Privacy-preserving Data Sharing with Attribute-based Private Matching Based on Edge Computation in the Internet-of-Things

Ruei-Hau Hsu*, Yu-Hsaing Hu† , Guan-Wei Lin‡ and Bing-Cheng Ko§
, National Sun Yat-sen University, Kaohsiung, Taiwan
rhhsu@mail.cse.nsysu.edu.tw*, m073040029@student.nsysu.edu.tw†,
b043040022@student.nsysu.edu.tw‡, m063040031@student.nsysu.edu.tw§

***Abstract*—The data sharing is essential for the analytics of specific tasks in internet-of-things (IoT). The availability and the latency of data exchange affect the validity of critical and real-time IoT services. Thus, a new computing model, i.e., edge computing model, is urgently required for data sharing in IoT. However, data sharing based on edge computing model needs to address additional security issues, i.e., the privacy protection of data acquisition and transmission against honest-but-curious edge devices. Thus, this work proposes a privacy-preserving data sharing with attribute-based private matching based on edge computation in IoT. In the proposed scheme, IoT users/devices can acquire/distribute data based on attribute-based private matching on honest-but-curious edge devices without exposing attribute/policy information and exchanged data. Moreover, the proposed scheme guarantees anonymous IoT user/device authentication with the support of handover between the regions covered by two adjacent edge devices. Nonetheless, the data transmission is of secure end-to-end communications between IoT users and devices to reduce the consumption of bandwidth between IoT users/devices and edge devices. Finally, this work implements the system to evaluate the performance and provides the security analysis of the proposed security system.**

I. Introduction

With the development of network technology and hardware devices, IoT impacts the service types of information and operation technologies. IBM [1] predicts that the number of IoT devices will exceed 25 billion by 2020 and will exceed 100 billion by 2050. IoT also has much research in various aspects and application methods [2], [3], [4]. For example, how to secure data sharing between IoT devices in smart healthcare, smart factories, or internet of vehicle (IoV) has been a challenge due to the complexity of key management and the restriction of resource-limited IoT devices for security operations.

The rise of the IoT is also driving another service, "sensing as a service" [5], which uses sensors from mobile devices to collect data. With the popularization of diverse types of IoT devices, IoT devices can collect sensing data by the equipped sensors and provide services, such as air quality in a place, noise measurement, parking space status, washing machine information, and takeaway information, based on the analytics of the collected data.

However, such services bring privacy issues. First, regarding the leakage of privacy data [6], [7], [8], [9], whether it is a requester or a service provider, an attacker can infer some private information from the requested data or collected data, such as health status, activities, eating habits, living location, etc.

Second, regarding the resource limitation [10], [11], [12], due to the resource limitation, it is difficult for IoT devices to calculate complicated cryptographic algorithms, such as ABE, homomorphic encryption, group signatures, etc., to protect data privacy or integrity.

In traditional cloud computing, IoT devices generate large amounts of data and upload them to a central server for processing, thereby reducing the computing cost of the device, but in the face of millions of IoT devices, it will bring considerable overhead of communication bandwidth. Excessive network overhead will fail some real-time services of 5G, such as intelligent transportation [13], smart healthcare [14]), etc., due to the high communication latency between IoT devices/users and cloud platforms. Moreover, when massive amount of data sending from huge number of IoT devices, cloud platforms will not be able to service all the requests since the computational resources are exhausted.

Due to the above reason, this work proposes an IoT system moodel based on edge computing to mitigate the heavy consumption of communication bandwidth and computational resources caused by the massive among of data upload/access to a single central could platform. Moreover, edge devices are closer to user ends; the transmission latency can be reduced accordingly.

A. Application Scenario

Taking internet of vehicles (IoV) as an example, every vehicle contains a large number of sensors to collect data from the component of vehicles. If the collected data needs to be sent to the cloud server for analyses, the transmission delay is inevitable. Therefore, the cloud can not provide low latency service and it is a fatal problem in the rapidly changing of roadside conditions and the high mobility of vehicles in IoV application scenarios.

With the assistance of edge device, the driver behaviour does not need to be directly uploaded to a remote cloud server, but transmit it to the adjacent roadside Unit (RSU) for real-time

processes and analyses. The RSU needs to make a decision for the near vehicles according to the real-time traffic conditions, which can provide real-time feedback for vehicles. Moreover, when the RSU receives certain error reports, they can also be sent back to the cloud server for backup or more advanced analyses.

B. Contributions

In order to solve the problems mentioned above, this paper proposed a privacy-preserving device searching for secure data sharing based on edge computing in IoT. The utilization of distributed edge devices in IoT regions can reduce the inevitable latency by the long-distance transmission of data from/to cloud platforms and alleviate the overhead of bandwidth. The contributions of this work are shown as follows.

1) This work proposes a security system for privacy-preserving device searching (PPDS) to help IoT users to retrieve the required data without exposing their identities, and the searching conditions and device attributes.
2) IoT users/devices and edge devices can authenticate to each other anonymously. If IoT users/devices enter to the same group of the edge device, the system provides fast authentication to reduce the transmission cost of contacting the central security server.
3) When the IoT users/devices located in the regions covered by different edge devices, the data can still be shared among them due to the support of a handover mechanism of the mobility of IoT users/devices .
4) The outsourcing of security computation can reduce the computing cost of IoT resource-constrained devices for the computation tasks of advanced cryptographic algorithms.

C. Organization

The remainder of this paper is organized as follows. Section II presents the related works regarding the privacy-preserving matching problem and authentication in IoT. Section III introduces the system construction of IoT networks based on edge computing. Section IV presents the preliminaries required by the proposed scheme. Section V introduces the detailed proposed scheme. The security analysis and performance evaluation to the proposed scheme are presented in section VI and section VII, respectively. Finally, we conclude this paper in Section VIII.

II. Related Works

Privacy protection has being a long-term issue for data sharing in IoT. One of that is the matching problem. Matching users' requirements is to guarantee the effectiveness of IoT services. Several protocols solving the matching problem with privacy-preserving have been proposed in recent research.

Ni et al. [6] proposed a matrix method to deal with the matching problem based on the property of matrix orthogonal. According to the property, the mismatching request will return a zero matrix. The drawback of this method is the flexibility. It can only take one attribute as a matching condition; however, we may need to match several requirements at the same time.

Attribute-based encryption (ABE) is a modern cryptographic algorithm for secure and fine-grained access control. ABE can realize private matching by embedded the condition of matching as a policy in the ciphertext and the values to be matched as attributes embedded in user secret keys. If the conditions and values match, the ciphertext of the specified policy can be decrypted by the user secret keys of the attributes satisfying the policy. However, the drawback of ABE is the computation costs. As the number of attributes increasing, the computation costs can be an issue in practice due to the high complexity of ABE.

Due to the problems of ABE, Xiong et al. [19] proposed a partial-hidden policy and computation-outsourced method. Linear secret sharing scheme (LSSS) is used to setup the policy, which creates a matrix for matching; only the user with certain attributes can recover the shared secret. However, edge outsourcing only proxy-decrypts ciphertext. It does not have any information to check whether the transformed ciphertext it computes is true or false, which can only be verified by the devices. It would become a problem since the wrong result still be broadcast to all the devices. This problem should have some mechanisms to pre-handle in the edge.

