Revocable Policy Hiding Bilateral Access Control Scheme With Equality Test for IIoT Environment

Junaid Hassan , Zhen Qin , *Senior Member, IEEE*, Muhammad Arslan Rauf , Muhammad Umar Aftab , Negalign Wake Hundera and Xinzhong Zhu , *Senior Member, IEEE*

***Abstract*—With the development of the Industrial Internet of Things (IIoT), the amount of data generated by industrial manufacturing equipment is growing rapidly, creating a critical need for secure and efficient cloud-based data sharing. While bilateral access control enables both data senders and receivers to define their own fine-grained access policies, existing schemes transmit these policies in plaintext, exposing sensitive operational metadata and creating security vulnerabilities. Furthermore, they lack essential functionalities for practical IIoT deployment, including efficient duplicate data detection, a robust user revocation mechanism, and computationally lightweight operations suitable for resource-constrained devices. To address these challenges, this paper proposes a novel policy-hiding, lightweight, and revocable bilateral access control scheme with equality test (RBAC-ET) for cloud-enabled IIoT systems. RBAC-ET is the first scheme to integrally combine five critical capabilities: first, fine-grained bilateral access control using Linear Secret Sharing Schemes (LSSS) supporting `AND`/`OR` logical operations. Second, policy confidentiality is achieved by encrypting access policies within the ciphertext to prevent the leakage of sensitive relationships. Third, the equality test functionality enables cloud servers to identify duplicate ciphertexts for efficient storage and processing without requiring decryption. Fourth, a proposed lightweight design is based on elliptic curve scalar multiplication operations instead of the more resource-intensive bilinear pairing operations. Fifth, a robust user revocation mechanism that ensures both forward and backward secrecy in dynamic industrial environments. Theoretical analysis and experimental results demonstrate that RBAC-ET offers superior security and functionality while maintaining computational efficiency comparable to that of state-of-the-art schemes, making it a scalable and practical solution for modern IIoT data-sharing applications.**

***Index Terms*—Fine-grained access control, bilateral access control, cloud computing, equality test, user revocation, policy hiding, attribute-based encryption, matchmaking encryption.**

This work was supported by the National Natural Science Foundation of China (62376252); Zhejiang Province Leading Geese Plan (2025C02025, 2025C01056); and Zhejiang Province Province-Land Synergy Program (2025SDXT004-3). (*Corresponding author: Xinzhong Zhu.*)

Junaid Hassan, Zhen Qin, and Muhammad Arslan Rauf are with the Network and Data Security Key Laboratory of Sichuan Province, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China, (e-mail: junaidjh2101@gmail.com, qinzhen@uestc.edu.cn, marslanrauf@hotmail.com)

Muhammad Umar Aftab is with the Department of Computer Science, National University of Computer and Emerging Science, Islamabad, Chiniot-Faisalabad Campus, 35400, Pakistan (e-mail:umar.aftab@nu.edu.pk)

Negalign Wake Hundera is with the Network and Data Security Key Laboratory of Sichuan Province, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with the School of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, China (e-mail:nigaccna21@uestc.edu.cn)

Xinzhong Zhu is with the School of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, China, AI Research Institute of Beijing Geekplus Technology Co., Ltd., Beijing, 100101, China, and also with the Research Institute of Hangzhou Artificial Intelligence, Zhejiang Normal University, Hangzhou, 311231, China (e-mail:zxz@zjnu.edu.cn)

I. Introduction

With the rapid advancement of wireless communication technologies, the industrial Internet of Things has emerged as a transformative paradigm [1]. This paradigm significantly enhances the connectivity between the internet and various physical objects within industrial environments. In recent years, IIoT has been integrated into numerous sectors, particularly in the manufacturing industry. In IIoT systems within industrial settings, smart devices are used to gather real-time operational data. Industrial IoT [2] upgrades traditional manufacturing processes to intelligent systems, thereby increasing operational efficiency and reducing production costs. To achieve these improvements, IIoT requires real-time online processing [3]. Consequently, IIoT devices must minimize communication latency to meet performance expectations. The rapid increase in IIoT data has created significant challenges in processing and sharing large amounts of real-time information [4]. Cloud computing is utilized to manage large amounts of IIoT data [5]. Nevertheless, cloud servers are typically "honest-but-curious," raising a critical issue in cloud data-sharing applications: how to preserve the user's private information.

