CP-ABE from the edge servers.

3) Roadside Infrastructure: Roadside cameras and other sensors send their sensed data such as video to the edge servers using KP-ABE. Certain roadside devices, e.g. smart traffic lights and traffic control robots can receive the messages and take action based on the commands from the edge servers.

4) Edge servers: Edge servers can decrypt the data from various vehicles and roadside sensors using its private KP-ABE key. They process the data and send the results/warnings/commands to the involved entities with CP-ABE.

III. Secure Communication Scheme Design

In this section, we discuss how we use KP-ABE and decentralized CP-ABE to protect the data generated by vehicles and roadside sensors as well as the processed results and warning messages generated by the edge servers. We mainly focus on the communication protocol design and related algorithms for key distribution, secure information collection and dissemination. More details of theoretical preliminaries for CP-ABE and KP-ABE cryptography used in this paper can be found in [6, 7, 10].

The system is first set up and initialized. For KP-ABE, select a bilinear group $\mathbb{G}1$ of prime order p and a generator g of $\mathbb{G}1$ [7]. Let $e : \mathbb{G}1 \times \mathbb{G}1 \rightarrow \mathbb{G}2$ denote the bilinear map. $\mathcal{U} = \{1, 2, \ldots, n\}$ represents a set of attributes. For each attribute $i \in \mathcal{U}$, KP-ABE KA randomly chooses a number t_i from $\mathbb{Z}_p$. In addition, it chooses a random number y in $\mathbb{Z}_p$. The KP-ABE KA then generates and publishes the public key parameters (PK), $T_1 = g^{t_1}, ..., T_{|\mathcal{U}|} = g^{t_{|\mathcal{U}|}}, Y = e(g, g)^y$. It keeps $t_1, ..., t_{|\mathcal{U}|}, y$ as the master secret key (MSK). For decentralized CP-ABE, select a bilinear group G of order $N = p_1p_2p_3$ and a generator g_1 of G_{p_1}. N and g_1 are the global public parameters (GP) for CP-ABE. In addition, define a Hash function $H : \{0, 1\}^* \rightarrow G$ that maps global identities GIDs of users to elements of G. Global parameters GP $= N, g_1$ and H are published for decentralized CP-ABE KA setup. For an attribute i that belongs to one of CP-ABE KAs, the KA chooses two random numbers $\alpha_i, y_i \in \mathbb{Z}_N$. It keeps $\text{MSK}_i = \{\alpha_i, y_i \forall i\}$ as its master secret attribute key and publishes $\text{PK}_i = \{e(g_1, g_1)^{\alpha_i}, g_1^{y_i} \forall i\}$ as its corresponding public attribute key for attribute i. In addition, we assume a user such as an autonomous vehicle, roadside infrastructure device, or edge server registers with its home authority and obtains a global identifier, GID, a digital certificate binding its GID and its identity public key, as well as an identity private key for authentication with KAs to obtain the secret attribute keys.

After the system initialization and KA setup, a user can request for the KP-ABE and CP-ABE attribute keys to perform message encryption and decryption. Fig. 2 shows the protocol for CP-ABE secret key request and distribution. To request a secret CP-ABE key for an attribute i, a vehicle will send its GID to the responsible CP-ABE KA, and authenticate and establish a secure communication channel with the KA using its digital certificate and a standard SSL/TLS protocol [11]. After the verification of the requester's identity and attribute, the KA will generate the corresponding attribute key using Algorithm 1, which will be bound to the requester's GID to prevent collusion attacks. The secret attribute key will be sent to the requester through the SSL/TLS secure connection. An edge server will get the public keys for the attributes from the responsible CP-ABE KAs in order to encrypt the messages.