Yang et al. [29] proposed a subscriber-publisher matching protocol. Linear secret sharing scheme (LSSS) is used for matching. Outsourcing is also preserved to help IoT devices. Different from [19], it provides a mechanism to check the matching result. The ciphertext-transform phase-only processes when the policy and the attribute are matched. The drawback of this method is that the policy and the attribute need to match specifically. We can not have a subset of the attribute or policy, which is not flexible for matching.

Zhou et al. [18] proposed an ABE-based protocol to do the match. The special point of this method is that the policy can have wildcard value, which means the policy does not care about the certain attributes the devices have or not. This property may solve the problem in [20].

In addition to the match phase, data feedback is the key phase in IoT architecture. All need to do is to get the data from the IoT devices or get the service. One of the methods is Multi-receiver. Base on some mathematical basis, a ciphertext can be decrypted by some certain private keys, which can reduce the transmission cost. In 2006, Ng et al. [20] proposed a protocol based on polynomial inner product to achieve the goal. With basic security requirements, the computing cost is low in data encryption, which is appropriate to IoT devices due to the low computing resource. In 2010, Fan et al. [21] proposed an anonymous multi-receiver protocol based on the Lagrange interpolating polynomial theorem. However, the sacrifice of computing cost represented in both encryption and decryption phases.

Li et al. [24] proposed a scheme for a fuzzy search of encrypted data. With this scheme, data providers can upload encrypted data to the server, and data consumers can search for encrypted data. This scheme can achieve private search data. However, this scheme is based on cloud architecture. There will be high latency and high computational load. In this

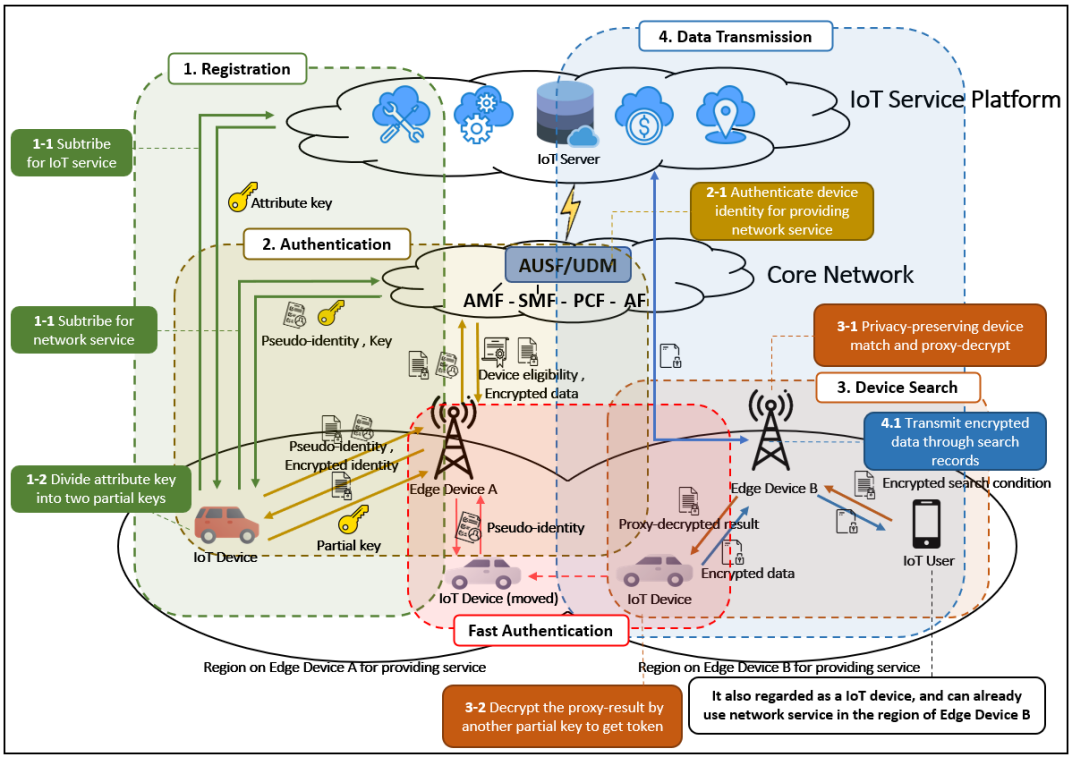

Fig. 1. System model

scheme, the data consumer must have data providers trapdoor, so this scheme cannot be used to search for unfamiliar devices.

Mutual authentication is the basis between devices. Only the authenticated parties can process the following phase. Fan et al. [25] proposed a region-based fast handover authentication. The base-station topology is Hierarchical, which is referring to as eNB and HeNB in 5G. Handovers between HeNBs do not need high-level authentication, which is appropriate for IoT architecture. In future 5G, small cell base-stations would be a trend. The data can be shared between authenticated parties, which authentication mechanism is much more important.

III. System Construction

A. System Model

This section introduces IoT networks within the edge devices. In the future 5G environment, the base station will divide into many small cells, and the IoT devices under each small cell will provide various services for the needs of IoT users. In one of the use cases, the IoT users want to get some interesting data from certain IoT devices, but they do not want others to know what they interested in. Thus, in this case, both parties must encrypt the message to hide some information. However, both parties are limited-computing resource devices, so our proposed scheme has the edge device between the two parties to outsource to match hidden information from the two parties by using its powerful computing resource. Fig. 1 shows our system model, there are five entities: Core Network, IoT Server, Edge Device, IoT User, and IoT Device.

- **Core Network:** It provides network for IoT devices and creates pseudo identity for IoT devices to preserve privacy of the IoT devices. Thus, the core network can leverage pseudo identity to authenticate the IoT devices and trace back the real Identity of malicious IoT devices.
- **IoT Server:** It provides the key for the IoT devices and has a secure server to store IoT data.
- **Edge Device:** The edge device is honest-but-curious. It will verify the permission of the IoT user and the IoT devices through the certificate issued by the core network. It has secure storage space to store data and powerful computing resources for matching of device search. Also, it can provide data of IoT devices under other edge devices through the backend server.
- **IoT User:** The IoT user may be a mobile phone or device. It has the certificate and key issued after authentication. It can send a device search request to the edge device and prove its search permission through the certificate.
- **IoT Device:** The IoT device is a device with limited computing resources. It has the pseudo identity and attribute key issued by the IoT server. It regularly sends information to matching IoT user through edge device.

B. Security Threats

In [30], several types of attacks have been discussed, including user authentication, user privacy, data integrity, etc., in 5G networks. Besides, this section discuss the additional security threats to the proposed system model as follows.