Bilateral access control, an emerging security paradigm, is increasingly adopted to preserve privacy in cloud data-sharing applications [6]. Unlike the traditional one-sided access control scheme [7], bilateral access control allows both data senders and receivers to specify their own access preferences, enabling selective data sharing and acquisition. Furthermore, bilateral access control ensures that data can only be decrypted when the preferences of both parties are satisfied simultaneously. A naive approach to implementing bilateral access control is to perform twice one-sided access control. However, this naive approach inadvertently leaks preference-matching information, thereby compromising the fundamental privacy guarantees of bilateral access control. To address security challenges in the matching process, many access control schemes that preserve privacy have been proposed, such as attribute-based encryption (ABE) [8], [9], and identity-based encryption (IBE) [10]. However, some issues remain while constructing these secure data-sharing schemes in IIoT. For example, existing ABE and IBE schemes can provide access control on receivers but not on senders for data source authentication. Furthermore, most of these cryptographic schemes have high computational costs for encryption and decryption, which is not feasible for most IIoT devices due to their limited computational and storage resources. In ABE, pairing operations in encryption and decryption grow linearly with the number of attributes. Therefore, the high computational overhead of ABE is an inherent barrier to the large-scale implementation of ABE in the IIoT [11]. To reduce the ABE encryption cost, some studies [12], [13] have proposed an online/offline scheme for ABE. A portion of complex encryption is performed in the offline phase, while the rest is done in the online phase, making the encryption phase efficient.

Another fundamental technique for achieving privacy-preserving data matching in IIoT environments is the equality test, using a cryptographic trapdoor to compare data without revealing the underlying content [14]. In the IIoT environment, this functionality is profoundly beneficial. For instance, multiple factory machines may generate identical encrypted status reports. Instead of processing each one, the cloud server can use the equality test with the corresponding trapdoors to identify these duplicate ciphertexts and process the information only once, significantly improving system efficiency and reducing computational load. Because of its possible practical significance, it has attracted a lot of attention from researchers in a variety of IoT subdomains, including Wireless Body Area Network [15], Internet of Medical Things [16], cloud-storage-based network [17], and Internet of Vehicles [18]. Although the wide range of applications provided by the powerful capabilities of the equality test is impressive, the effectiveness of the equality test is usually constrained by the shortcomings in security and efficiency. Some of the current schemes fail to meet the necessary security requirements, and they do not provide assurances such as indistinguishability and resistance to forgery. Additionally, all these schemes utilize bilinear pairing operations, which are computationally expensive in the equality test. Thus, they are not suitable for use in sensitive or large-scale IoT systems.

In such a dynamic industrial environment, where personnel and device roles frequently change, a robust user revocation mechanism is not merely an auxiliary feature but a critical security requirement [19]. Any practical access control system must be able to immediately and effectively revoke a user's privileges when they leave the organization or their role changes [20]. An effective scheme ensures that all cryptographic artifacts associated with a revoked user, including their keys and any ability to authorize actions like equality tests, are rendered invalid. This prevents collusion attacks and ensures both forward and backward secrecy, upholding the strict security posture required in collaborative industrial systems.