Note that a vehicle has two kinds of attributes, static and dynamic attributes. For the static attributes such as model, year, and autonomous driving or ADAS capability, it can get the corresponding secret attribute keys from the manufacture. For the dynamic attributes such as subscription to a service or in a service region, it can request for the corresponding secret attribute keys from responsible key authorities, e.g. the secret key for the collision warning service from a service provider, and the secret attribute key in a service region from a regional authority when it drives into the region and registers with the regional KA. A service region could be a large area,

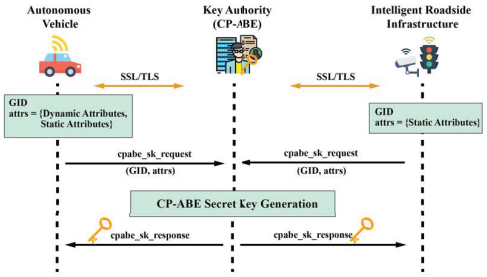


Fig. 2: CP-ABE key distribution.

Algorithm 1 CP-ABE Key Generation

INPUT: GID, $\{i\}$ (the attributes requested by entity) CPABE_MSK, GP.
OUTPUT: CPABE_SK

1: GID Verification
2: Static and dynamic attributes authentication
3: **for** Each attribute i requested by the entity **do**
4: Calculate the key for attribute i.
5: $K_{i,GID} = g_1^{\alpha_i} H(GID)^{y_i}$ ▷ CPABE_SK
6: **end for**

e.g. a metropolitan area, serviced by a local collision warning service provider, and it may contain many edge servers, e.g. one at an intersection. With ABE, the vehicle only needs to be authenticated once and obtain the secret attribute key for the metropolitan region. It does not have to perform mutual authentication and establish a secure connection with every edge server at each intersection it passes through, which reduces latency and improve scalability.

Fig. 3 illustrates the protocol for KP-ABE key request and distribution. An edge server will authenticate and establish a secure connection with the KP-ABE KA via SSL/TLS and then request the secret KP-ABE key. The KA will take the master secret key MSK and an access policy T to generate a secret key SK = $\{D_x\}$ using Algorithm 2. An access policy expressed by a Boolean formula is transformed into an access tree structure T for KP-ABE SK generation [7]. A leaf node of the tree represents an attribute, and the function $att(x)$ denotes the attribute associated with the leaf node x. Each non-leaf node of the tree represents a threshold gate. Let num_x denote the number of children of a node x and k_x be its threshold value, then $0 < k_x \leq num_x$. The threshold gate is an OR gate if k_x = 1 and an AND gate if k_x = num_x. The SK embeds the policy to decrypt a ciphertext encrypted with a set of attribute public keys. The SK is sent to the edge server through the SSL/TLS connection. An vehicles and roadside sensor obtains the corresponding public attribute keys from the KP-ABE KA to encrypt the sensed data.

Fig. 4 shows the secure message exchanges between AVs and edge servers as well as between roadside infrastructure and edge servers using ABE. AVs and roadside sensors encrypt their sensed information, such as camera and LiDAR

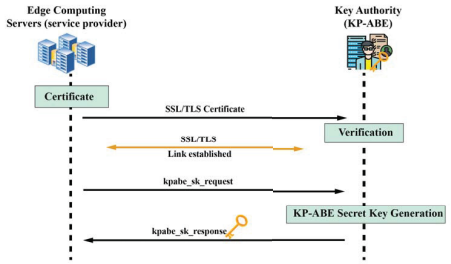


Fig. 3: KP-ABE key distribution.

Algorithm 2 KP-ABE Key Generation

INPUT: T (Tree structure access policy), KPABE_MSK
OUTPUT: KPABE_SK

1: GID Verification
2: **for** Each node x in the tree T **do**
3: Choose a polynomial q_x
4: Set degree $d_x = k_x - 1$ ▷ k_x: threshold value
5: Set $q_r(0) = y$ ▷ r: root node
6: Set $q_x(0) = q_{parent(x)}(index(x))$
7: Give:
8: $D_x = g^{\frac{q_x(0)}{t_i}}$ where $i = att(x)$ ▷ KPABE_SK
9: **end for**

data with KP-ABE using Algorithm 3 and send the ciphertext (CT) to the edge servers. AVs also send their position, velocity, and planned path with KP-ABE to the edge servers. The ciphertext contains a set of attributes that they have such as "collision warning service data" and "service region". An edge server with the access tree structure embedded in its private key matching the data attribute set can successfully decrypt the ciphertext using Algorithm 4 and obtain the original data in plaintext.