- **Mutual authentication of devices:** Due to the defect of the Global System for Mobile Communications (GSM) specification, authentication is not required in the process of communication, which derived the problem of fake base station attack. Fake base station attack may pretend the Telecom operator to induce users to click the malicious website or get the sensitive information of the user for some benefits. The user does not know that it is an attack and may follow the steps of attack, which has occurred in practical. Therefore, mutual authentication is needed to ensure that the person or device communicating is a certain one, not an adversary or a third-party.
- **Honest-but-curious edge device:** The edge device can help match IoT users/devices, transmit data, even calculate for devices. However, they may be honest-but-curious which means they may record the data pass through them intentionally or unintentionally. With this personal information, the edge device may analyze and get benefits, for example, advertisement recommendation. The edge devices do not need to know the data unless they are the data consumer. We need some mechanisms to protect the data from eavesdropping.
- **User anonymity:** Once the adversaries know the type of devices, they can devise a specific attack method to invade the devices or the system. On the other hand, real identity would be aim to be the attack target if we use real identity in communication. We need to hide the real identity or change the pseudo identity in each communication.
- **Data integrity and source:** In the man-in-the-middle attack(MITM), The adversary may modify or forge the data during the communication, which distorts the original meaning of data. On the other hand, the source of data also a significant problem we need to consider. For example, an update firmware from an adversary may endanger a system. Therefore, some cryptographic algorithms are needed to ensure data integrity and source.
- **User cheating:** Due to the benefits from the system, we assume that the user would not want to break the system. However, the Users may be curious about other users' services or data they request or prefer to. Therefore, they may pretend other legitimate users to send the request and analysis the response, moreover, colluding with other illegal users. We need some methods to track back the real identity of the cheating user.

C. Security Requirements

- **Authenticated key exchange:** Before the the IoT user and the IoT device send requests or information, both of them need to be authenticated to the edge device to ensure their permission. Moreover, both of them should be able to exchange the session key securely to protect the security of data transmission between the edge device.
- **Device anonymity:** The IoT user and the IoT device should remain anonymous. When the same device sends a request or information to the edge device again, the edge device can not know the message sent by the same device.
- **Privacy-preserving match:** The requests and information sent by the IoT user and the IoT device to the edge device should be protected. The edge device only can match the encrypted message without knowing its original message and the attributes of interest.
- **Computation cost outsourcing:** Due to the limited computing cost, the IoT device cannot afford much computation task. Therefore, we want to outsource part of decryption to edge-devices to form a transformed ciphertext, i.e. proxy-decryption in edge-devices. After proxy-decryption, the transformed ciphertext can be simply decryption in IoT device to get the plaintext.
- **Traceability:** The system can trace back the real identity of the device to avoid malicious access control of the IoT user and malicious behaviour provided by the IoT device even if the device is anonymous.

IV. Preliminaries

A. Bilinear Pairings

Given two multiplicative cyclic groups with same prime order P. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear pairing with following properties[27]:

1) Bilinearity: For $g, h \in \mathbb{G}, a, b \in \mathbb{Z}_p, e(g^a, h^b) = e(g, h)^{ab}$.
2) Non-degenracy: For $g \neq 1_{\mathbb{G}}, h \neq 1_{\mathbb{G}}, e(g, h) \neq 1_{\mathbb{G}_T}$.

B. The Vites Formulas

Given a polynomial formula[28]:

$$Let \prod_{i=1}^{n}(x - r_i) = \sum\nolimits_{k=0}^{n} a_k x^k$$

with $r_1, r_2, ..., r_n$ as roots, the coefficient a_k of each degree can be presented as:

$$\begin{cases} r_1 + r_2 + \cdots + r_{n-1} + r_n = -\frac{a_{n-1}}{a_n} \\ (r_1 r_2 + r_1 r_3 + \cdots + r_1 r_n) + \cdots + r_{n-1} r_n = \frac{a_{n-2}}{a_n} \\ \vdots \\ r_1 r_2 \ldots r_n = (-1)^n \frac{a_0}{a_n} \end{cases}$$

C. Access Structure

We use the Phuong proposed access structure[26] which is AND-gates with wildcard in ABE. Because it has wildcard, it makes the policy more flexible. Let $U = \{Att_1, Att_2, ..., Att_L\}$ be the universe of attributes in the system. it has L attributes and each attribute has unique value A_i. When a new user joins the system, the user will be tagged as $S = \{S_1, S_2, ..., S_L\}$ which is his/her personal attribute list. In this list, each element has two possible symbols: $+$ and $-$ where donate positive and negative. However, in the list of access policy with wildcard such as $W = \{W_1, W_2, ..., W_L\}$, each element has three possible symbols: +, - and * where donates wildcard and means "do not care", in other words, it imply that both positive and negative attributes are accepted.

Then, we can use the notation $S \vDash W$ to donate that the attribute list S of a user satisfies W.

D. Hiding Policy by Inner Product Encryption

According to the idea of Phuong et al. [26], the policy of the data consumer and the attribute of the data provider are converted into two vectors to apply the technique of inner product encryption to hide policy. According to the access policy structure, the policy of the data consumer has three sets J, V and Z,and the attribute of the data provider has two sets V and Z, where J, V and Z donate the positions for wildcards, positive and negative attributes.

According to Vieta's formulas, the set J of the policy of the data consumer can construct a polynomial $\sum_{k=0}^{n} a_k i^k$ with coefficients $(a_0, a_1, ..., a_n)$, and other sets of the data consumer V, Z can respectively combine to calculate as follows:

$$\Pi_V = +\sum\nolimits_{i \in V} \prod\nolimits_{w_j \in J} (i - w_j)$$

$$\Pi_Z = -\sum\nolimits_{i \in Z} \prod\nolimits_{w_j \in J} (i - w_j)$$

So, the policy of the data consumer can express as a vector as:

$$\vec{V} = (a_0, a_1, ..., a_n, 0_{n+1}, ..., 0_{N_1}, \Pi_V, \Pi_Z)$$

The vector $\vec{V}$ will be used in encryption phase.

The sets of the attribute of the data provider will be respectively calculated as two vectors $\vec{X_v}$ and $\vec{X_z}$ as follows:

$$v'_k = -\sum\nolimits_{i \in V'} i^k, k = \{0, 1, ..., N_1\}$$

$$z'_k = +\sum\nolimits_{i \in Z'} i^k, k = \{0, 1, ..., N_1\}$$

where N_1 is the maximum number of wildcard in access policy structure. The two vectors are expressed as follows:

$$\vec{X'_V} = (v'_0, v'_1, ..., v'_{N_1}, 1, 0)$$

$$\vec{X'_Z} = (z'_0, z'_1, ..., z'_{N_1}, 0, 1)$$

The two vectors will be used in generating key phase.

Finally, if the attribute meet the policy, both the inner products of $(\vec{V}, \vec{X'_V})$ and $(\vec{V}, \vec{X'_Z})$ will be calculated as 0.

V. Propose Scheme

A. The high-level description of the proposed scheme

In this paper, the proposed scheme will perform in Six phases: Initialization, Registration, Authentication, Device Search, Data Transmission and Fast Authentication, and the detail flow is shown as fig.2.