This article considers the following challenges: *First, granting fine-grained bilateral access privileges over industrial data for factory managers and supervisors is essential* . On the one hand, factory supervisors may wish to grant access to specific categories of managers or assembly lines. On the other hand, managers can also define their access policies to specify which categories of operational data they want to receive. *Second, access policy confidentiality protection.* . Current schemes transmit access policies to the cloud in plaintext, exposing sensitive operational relationships and creating potential attack surfaces. Our scheme fundamentally enhances security by encrypting policy content within the ciphertext itself. *Third, enabling efficient equality testing for similar IIoT data.* In resource-limited IIoT environments, eliminating duplicate data is essential for reducing storage overhead and improving overall system performance. *Fourth, Robust Revocation Capability.* Frequent changes in personnel, devices, and access rights in IIoT environments demand effective revocation mechanisms. Without them, ex-employees or retired devices may retain access to sensitive data or compromise analytics by producing invalid trapdoors. *Fifth, lightweight encryption and decryption.* Lightweight encryption and decryption are essential in the IIoT due to limited computational resources, especially for devices with constrained processing power and memory in large-scale industrial deployments.

A. Contributions

To address the challenges mentioned above, this paper presents the first bilateral access control scheme that integrates equality testing functionality, addressing a critical gap in existing bilateral access control schemes. Furthermore, it provides lightweight encryption and decryption, as well as a robust user revocation mechanism, for the IIoT environment. Below, we outline our key contributions.

1) *Efficient equality testing:* This paper introduces the first bilateral access control scheme with integrated equality testing, allowing cloud servers to detect duplicate encrypted IIoT data via trapdoors without decryption. This approach overcomes the scalability limitations of ABE, making it practical for resource-constrained IIoT environments.
2) *Fine-grained access control:* Our scheme uses a linear secret sharing scheme to define matching policies based on the sets of attributes, supporting both `AND`/`OR` logical operations within the policies for fine-grained access control.
3) *Ensuring access policy confidentiality:* This paper introduces policy-hiding by encrypting access policies within ciphertexts to conceal sensitive operational relationships. Unlike existing schemes that transmit policies in plaintext, our approach eliminates these attack surfaces and prevents operational metadata leakage from curious cloud servers.
4) *Lightweight Encryption and Decryption:* Our RBAC-ET scheme achieves efficient encryption and decryption by utilizing elliptic curve scalar multiplication operations, which are 13.93 ms faster per operation than bilinear pairing operations [21], making it highly practical for deployment in IIoT environments.
5) *Robust revocation with forward/backward secrecy:* Our scheme presents a robust user revocation mechanism designed for dynamic industrial environments. It prevents collusion attacks and ensures forward/backward secrecy by invalidating all cryptographic artifacts of a revoked entity, including keys and equality test capabilities.

B. Organization

The following structure organizes this article. Related work is presented in II. The materials and methods used in this article are discussed in Section III. Section IV describes the system model, threat model, and security model of RBAC-ET. Workflow and concrete construction of RBAC-ET based on elliptic curves, along with correctness and a rigorous security proof, are provided in Section V. Section VI presents the experimental evaluation and theoretical complexity. The conclusion of this paper is found in Section VII.

II. Related Work

In this section, we describe the state-of-the-art works on matchmaking bilateral fine-grained access control for cloud-enabled IoT systems. A comparison of functionality among existing data sharing schemes is provided in Table I.

A. Equality-Test-Based Encryption Schemes

Zhou et al. [14] present a forward-secure identity-based encryption with equality test (FS-IBEET) scheme that utilizes time encoding. However, it incurs higher computational and storage costs. Zhang et al. [15] propose a pairing-free signcryption scheme with group equality testing for WBANs that reduces computational overhead but may not support complex authorization policies. Zheng et al. [16] introduce a revocable identity-based matchmaking encryption scheme with an equality test (RIBME-ET) for smart healthcare, which improves efficiency and provides bilateral access control, but with limitations in one-to-one communication. Li et al. [17] proposed a public-verifiable PKEET scheme that enables verification without private keys, and allows users to specify authorized testers for multi-cloud environments. However, it has a higher equation test overhead compared to some existing schemes. Hou et al. [18] developed a heterogeneous broadcast signcryption with equality test (HBSC-ET) scheme that supports cross-domain communication; however, it is vulnerable to key compromise without an update mechanism.