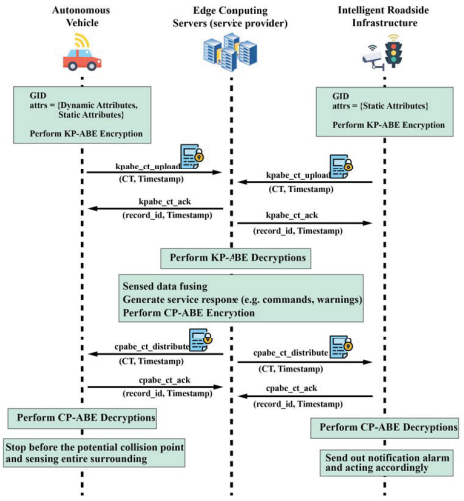


Fig. 4: Secure message exchange with KP-ABE and CP-ABE.

Algorithm 3 KP-ABE Encryption

INPUT: $M \in \mathbb{G}_2$ (plaintext sensed data), γ (a set of attributes), KPABE_PK (public key)

Choose a random value $s \in \mathbb{Z}_p$
Compute the ciphertext as:
$E = (\gamma, E^{'} = MY^s, \{E_i = T_i^s\}_{i \in \gamma})$

Algorithm 4 KP-ABE Decryption

INPUT: E (KP-ABE Ciphertext), D (KPABE_SK)

Define a recursive function KPABE_Dec(E, D, x)
Take ciphertext $E = (\gamma, E^{'}, \{E_i\}_{i \in \gamma})$ as input
Take KPABE_SK D as input
for Each node x in tree T and x is leaf node **do**
    Let $i = att(x)$
    KPABE_Dec(E, D, x) =
$\begin{cases} e(D_x, E_i) = e(g^{\frac{q_x(0)}{t_i}}, g^{s \cdot t_i}) = e(g,g)^{s \cdot q_x(0)} \text{ if } i \in \gamma \\ \perp \text{ otherwise} \end{cases}$
end for
for Each node x in tree T and x is not leaf node **do**
    for Each node z in tree T that is a child of x **do**
        KPABE_Dec(E, D, x) and store output as F_z
        Let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$
        Compute F_x = $e(g,g)^{s \cdot q_x(0)}$
    end for
end for
Call the function on the root node r:
KPABE_Dec(E, D, r) = $e(g,g)^{ys}$ = Y^s
Since $E^{'} = MY^s$, divided by Y^s and recover M

The edge server then fuses the sensor data, detects the objects, and computes the potential collision points based on the current trajectory of all the vehicles. The warning messages containing the information about a list of detected objects, the object type (pedestrian, car, etc.), the location (latitude and longitude), and the velocity of each object, as well as the potential collision points are generated and conveyed to the involved AVs, which will assist AVs to take action to avoid collision in advance and to plan safer and more efficient paths. The edge server takes a plaintext message M, an access policy matrix A with its rows mapped to attributes that is converted from a Boolean formula or a tree structure [10], and the attribute public keys as input to encrypt M using CP-ABE encryption Algorithm 5 and it transmits the produced ciphertext. Only the AVs with the secret attribute keys satisfying the access policy of the ciphertext can successfully decrypt the ciphertext and access to the messages. For example, a vehicle with secret keys for attributes {"ADAS", "edge cloud-assisted collision warning service", "in service region"} will be able to decrypt the messages with an access policy {("autonomous driving" OR "ADAS") AND ("edge cloud-assisted collision warning service" OR "one of its partner services") AND "in service region"}. The decryption procedure at an AV is detailed in Algorithm 6 which takes the ciphertext, secret attribute keys, and system global public parameters as the input to obtain the plaintext. As discussed before, there is no need to establish a secure channel between AVs/roadside sensors and edge servers for data message exchanges because the transmitted data are already encrypted with KP-ABE and CP-ABE, which is one of the advantages for using ABE.