- **Initialization** In this phase, there are two entities, the core network operator and the IoT application server. Both of them need to setup some parameters to initialize the whole system. The core network operator will initialize the component of the core network for IoT device, generate the system parameter bP_j and the group key GK to each group of the edge devices. The IoT application server will generate paring parameter and the random number to compute master key pairs, MSK and MPK. The key pairs of all IoT devices derive from the master key pairs.
- **Registration** In order to access this system, first, the IoT users/devices need to register with the core network with their real identity ID to build a network connection. The core network operator will share the key K_i that only both parties know and create two pseudo identities, rID and pID, which are used to hide its real identity. Then, the IoT users/devices also need to register with the IoT application server with their attributes att to access the IoT service, and the IoT application server will create the key $attkey$ for them according to their attributes. Finally, the IoT devices divide this key into partial decryption key pdk and decryption key dk.
- **Authentication** Authentication is one of the key phases in our system. Only the authenticated devices can get the allowance to communicate with others legally. Once the devices get hacked, unauthenticated devices may be a choice for the adversary to launch attacks. In our system, [25] is adopted to support our authentication demand. However, we make a little change on the nyeber function due to the high cost of computation. Thus, when the IoT user/device enters the coverage of the new edge device, it will send a challenge with its pseudo identity pID to the Authentication service function(AUSF) in the core network via the edge device. The AUSF will check the validation of pID and response the challenge. Finally, the IoT user/device complete authentication with the edge device and the core network, and send its pdk to the edge device.
- **Device Search** The IoT user wants to get some data that it is interested in from certain devices, but it does not want to leak any information to any devices, even edge device. On the other hand, the IoT devices also do not want to leak any of their attribute information to other devices. Therefore, they need a hidden policy ABE scheme to meet the requirements of privacy matching. Fortunately, the scheme which Phuong et al. proposed [26] suites to our system. Nevertheless, some IoT devices in our system are limited-computing resource device, so they can not afford the complicated computation of ABE. So, base on the idea of the scheme which Phuong et al. proposed, we add an outsourcing mechanism that the edge device can help IoT devices to make complex compute such as ABE decryption. Moreover, the edge device can not know the context of ciphertext because it can only partially decrypt ciphertext by a partial key provided by the IoT devices. Thus, the IoT user encrypts his/her token which as a session key with the matched device by his/her policy and sends the ciphertext CT as the data search request to the edge device.

  After receiving the requests, the edge device will record the CR of IoT user for data transmission phase and proxy-decrypt the requests using each pdk in its pdk list. The proxy results will be sent back to the IoT devices according to the pdk. Finally, only IoT devices whose attributes meet

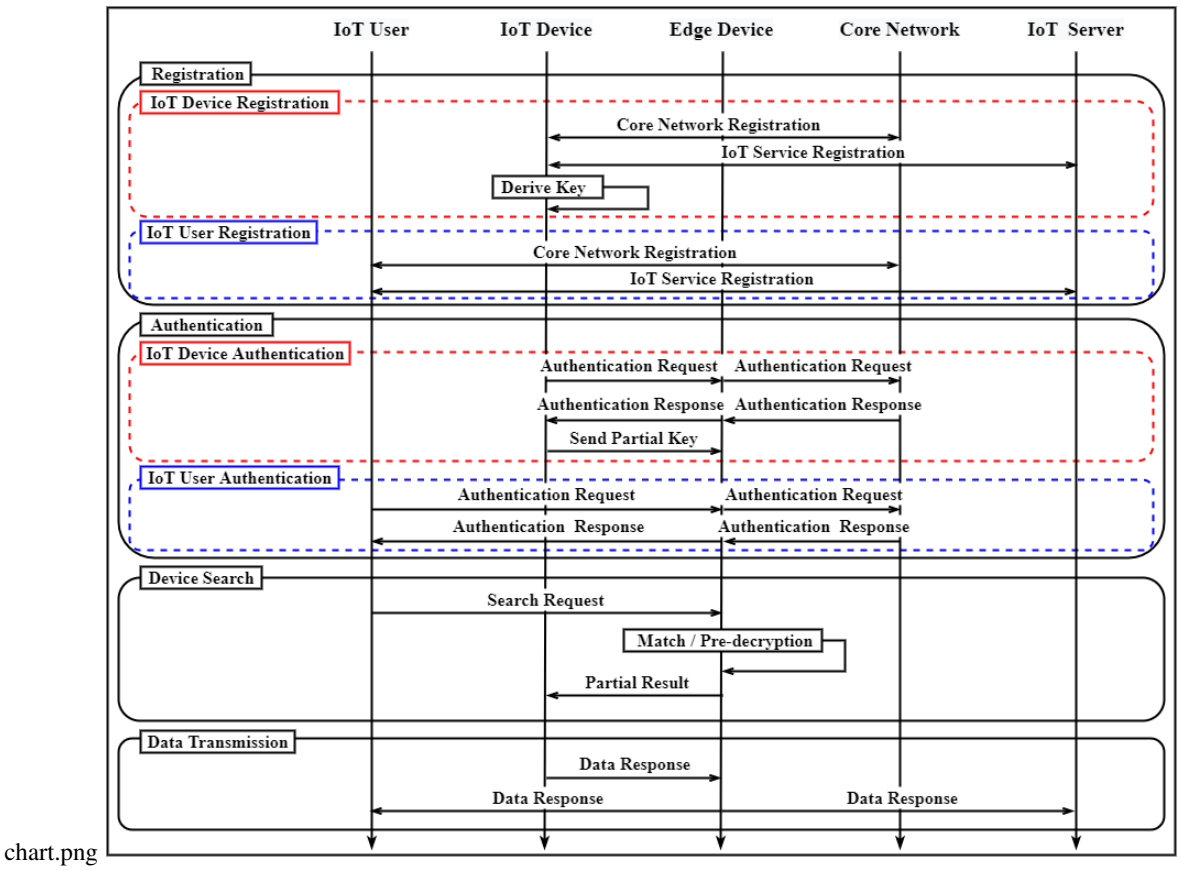

Fig. 2. The flow chart of the proposed scheme

to the policy which the IoT user requests can get the real token.

- **Data Transmission** The IoT device would get a token from the IoT user if its attributes met the policy of the IoT user. The token is used for the decryption of the data; therefore, we need to design the data transmission scheme between IoT users and IoT devices. The multi-receiver method in [20] is more appropriate than our system since the IoT devices have limited computing resource. Base on the polynomial inner product, the data only needs to be encrypted one time without any pairing computing cost, which can eliminate the computing cost in IoT devices sharply. Thus, the IoT devices use tokens to encrypt the data by the polynomial inner product and regularly send the receiver list and the ciphertext to the edge device. When the edge device receives the message, it will take the CR in the list as a label to search the CR table and send the ciphertext to the corresponding IoT user. If the CR of the list is not in the CR table of this edge device, the edge device will upload the ciphertext to the database of the IoT server, and other edge devices can get the ciphertext from the database. Finally, the IoT user can decrypt the receiving ciphertext by own token.
- **Fast Authentication** When the IoT device moves from this edge device to a new edge device within the same group, it can fast authenticate with new edge device and successfully send data response through the IoT server to the IoT users who subscribe this IoT device.

B. The detail scheme

In our scheme, we assume that the system has L attributes in the universe, each attribute has two possible symbols which express as positive or negative. In addition, a wildcard symbol can be used in the access policy structure. Then, there are maximum number of these three symbols N_1, N_2, N_3 as their upper bound, where $N_1 \leq L$ as the maximum number of the wildcard symbol in access structure, $N_2 \leq L$ as the maximum number of the positive symbol in access structure, $N_3 \leq L$ as the maximum number of the negative symbol in access structure.

1) ***Initialization****:* In this phase, the the core network operator will create GK_j, bP_j^i and P_j^{st} for macro edge devices to ensure the validity of IoT users/devices under the macro edge devices.