B. Identity-Based Matchmaking Encryption Schemes

To provide access control for both the sender and receiver simultaneously, an identity-based encryption scheme was proposed by Ateniese et al.[22]. Building on this, a certificateless pairing-free matchmaking (CL-ME) scheme was introduced by Chen et al.[23], which addresses data security and privacy problems in IoT and significantly reduces computational overhead. However, CL-ME requires receivers to retrieve ciphertexts one at a time, resulting in high computational costs. To address this, Yan et al.[24] introduced an identity-based proxy ME scheme that enables parallel ciphertext retrieval through multiple proxies. However, it requires all proxies to be honest. Since ME only supports one-to-one access, Sun et al.[25] proposed a security-enhanced ME scheme (PS-ME) that uses inner product encryption to allow multiple receivers to decrypt the same ciphertext. Bao et al.[26] presented a lightweight bilateral access control scheme (LFBA) for vehicle platoons, which supports policy hiding.

C. Attribute-Based Matchmaking Encryption Schemes

Zhang et al. [27] extended matchmaking IBE to matchmaking ABE, making ME suitable for distributed IoT scenarios and one-to-many communication. A new data-sharing scheme for cloud-fog-enabled systems (CFDS) was proposed by Xu et al. [28] that avoids the linkability of different ciphertexts using the rerandomization technique. Sun et al. [29] developed a scheme for IIoT healthcare that guarantees sender privacy with bilateral access control. Unfortunately, unlike ME, CFDS, and PBAC-FG, suffer from the issue of preference and matching information leakage. To overcome this issue, Wu et al. proposed a scheme [30] combined ME and ABE with a secret sharing scheme for AND/OR operations. Hu et al. [31] integrated identity-based and attribute-based encryption for bilateral policy verification. However, schemes [30] and [31] involve complex operations unsuitable for resource-constrained IIoT devices. Huang et al. [32] introduced an SGX-based framework for filtering encrypted data. This framework utilizes Intel Software Guard Extensions (SGX); however, it is not universally supported on all hardware platforms, and its dependence on a specific technology can be a limitation. Another scheme [33] proposed policy-hiding ABE with user revocation, but incurs computational overhead. Yao et al. [34] utilized blockchain and proxy re-encryption; however, they cannot resist collusion attacks.

D. Revocation-Based Encryption Schemes

Jiguo et al. [35] proposed a blockchain-aided multi-authority ABE scheme for Cloud IoT that enables anonymous traceable revocation, dynamic policy updating, and privacy protection via consensus nodes; however, it limits complete decentralization by requiring a central authority, lacks instant revocation flexibility, and imposes high storage overhead on non-revoked users for key updates. Yao et al. [36] designs a lightweight, revocable CP-ABE scheme with IPFS deduplication and constant-time updates for resource-constrained smart consumer electronics. However, it likely faces scalability limitations and increased latency due to blockchain dependencies, which may impact real-time performance in consumer electronics applications. Zhao et al. [37] propose a collusion-resistant CP-ABE scheme that efficiently supports user revocation through group management. However, this scheme has a limitation in that it slightly increases storage requirements compared to previous schemes. Xia et al. [19] developed an efficient revocable ABE scheme using FAME, but it relies on computationally heavy bilinear pairings. Hou et al. [20] proposes RBF-AC, a bilateral fine-grained access control scheme for VSNs that require predefined user numbers and leave physical vehicle capture attacks unresolved.