Algorithm 5 CP-ABE Encryption

INPUT: plaintext message M, Boolean formula T, global parameter GP and the public keys for the corresponding attributes CPABE_PK.

A is an n x ℓ access matrix with ρ mapping its rows to attributes
Choose a random $s \in \mathbb{Z}_N$
Let λ_x denote $A_x \cdot v$ where A_x is row x of A
Choose a random vector $w \in \mathbb{Z}_N^{\ell}$
Let w_x denote $A_x \cdot w$
for For each row A_x in A **do**
    Choose a random $r_x \in \mathbb{Z}_N$
    Computes the ciphertext CT as:
    $C_0 = M e(g_1, g_1)^s$,
    $C_{1,x} = e(g_1, g_1)^{\lambda_x} e(g_1, g_1)^{\alpha_{\rho(x)} r_x}$,
    $C_{2,x} = g_1^{r_x}$
    $C_{3,x} = g_1^{y_{\rho(x)} r_x} g_1^{w_x} \forall x$
end for

Algorithm 6 CP-ABE Decryption

INPUT: ciphertext CT, CPABE_SK, GID, global parameters GP.

We assume the CT is encrypted under access matrix (A, ρ)
Compute H(GID)
if decryptor has secret key $\{\text{CPABE\_SK}_{\rho(x),\text{GID}}\}$ and for a subset of rows A_x of A such that (1, 0, ..., 0) is in this row **then**
    for Each such x **do**
$\frac{C_{1,x} \cdot e(H(\text{GID}), C_{3,x})}{e(CPABE\_SK_{\rho(x),\text{GID}}, C_{2,x})}$
$= e(g_1, g_1)^{\lambda_x} e(H(\text{GID}), g_1)^{w_x}$
    end for
end if
Choose constants $c_x \in \mathbb{Z}_N$ such that
$\sum_x c_x A_x = (1, 0, ..., 0)$
Computes:
$\prod_x (e(g_1, g_1)^{\lambda_x} e(H(\text{GID}), g_1)^{w_x})^{c_x} = e(g_1, g_1)^s$
(recall that $\lambda_x = A_x \cdot v$ and $w_x = A_x \cdot w$, where $v \cdot (1, 0, ..., 0) = s$ and $w \cdot (1, 0, ..., 0) = 0$)
Plaintext message can be obtained as:
$M = \frac{C_0}{e(g_1, g_1)^s}$

IV. Implementation and Experiments

We have implemented a prototype system with the ABE-based secure communication scheme designed in Sections II and III on our edge computing testbed to prove the concept. In this section, we present our implementation and evaluation effort.

A. Prototype Implementation

We developed three applications using Python and OpenABE Library [12] for the above collision warning application scenario, including key authorities, vehicles/roadside infrastructure devices, and edge servers. OpenABE library is an open source cryptographic library that incorporates a variety of ABE algorithms and an application programming interface (API) implemented in C/C++. We used Python to develop the protocols for authentication, key distribution, ABE encrypted message exchanges among vehicles/roadside infrastructure devices, key authorities, and edge servers. These applications run in containers on the Linux based VMs.

The KA application performs system initialization and KA setup as well as key generation and distribution. It can be configured to be a KP-ABE and/or CP-ABE. With decentralized CP-ABE, there may be multiple KAs and each KA handles its own set of attributes and issues corresponding keys. These authorities only need a set of initial common public parameters at the setup phase, and they do not communicate each other. A separate KA application can handle KP-ABE. When an KA application is started, it will first set up the global public parameters and generate the master keys for CP-ABE and/or KP-ABE. Then the application will wait for the key requests.