On the other hand, the IoT application server sets $n = N_1 + 3$ and randomly generates bilinear mapping parameter $(p, g, \mathbb{G}, \mathbb{G}_T, e)$ and random number $\mu_1, \mu_2, \omega_1, \omega_2, r_{1,i}, t_{1,i}$ and$\{a_{j,i}, b_{j,i}\}_{j=1}^2 \in \mathbb{Z}_p$,where $i = 1$ to n. Then, it computes $r_{2,i}$ and $t_{2,i}$ by $\lambda = r_{2,i}\mu_1 - r_{1,i}\mu_2 = t_{2,i}\omega_1 - t_{1,i}\omega_2$, where

$\lambda \in \mathbb{Z}_p$ and $i = 1$ to n. Next, it sets $g_1 = g^\lambda$ and $g_2 \in \mathbb{G}$. Finally, it generates MPK and MSK as:
$MPK = (p, g, \mathbb{G}, \mathbb{G}_T, e, g_1, (g, g_2), \{R_{j,i} = g^{r_{j,i}}, T_{j,i} = g^{t_{j,i}}, A_{j,i} = g^{a_{j,i}}, B_{j,i} = g^{b_{j,i}}, U_j = g^{\mu_j}, W_j = g^{\omega_j}\}$,where $j = 1$ to 2, $i = 1$ to n).
$MSK = (g_2, \{r_{j,i}, a_{j,i}, t_{j,i}, b_{j,i}, \mu_j, \omega_j\}$, where $j = 1$ to 2, $i = 1$ to n).

*2) **Registration**:* when a new IoT $\text{user}_\text{i}/\text{device}_\text{i}$ join the system, it needs to register to the IoT application server with its ID_i and attributes $att = \{att_1, att_2, ..., att_{N_2}\}$. According to the ID_i, the core network operator will generate and compute K_i, rID_i and $pID_i = E_{K_H}(rID_i)$ to send to the IoT $\text{user}_\text{i}/\text{device}_\text{i}$. On the other hand, after receiving the att of the IoT $\text{user}_\text{i}/\text{device}_\text{i}$, the IoT application server will convert att into two vector $\vec{X_v}$ and $\vec{X_z}$ where $\vec{X_z} \notin att$ and generates random number $\theta_{1,i}, \theta_{2,i}$ for $i = 1$ to n and $\gamma_1, \gamma_2, \theta_1, \theta_2 \in \mathbb{Z}_p$. Finally, the IoT application server creates a exclusive key for the IoT user/device.
$attkey = (K_\alpha = g_2 \cdot \prod_{i=1}^n K_{1,i}^{-a_{1,i}} K_{2,i}^{-a_{2,i}} K_{3,i}^{-b_{1,i}} K_{4,i}^{-b_{2,i}}$,
$K_\beta = \prod_{i=1}^n g^{-(\theta_{1,i}+\theta_{2,i})}, K_{1,i} = g^{-\mu_2\theta_{1,i}} g^{\gamma_1 x_{V_i} r_{2,i}}, K_{2,i} = g^{\mu_1\theta_{1,i}} g^{-\gamma_1 x_{V_i} r_{1,i}}, K_{3,i} = g^{-\omega_2\theta_{2,i}} g^{\gamma_2 x_{Z_i} t_{2,i}}$, and $K_{4,i} = g^{\omega_1\theta_{2,i}} g^{-\gamma_2 x_{Z_i} t_{1,i}})$.

When the IoT $\text{user}_\text{i}/\text{device}_\text{i}$ receives $attkey$ from the IoT application server, it generates a random number $\nu \in \mathbb{Z}_p$, then sets $dk = \nu$ and $pdk = attkey^{1/\nu}$ as:
$pdk = attkey^{1/\nu} = (K'_\alpha = K_\alpha^{1/\nu}, K'_\beta = K_\beta^{1/\nu}$,
$K'_{1,i} = K_{1,i}^{1/\nu} = (g^{-\mu_2\theta_{1,i}} g^{\gamma_1 x_{V_i} r_{2,i}})^{1/\nu}$,
$K'_{2,i} = K_{2,i}^{1/\nu} = (g^{\mu_1\theta_{1,i}} g^{-\gamma_1 x_{V_i} r_{1,i}})^{1/\nu}$,
$K'_{3,i} = K_{3,i}^{1/\nu} = (g^{-\omega_2\theta_{2,i}} g^{\gamma_2 x_{Z_i} t_{2,i}})^{1/\nu}$,
$K'_{4,i} = K_{4,i}^{1/\nu} = (g^{\omega_1\theta_{2,i}} g^{-\gamma_2 x_{Z_i} t_{1,i}})^{1/\nu})$.

*3) **Authentication**:* when a IoT $\text{user}_\text{i}/\text{device}_\text{i}$ move from an edge device_k to an edge device_j i.e. from $group_k$ to $group_j$, some steps need to be set up to authenticate the IoT $\text{user}_\text{i}/\text{device}_\text{i}$. First, the IoT $\text{user}_\text{i}/\text{device}_\text{i}$ selects a one-time key κ at random and sends pID_i to the edge device_j and $(pID_i, \Delta_1 = E_{k_i}(pID_i, \kappa))$ to the AUSF in the core network through the edge device_j. After receiving Δ_1, the AUSF decrypts Δ_1 to get pID_i and κ. To ensure the identity of the IoT $\text{user}_\text{i}/\text{device}_\text{i}$, the AUSF check whether pID_i (decrypted) is equal to the pID_i (sent by IoT $\text{user}_\text{i}/\text{device}_\text{i}$). It then selects a new rID_i^* and computes $TID_{ij} = rID_i^* \oplus bP_j^I$ where $I \in \{1, k\}, pID_i^* = E_{K_H}(rID_i^*), Q_{ij} = H(GK_j, rID_i, T_x), \Delta_2 = E_{K_i}(TID_{ij}, Q_{ij}, T_x, \kappa, pID_i^*)$ and adds Q_{ij} into P_j^{st}. After receiving Δ_2 through edge device_j, the IoT $\text{user}_\text{i}/\text{device}_\text{i}$ decrypts Δ_2 by K_i to get $TID_{ij}, Q_{ij}, T_x, \kappa$ and pID_i^*. It then replaces $pID_i = pID_i^*$ and updates TID_{ij}.

The P_j^{st} will be sent from the core network to $group_j$ in every time slot, which is used to check whether the IoT $\text{user}_\text{i}/\text{device}_\text{i}$ should in this group.

After the authentication phase, the edge device_j can authenticate the upcoming IoT $\text{user}_\text{i}/\text{device}_\text{i}$, and the IoT $\text{user}_\text{i}/\text{device}_\text{i}$ will send its pdk to the edge device_j to prepare for device search phase.

*4) **Device Search**:* As mention above, the IoT user converts its policy into a vector $\vec{V}$, and generates random numbers $Symkey, token, F_1, F_2, \alpha$ and $\beta \in \mathbb{Z}_p$, and sets $MK = Symkey$ and creates the ciphertext $CT = (C_M = (MK \cdot e(g, g_2)^{F_2}), C_X = g^{F_2}, C_Y = g_1^{F_1}, C_R = Enc_{Symkey}(token), C_{1,i} = R_{1,i}^{F_1} T_{1,i}^{F_2} V_1^{v_i\alpha}, C_{2,i} = R_{2,i}^{F_1} A_{2,i}^{F_2} U_2^{v_i\alpha}, C_{3,i} = T_{1,i}^{F_1} B_{1,i}^{F_2} W_1^{v_i\beta}$, and $C_{4,i} = T_{2,i}^{F_1} B_{2,i}^{F_2} W_2^{v_i\beta})$.