III. Materials and Methods

A. Preliminaries

Let p be a large prime and let $\mathbb{F}_p$ denote the finite field with p elements. $E/\mathbb{F}_p$ be a (non-singular) elliptic curve given by

$$E : y^2 = x^3 + ax + b \quad \text{over } \mathbb{F}_p,$$

TABLE I
FUNCTIONALITY COMPARISON OF RELATED DATA SHARING SCHEMES

Schemes	System architecture	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	Security model
[14]	Cloud Storage	●	○	○	●	○	–	◑	○	●	●	IND-CCA
[17]	Cloud Storage	●	○	○	●	○	○	○	○	●	○	IND-CPA
[19]	Cloud Storage	●	○	○	○	○	○	○	●	●	○	IND-CPA
[20]	Edge-cloud Storage	●	●	●	○	○	●	●	●	●	○	IND-CPA & EU-CMA
[22]	Cloud Storage	●	●	●	○	●	●	○	○	○	●	IND-CPA
[23]	Cloud Storage	●	●	●	○	●	●	●	○	○	○	IND-CPA & EU-CMA
[27]	Edge-cloud Storage	●	●	●	○	●	●	○	○	○	○	IND-CCA
[29]	Cloud Storage	●	●	●	○	○	●	●	○	●	○	IND-CPA & EU-CMA
[32]	Edge-cloud Storage	●	●	●	○	●	●	●	○	●	●	IND-CPA & EU-CMA
[33]	Cloud Storage	●	○	○	○	◑	●	●	●	●	○	IND-CPA
[35]	Cloud Storage	●	○	●	○	●	●	●	●	●	○	IND-CCA
[36]	Edge-cloud Storage	●	●	●	○	●	●	●	●	○	◑	IND-CCA
Ours	Cloud Storage	●	●	●	●	●	●	●	●	●	●	IND-CCA2 & OW-CCA2

F_1: Data confidentiality; F_2: Sender authenticity; F_3: Bilateral access control; F_4: Equality test; F_5: Policy-hiding; F_6: Collusion attack resistance; F_7: Lightweight encryption/decryption; F_8: User revocation; F_9: Outsourced Verification; F_{10}: High practicality; IND-CCA: Indistinguishability under chosen ciphertext attacks; OW-CCA2: One-way against chosen ciphertext attacks; IND-CPA: Indistinguishability under chosen plaintext attack; EU-CMA: Existential unforgeability under chosen message attacks; "●": Fully supported; "◑": Partially supported; "○": Not supported; "–": Not applicable.

with $4a^3+27b^2 \not\equiv 0 \pmod p$. Let $E(\mathbb{F}_p)$ denote the group of $\mathbb{F}_p$-rational points on E with identity $\mathcal{O}$ (the point at infinity).

Fix a cyclic subgroup $\mathbb{G} \subseteq E(\mathbb{F}_p)$ of prime order q, and let $g \in \mathbb{G}$ be a generator; we write $\mathbb{G} = \langle g \rangle$ and $|\mathbb{G}| = q$. Let the cofactor be

$$h = \frac{|E(\mathbb{F}_p|)}{q}.$$

All scalar operations are taken in the finite field $\mathbb{Z}_q$. For $k \in \mathbb{Z}_q$ and $X \in \mathbb{G}$, scalar multiplication is written additively as

$$k \cdot X = \underbrace{X + \cdots + X}_{k \text{ times}}$$

, and in particular $X = x \cdot g$ for a unique $x \in \mathbb{Z}_q$.

Elliptic curve cryptography relies on certain well-known hard problems:

1) Elliptic Curve Discrete Logarithm Problem (ECDLP): Given $A, B \in \mathbb{G}$, determine the integer $\alpha \in \mathbb{Z}_q$ such that $B = \alpha A$. Under the Elliptic Curve Discrete Logarithm Problem (ECDLP), computing α is computationally infeasible.
2) Elliptic Curve Decisional Diffie–Hellman (ECDDH) Problem: Given random elements $\alpha, \beta, z \in \mathbb{Z}_q^*$, it is computationally infeasible to distinguish between the tuples $(\alpha g, \beta g, \alpha\beta g)$ and $(\alpha g, \beta g, z g)$ in group $\mathbb{G}$. In other words, determining whether $\alpha\beta g = z g$ is a hard decisional problem.