The device application was developed for autonomous vehicles, roadside infrastructure sensors/actuators and other intelligent devices that require ABE. As the application starts, it establishes secure communication connections with the KA applications using SSL/TLS, and sends CP-ABE secret key requests for one or more attributes to the responsible KAs to obtain the attribute secret keys. It also obtains the public keys for its KP-ABE attributes from the KP-ABE KA application. The device application includes KP-ABE encryption and CP-ABE decryption functions. The encryption function uses KP-ABE to encrypt the plaintext with the attributes and public key. After encryption, the ciphertext is transmitted to the edge server. The decryption function allows to decrypt CP-ABE ciphertext received from the edge server.

The edge server application provides the functionalities of KP-ABE ciphertext decryption and CP-ABE plaintext encryption. When the edge server application is started, it obtains its KP-ABE secret key from the KP-ABE KA. SSL/TLS is also used for the authentication and secure communications between the edge server and the KA. In addition, It gets the CP-ABE public attribute keys from the responsible CP-ABE KAs. The edge server application then start listening to the messages from any device client application. When the server receives an encrypted message, it will take the ciphertext and KP-ABE secret key as input to decrypt the ciphertext into plaintext message. The edge server further processes the received information and sends the responses. It also has an function interface for CP-ABE encryption which encrypts the plaintext message with CP-ABE public key and an access structure into the ciphertext. The ciphertext messages are sent to the devices.

B. Experiments and Results

We conducted the evaluation of our ABE-based information collection and dissemination scheme on the testbed. We provide the experiment results in terms of encryption/decryption time, key generation time, and end-to-end service response time under various numbers of attributes and sizes of messages. The vehicle, roadside sensor, edge server, and KA applications run in Linux based VM containers on the hosts with an Intel core i5-8600K 3.6 GHz 6-core processor and 32 Gbyte memory. The hosts are connected through a 802.11n wireless LAN.

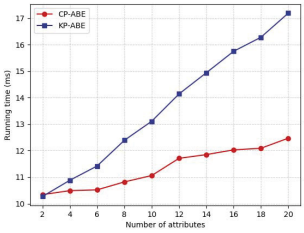


Fig. 5: ABE secret key generation time vs. the number of attributes.

Figure 5 shows the CP-ABE and KP-ABE secret key generation time versus the number of attributes. The KAs generate the secret keys once receiving a request. As we expected, the key generation time increases with the number of attributes. In addition, the secret key generation time of KP-ABE is much longer than that of CP-ABE because KA needs to construct the access policy structure associated with the KP-ABE secret key. On the other hand, KA only need to tag the attributes associated with the secret key in the CP-ABE key generation process.

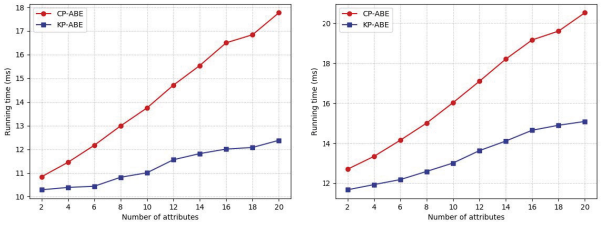
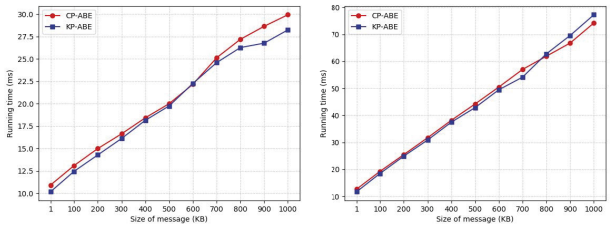


Fig. 6: (a) ABE encryption time and (b) ABE decryption time vs. the number of attributes.