When the edge device receives the request which the IoT user send, the edge device executes matching algorithm with pdk of the IoT device to decrypt CT partially, and store C_R in secure storage. It will compute:

$$C'_{M_1} = C_M$$

$$C'_{M_2} = e\left(C_X, K'_\alpha\right) \cdot e\left(C_Y, K'_\beta\right) \prod_{j=1}^{4} \prod_{i=1}^{n} e\left(C_{j,i}, K'_{j,i}\right)$$

$$= e(g, g_2)^{F_2/\nu}$$

$$C'_M = (C'_{M_1}, C'_{M_2})$$

When the IoT device receives C'_M, it uses dk to decrypt it.

$$MK = \frac{C'_{M_1}}{{C'_{M_2}}^{\nu}} = \frac{MK \cdot e(g, g_2)^{F_2}}{(e(g, g_2)^{F_2/\nu})^{\nu}}$$

Then, it uses $MK = Symkey$ decrypts C_R.

$$token = Dec_{Symkey}(C_R)$$

Finally, the IoT device and the IoT user share secret $token$, and the IoT device uses $token$ to encrypt data. The next subsection will introduce how to securely transmit data to the matched IoT device user.

*5) **Data Transmission**:* First, the IoT devices use the tokens from the IoT users to construct the polynomial function $f(x) = \Pi_{i=1}^n (x - token_i) = \Sigma_{i=0}^n a_i x^i$ and compute $\{g_0, g_1, ..., g_n\} = \{g^{a_0}, g^{a_1}, ..., g^{a_n}\}$. Second, the IoT devices select two random number $k_1, k_2 \in \mathbb{Z}_q^*$ and raise each component in the encryption tuple of power k_2, i.e., calculate $\{g_0^{k_2}, g_1^{k_2}, ..., g_n^{k_2}\}$ and compute $A = k_1 \cdot g_0^{k_2}$. The message m is encrypted as $Z = m \oplus k_1$ and the ciphertext $C = \{Z, A, g_1^{k_2}, ..., g_n^{k_2}\}$, and $C_R = \{C_{R_1}, C_{R_2}, ..., C_{R_n}\}$ are sent to the edge device. The edge device checks IP according to the C_R in IP_table and send C to $IP = \{IP_1, ..., IP_n\}$. After receiving C from edge device, the IoT users compute $k = A \cdot \Pi_{j=1}^n g_j^{k_2 \cdot token^j} = k_1 \cdot g_0^{k_2} \cdot \Pi_{j=1}^n g_j^{k_2 \cdot token^j} = k_1 \cdot \Pi_{j=0}^n g_j^{k_2 \cdot token^j} = k_1 \cdot g^{k_2 \cdot \Sigma_{j=0}^n a_j token^j} = k_1 \cdot g^{f(token \cdot k_2)} = k_1 \cdot 1^{k_2} = k_1$. The IoT users decrypt ciphertext Z as $m = Z \oplus k_1$.

*6) **Fast Authentication**:* At the fast authentication phase, the edge device_j2 (in the same $group_j$) can authenticate the upcoming IoT $\text{user}_\text{i}/\text{device}_\text{i}$ in several steps. The IoT $\text{user}_\text{i}/\text{device}_\text{i}$ sends TID_{ij}, T_x and r_u to edge device_j2. edge device_j2 computes $\epsilon ID = TID_{ij} \oplus bP_j^I, Q_{ij} = H(\epsilon ID, GK_j, T_x)$ and check whether Q_{ij} in P_j^{st} or not. If so, it then checks the T_x. After both passed, The

edge $\text{device}_{\text{j2}}$ randomly selects r_h and computes $K_{H_{e,i}} = H(Q_{ij}, r_u, r_h), R = H(Q_{ij}, r_u, r_h, K_{H_{e,i}}), TID_{ij} = \epsilon ID \oplus bP_j^I$ and $\Delta_3 = E_{Q_{ij}}(TID_{ij}, I)$ and send $(R, r_h$ and $\Delta_3)$ to IoT $\text{user}_{\text{i}}/\text{device}_{\text{i}}$. After receiving $(R, r_h$ and $\Delta_3)$, the IoT $\text{user}_{\text{i}}/\text{device}_{\text{i}}$ decrypts Δ_3 by Q_{ij} and replaces its TID_{ij} with TID_{ij} and I with I. On the other hand, the IoT $\text{user}_{\text{i}}/\text{device}_{\text{i}}$ computes $K_{H_{e,i}} = H(Q_{ij}, r_u, r_h)$ and checks whether $R = H(Q_{ij}, r_u, r_h, K_{H_{e,i}})$. If so, it then computes $\Upsilon = H(K_{H_{e,i}}, r_h)$ and sent it back to the edge $\text{device}_{\text{j2}}$. The edge $\text{device}_{\text{j2}}$ checks whether $\Upsilon = H(K_{H_{e,i}}, r_h)$. If so, the IoT $\text{user}_{\text{i}}/\text{device}_{\text{i}}$ has been authenticated successfully.

VI. Security Analysis

- **Authenticated key exchange:** In our system, with the help of the AUSF in the core network, when the IoT users/devices want to build connection with the edge device, they will send the authentication request to the AUSF via the edge device. The AUSF will take the pID from IoT users/devices to extract the corresponding K_i in UMF. If the AUSF can get the κ by using K_i to decrypt the request Δ_1, it will generate new pseudo identity pID^* to computing new session key by $H(\kappa, pID^*)$.
  In fast authentication phase, the edge devices in a group can get a list that includes the information of the upcoming IoT device. After the IoT device sends the fast authentication request, the edge devices can check whether they get permission from the core network. If so, the edge-devices then start a challenge and response game to authenticate the IoT device.
  Challenge and response game: The IoT device generates a random nonce r_u to the edge device as the challenge. After the edge device receives the challenge, it will check the TID and T_{ex} of the IoT devices according to the list R_j^{st} from TA and timestamp. When the edge device responds, the IoT device can authenticate the edge device with the r_u. Similarly, the random nonce r_h created by the edge device is a challenge to authenticate the IoT device. After that, Both of them can share a session key by r_u and r_h to achieve secure communication and mutual authentication.
  Based on [25], we have an efficiency authenticate protocol to ensure the legal devices in the proposed scheme.
- **Device anonymity:** After registering to the core network, the IoT users/devices will get a pID and rID which is used in the following authentication phase. The real ID would not be sent to any other parties except for the core network. The edge devices only authenticates the legitimacy with pID and rID without knowing the real identity of IoT devices, which can achieve the demand of devices anonymity.
- **Privacy-preserving match:** The IoT user uses hidden ciphertext policy ABE to encrypt the message, and the edge device can not know the policy of the IoT user. The IoT device divides its key into two partial keys, which means that we need two keys to decrypt the ciphertext completely. One of the keys will be sent to the edge device to partial decrypt, which can reduce the computing burden of the IoT device. Thus, even if the edge device uses a partial key sent by the IoT device to decrypt the ciphertext, the edge device still can not know the message and whether the matching is successful.
- **Computation cost outsourcing:** The IoT device derives two partial key from its key and sends one partial key to the edge device to outsource the decryption. If the attribute of the partial key meets the policy of the ciphertext, the edge device can partially decrypt successfully. However, the edge device still can not know the content of the plaintext because the key is not complete. Therefore, the IoT device can safely outsource the decryption to the edge device.
- **Traceability:** At the beginning of the IoT devices join a system; they need to register to the core network with their real ID. Once they disclose or collude to any adversary or illegal device, we can trace the specific devices with the help of the core network and revoke them, which can maintain the stability and safety of the proposed scheme.

VII. Evaluate and Performance Comparisons

In this section, we compare the security features, performance with these related works [6], [19] and [29] and evaluate the computing costs.

A. Comparisons on Security Features

We compare the security features of the proposed scheme with these related works [6], [19] and [29] as shown in Table III. For mutual authentication, in Ref. [19] and [29], the edge device or cloud server did not authenticate both sides. For the privacy-preserving match, Ref. [6] only use one attribute as a matching condition which is not flexible in current IoT environments. In Ref. [29], the publishers and the subscribers did not hide all policy. For computation outsourcing, in Ref. [6], the service provider uses proxy re-encryption to transform the ciphertext, which can be decrypted by the key of the data consumer. However, the computing cost for IoT device does not reduce. For traceability, in Ref. [19], the data sharer can not trace back the identity of the data owner.

B. Computation Costs

We use java pairing-based cryptography (JPBC) library to implement the necessary pairing operations in our system. We deploy core network and IoT application server on a desktop with Intel Core i7-9700 CPU, the clock rate is 3.00GHz and the memory is 16GB. The operating system is 64-bits Windows 10 and the JDK version is 14.0.1 on Eclipse IDE 2020-06 64-bits. The Edge device is deployed on a laptop with AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx, the clock rate is 2.3GHz and the memory is 16GB. The operating system is 64-bits Windows 10 and the JDK version is 11.0.5 on Intellij IDEA Community Edition 2019.3.3. The IoT user/device is an OPPO R17 Pro smartphone running on Android 8.1 mobile operating system and equipped with 2.2GHz Qualcomm Snapdragon 710 CPU and 6GB RAM. We use the above hardware to evaluate the computing cost as shown in Table I.

TABLE I
COMPUTATION COSTS (DEVICE = 3)

RT(ms)[2] / OP[3] / PL,T_ds[1]	IoT Server		CN[4]	Edge Device		IoT Devices/Users					
	Setup	KeyGen	Auth[5]	FAuth[6]	Match	AuthReq	ReqGen	KeyDer[7]	FinalDec	DataEnc	DataDec
PL=2, T_ds=4	381	508	9	135	171	8	5364	672	1010	1	1
PL=6, T_ds=4	646	897	9	119	250	7	6132	1118	1038	1	1
PL=10, T_ds=4	909	1279	8	120	271	5	8712	1600	1117	1	1
PL=6, T_ds=8	649	902	7	120	203	9	6521	1166	1020	1	1
PL=6, T_ds=12	647	903	9	114	213	7	6617	1117	993	1	1
PL=6, T_ds=16	649	889	10	119	203	7	6250	1164	982	1	1

[1] PL,T_ds: Policy Length, Token for data sharing [2] RT(ms): Runtime(ms) [3] OP: Entites and Operations
[4] CN: Core Network [5] Auth: Authentication [6] FAuth: Fast Authentication [7] KeyDer: Key Derive

TABLE II
COMPUTATION COSTS (WILDCARD = 10, TOKEN = 4)

RT(ms)[2] / OP[3] / Dev[1]	IoT Server		CN[4]	Edge Device		IoT Devices/Users					
	Setup	KeyGen	Auth[5]	FAuth[6]	Match	AuthReq	ReqGen	KeyDer[7]	FinalDec	DataEnc	DataDec
Dev = 2	918	1304	10	119	227	8	8004	1615	1057	2	1
Dev = 4	907	1291	7	117	648	8	8720	1569	1010	1	1
Dev = 6	909	1279	8	120	1180	5	8712	1600	1131	1	1

[1] Dev: The number of devices [2] RT(ms): Runtime(ms) [3] OP: Entites and Operations
[4] CN: Core Network [5] Auth: Authentication [6] FAuth: Fast Authentication [7] KeyDer: Key Derive

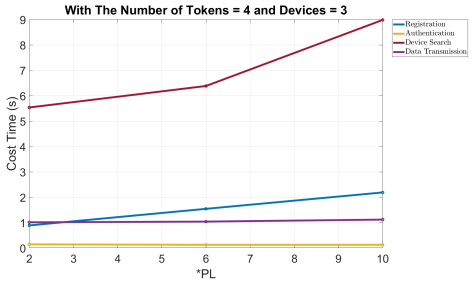

Fig. 3. Cost Time of different length of policy

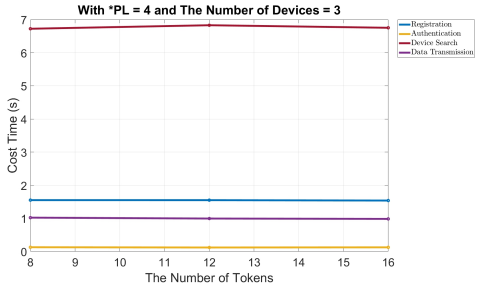

Fig. 4. Cost Time of different number of tokens

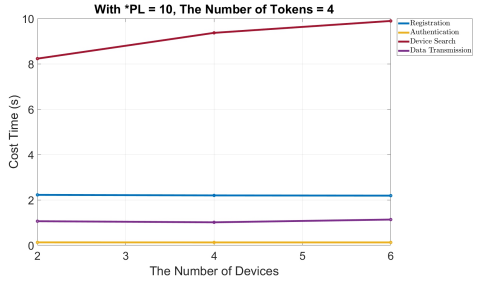

Fig. 5. Cost Time of different number of devices

TABLE III
COMPARISONS ON SECURITY FEATURES

Security Features	[6]	[19]	[29]	Ours
Mutual Authentication	✓	✗	✗	✓
Device Anonymity	✓	✓	✗	✓
Privacy-preserving Match	✓*	✓*	✓*	✓
Computation Outsourcing	✗	✓	✓	✓
Traceability	✓	✗	✗	✓

* The match of the policy is partial hidden.

In Table I, we have three devices. The policy length is based on the maximum number of wildcards. If the policy length changes, the computing cost of the functions Setup, KeyGen, Match, ReqGen, and KeyDer get higher as the maximum number of wildcards increased. However, the number of tokens for data sharing does not affect. In Figure 3, the Registration includes Setup and KeyGen time and the Device Search includes ReqGen and Match time. Therefore, the cost of them presents linear growth. The Authentication includes AuthReq and FAuth time and the Data Transmission includes FinalDec, DataEnc and DataDec time. Therefore, the cost of them presents constant. On the other hand, the number of tokens for data sharing does not affect the time in every function or procedure in Figure 4.

In Table II, we test the effect of the number of devices in every function. The results show that Match time is proportional to the number of devices and does not affect others. In Figure 5, the Device Search time presents linear growth since the growth of Match time.