Figure 6a and 6b show the encryption and decryption time versus the number of attributes involved for CP-ABE and

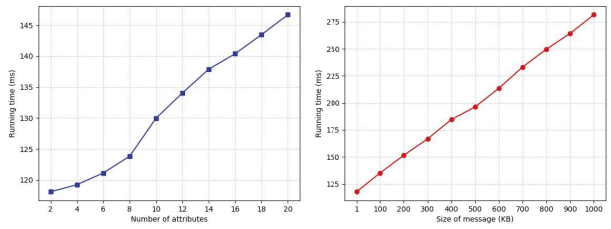
KP-ABE, respectively. The message size is 1 KB in the experiments. The encryption and decryption time increases with the number of attributes required. CP-ABE takes more time in encryption and decryption than KP-ABE under the same computing power, especially as the number of attributes is large. In our system design, KP-ABE is used for the vehicles and roadside sensors to send the sensed data to the edge servers. CP-ABE is used for the edge servers to send the process results and collision warning messages. The number of sensed data messages will be much more than the number of process result and collision warning messages. In this sense, our design is efficient. Note that more attributes in the CP-ABE and KP-ABE provide stronger security.



(a) (b)

Fig. 7: (a) ABE encryption time and (b) ABE decryption time vs. the message size.

Figure 7a and 7b illustrate the encryption and decryption time versus the size of messages, respectively, for CP-ABE and KP-ABE, when the number of attributes is fixed to two attributes in the experiments. The time for encryption and decryption linearly increases with the size of messages.



(a) (b)

Fig. 8: End-to-end service response time vs. (a) the number of attributes and (b) the message size.

We also analyze the overall end-to-end service response time for the ABE-based data collection and edge computing using our prototype. Figure 8a and 8b show the service response time versus the number of attributes and the size of messages, respectively. The end-to-end service response time includes the KP-ABE encryption time by the road entities, the ciphertext transmission time to the edge server, the KP-ABE decryption and CP-ABE encryption time on the edge server, the encrypted warning message transmission time to the road entities, and the CP-ABE decryption time at the road entities. The total service response time is less than 300 ms for an 1 Mbyte message. From the above experiments, the proposed ABE-based scheme is efficient and incurs reasonable overhead in terms of the computational cost and service response time for real-time vehicular edge computing applications.

V. Conclusions

In this paper, we develop a secure information collection and dissemination scheme based on KP-ABE and CP-ABE, which aims to support edge computing for real-time vehicular applications such as edge cloud-assisted collision warning service. The communication protocols are designed for key management, ABE-based secure data collection, and processing result dissemination. The proposed ABE-based security scheme is validated through a prototype for the proof of the concept. The evaluation results show that the proposed scheme is efficient and incur reasonable overhead. Moving forward, we will develop and evaluate the attributed-based scheme to ensure the information authenticity and integrity for vehicular edge computing applications.

References

[1] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," Phil. Trans. Roy. Soc. vol. A368, no. 1928, pp. 4649-4672, 2010.

[2] S. Kim, W. Liu, M. H. Ang, E. Frazzoli and D. Rus, "The Impact of Cooperative Perception on Decision Making and Planning of Autonomous Vehicles," IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 3, pp. 39-50, 2015.

[3] K. Sasaki, N. Suzuki, S. Makido and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Tsukuba, 2016.

[4] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," IEEE Communications Magazine, vol. 55, no. 7, pp. 94-100, July 2017.

[5] K. Zhang, Y. Mao, S. Leng, Y. He and Y. ZHANG, "Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading," in IEEE Vehicular Technology Magazine, vol. 12, no. 2, pp. 36-44, June 2017.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," In proceedings of the 13th ACM conference on Computer and Communi- cations Security, 2006.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," In Proceedings of 2007 IEEE Symposium on Security and Privacy (SP '07), 2007.

[8] S. Kumar, S. Gollakota, and D. Katabi. "A cloud-assisted design for autonomous driving." In Proceedings of the First Workshop on Mobile Cloud Computing, 2012.

[9] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in Topics in Cryptology-CT-RSA 2011, Lecture Notes in Comput. Sci., vol. 6558, pp. 376–392, 2011

[10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," Advances in Cryptology – EUROCRYPT 2011, Lecture Notes in Computer Science, vol. 6632, 2011.

[11] William Stallings, Network Security Essentials: Applications and Standards, 5th Ed., Prentice Hall, 2014.

[12] OpenABE Library, https://github.com/zeutro/openabe.