VIII. CONCLUSION

In this work, we proposed a privacy-preserving matching and outsourcing work with an authenticated multi-receiver protocol to support future IoT architecture. Due to the low

computing resource in IoT devices, the setting of edge devices has been discussing in recent years. The edge devices can help pre-process the data from both parties, which is useful in many scenarios. However, edge devices are honest-but-curious, which means they may save any information passed through them. Privacy-preserving is a big problem in such a situation. On the other hand, mutual authentication between devices is required. Only authenticated parties can get the allowance to access the data. A hierarchical topology of edge devices is proposed to solve this problem. Finally, to reduce the transmission cost, a multi-receiver method is adopted based on the polynomial inner product, which can achieve the goal of one encryption and multi decryption. All in all, this paper found a balance between the users privacy and requirement matching and proposed corresponding methods to solve the derived security requirement.

ACKNOWLEDGMENT

This work was partially supported by the Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU), in part by the Ministry of Science and Technology of Taiwan under Grant MOST 108-2221-E-110-033, in part by the Information Security Research Center, National Sun Yat-sen University, Taiwan.

REFERENCES

[1] Veena Pureswaran, IBM Institute for Business Value, "Device democracy Saving the future of the Internet of Things," Accessed: 2015, [Online]. Available: https://www.ibm.com/thought-leadership/institute-business-value/report/device-democracy

[2] G. A. Akpakwu, B. J. Silva, G. P. Hancke and A. M. Abu-Mahfouz, "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges," *IEEE Access,* vol. 6, pp. 3619-3647, 2018.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 4, pp. 2347-2376, Fourth quarter 2015.

[4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal,* vol. 4, no. 5, pp. 1125-1142, October 2017.

[5] Y. Hui, Z. Su, and S. Guo, "Utility based data computing scheme to provide sensing service in Internet of Things," *IEEE Transactions on Emerging Topics in Computing,* vol. 7, no. 2, pp. 337-348, 1 April-June 2019.

[6] J. Ni, K. Zhang, Q. Xia, X. Lin and X. S. Shen, "Enabling Strong Privacy Preservation and Accurate Task Allocation for Mobile Crowdsensing," *IEEE Transactions on Mobile Computing,* vol. 19, no. 6, pp. 1317-1331, 1 June 2020.

[7] J. Ni, A. Zhang, X. Lin and X. S. Shen, "Security, Privacy, and Fairness in Fog-Based Vehicular Crowdsensing," *IEEE Communications Magazine,* vol. 55, no. 6, pp. 146-152, June 2017.

[8] K. Yang, K. Zhang, J. Ren and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE Communications Magazine,* vol. 53, no. 8, pp. 75-81, August 2015.

[9] I. Krontiris, M. Langheinrich and K. Shilton, "Trust and privacy in mobile experience sharing: future challenges and avenues for research," *IEEE Communications Magazine,* vol. 52, no. 8, pp. 50-55, August 2014.

[10] Sarhan, Q.I., "Internet of things: a survey of challenges and issues," *International Journal of Internet of Things and Cyber-Assurance,* Vol. 1, No. 1, pp.4075, 2018.

[11] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," *IEEE Internet of Things Journal,* vol. 1, no. 4, pp. 349-359, August 2014.

[12] I. Ud Din et al., "The Internet of Things: A Review of Enabled Technologies and Future Challenges," *IEEE Access,* vol. 7, pp. 7606-7640, 2019.

[13] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu and W. Zhao, "A Real-Time En-Route Route Guidance Decision Scheme for Transportation-Based Cyberphysical Systems," *IEEE Transactions on Vehicular Technology,* vol. 66, no. 3, pp. 2551-2566, March 2017.

[14] V. Vippalapalli and S. Ananthula, "Internet of things (IoT) based smart health care system," in *Proc. 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES),* Paralakhemundi, 2016, pp. 1229-1233.

[15] Serena Nicolazzo, Antonino Nocera, Domenico Ursino, Luca Virgili, "A privacy-preserving approach to prevent feature disclosure in an IoT scenario," *Future Generation Computer Systems,* vol. 105, pp. 502-519, April 2020.

[16] Y. Wang, H. Li, Y. Cui, Q. Guo and Q. Huang, "Survey on public key encryption with equality test," *Chinese Journal of Network and Information Security,* vol. 4, pp 13-22, 2018.

[17] H. Qin, H. Wang, X. Wei, L. Xue and L. Wu, "Privacy-Preserving Wildcards Pattern Matching Protocol for IoT Applications," *IEEE Access,* vol. 7, pp. 36094-36102, February 2019.

[18] Z. Zhou, D. Huang and Z. Wang, "Efficient Privacy-Preserving Ciphertext-Policy Attribute Based-Encryption and Broadcast Encryption," *IEEE Transactions on Computers,* vol. 64, no. 1, January 2015.

[19] H. Xiong, Y. Zhao, L. Peng, H. Zhang and K. H. Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Generation Computer Systems,* vol.97, pp. 453461, 2019.

[20] C. Y. Ng, Y. Mu and W. Susilo, "An identity-based broadcast encryption scheme for mobile ad hoc networks," *Journal of Telecommunications and Information Technology,* vol 1, pp. 24-29, 2006.

[21] C. I. Fan, L. Y. Huang and P. H. Ho, "Anonymous Multireceiver Identity-Based Encryption," *IEEE Transactions on Computers,* vol. 59, no. 9, September 2010.

[22] Y. Chen, W. Lin and Y. Tseng, "On common profile matching among multiparty users in mobile D2D social networks," 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, 2014, pp. 3396-3401, doi: 10.1109/WCNC.2014.6953125.

[23] Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. Commun. ACM 13, 7 (July 1970), 422426. DOI:https://doi.org/10.1145/362686.362692

[24] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren and W. Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," 2010 Proceedings IEEE INFOCOM, San Diego, CA, 2010, pp. 1-5, doi: 10.1109/INFCOM.2010.5462196.

[25] C. I. Fan, J. J. Huang, M. Z. Zhong, R. H. Hsu, W. T. Chen and J. Lee, "ReHand: Secure Region-Based Fast Handover With User Anonymity for Small Cell Networks in Mobile Communications," *IEEE Transactions on Information Forensics and Security,* vol. 15, pp. 927-942, July 2019.

[26] T. V. X. Phuong, G. Yang and W. Susilo, "Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions," *IEEE Transactions on Information Forensics and Security,* vol. 11, no. 1, pp. 35-45, January 2016.

[27] A. Menezes, An introduction to pairing-based cryptography, *Recent Trends in Cryptography Contemporary Mathematics,* vol. 477, pp. 4765, 2009.

[28] S. Sedghi, P. van Liesdonk, S. Nikova, P. Hartel, and W. Jonker, Searching keywords with wildcards on encrypted data, *Security and Cryptography for Networks (Lecture Notes in Computer Science),* vol. 6280. Berlin, Germany: Springer-Verlag, pp. 138153, 2010.

[29] K. Yang, K. Zhang, X. Jia, M. A. Hasan, Xuemin Shen, Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms, *Information Sciences,* vol. 387, pp. 116-131, 2017.

[30] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, Security for 4G and 5G Cellular Networks: A Survey of Existing Authentication and Privacy-preserving Schemes, *ArXiv Preprint*, arXiv:1708.04027, 2017.