Both problems are considered computationally infeasible to solve efficiently, forming the foundation for the security of our proposed scheme.

B. Linear Secret-Sharing Schemes

Definition 1. Let define $\mathbb{Z}_q$ as a finite field and Π as a secret sharing scheme having a domain of secrets such as $S \subseteq \mathbb{Z}_q$, then we can say that Π is called linear over a set of parties $\mathcal{P}$ if:

1) The share of each party is a vector over $\mathbb{Z}_q$
2) $\mathbb{W} \in \mathbb{Z}_q^{l_{\mathbb{W}} \times n_{\mathbb{W}}}$ is a matrix that is used to generate the secret shares with n columns and l rows. Where the columns vector is considered as $\vec{y} = (\beta, \tau_\sigma, y_2, \ldots, y_{n_{\mathbb{W}}})$. For all $i \in [l_{\mathbb{W}}]$, the party's name is labelled with the ith row of the matrix $\mathbb{W}$, as $k_i \in \mathcal{P}$ and $\beta \in \mathbb{Z}_q$ is the secret that is to be shared. A vector with the secret β of l shares is denoted as $\mathbb{W}\vec{y}$. Here, a party k_i belongs to a share $\mathbb{W}_i y_i$, where y_i denotes the ith term of the vector $\vec{y}$ and $\mathbb{W}_i$ is the ith row of $\mathbb{W}$.

C. Formal Definition of Scheme

A lightweight, revocable, bilateral access control scheme with an equality test involves three main components: a key generation center (KGC), end devices (EDs), and a cloud service provider (CSP). The KGC interacts with senders and receivers, who have respective attribute universes (θ_{snd} and θ_{rcv}). These components collaborate to run the following eleven algorithms:

1) *Setup*$(1^\psi) \rightarrow (msk, mpk, pp)$: KGC uses the security parameter ψ to create some public parameters pp as an output for encryption and decryption, a master public key mpk, and a master secret key msk.
2) *Register*$(\mathcal{S}/\mathcal{R}) \rightarrow Rk_u$: The cloud server runs this algorithm. It takes the user's set of attributes as input, generates the user registration key Rk_u, and sends it to all the registered users.
3) *Encryption Key Generation*$(\mathcal{S}, msk) \rightarrow ek$: KGC executes the algorithm, taking msk and $\mathcal{S}$, the set of the sender's attributes as input, and outputs ek.
4) *Decryption Key Generation*$(\mathcal{R}, msk) \rightarrow dk$: KGC executes this algorithm, taking msk and $\mathcal{R}$, the set of the receiver's attributes as input, and generates dk as output.
5) *Encryption*$(\mathcal{S}, \mathbb{R}, \mathcal{M}, ek, Rk_{snd,n}, mpk) \rightarrow (C)$: This algorithm takes the encryption key ek, registration key of

sender $Rk_{snd,n}$, master public key mpk, as well as the attribute sets for the sender's ($\mathcal{S}$), access policy for the the receiver's ($\mathbb{R}$), and the message $\mathcal{M}$, producing ciphertext C as output.

6) *IntTag*$(\mathbb{S}, Rk_{rcv,h}) \rightarrow (Tag)$: The receiver runs this algorithm, taking the sender's access policy ($\mathbb{S}$) and the receiver's registration key $Rk_{rcv,h}$ as input. It then generates a Tag and sends it to the cloud server for verification.
7) *Verification*$(Tag, C) \rightarrow \{0, 1\}$: The cloud server runs this algorithm and takes the Tag generated by the receiver and the ciphertext C. It verifies the encrypted message C against the access policy of the sender's $\mathbb{S}$ and their set of attributes. It returns a success bit 1 if the verification is successful; otherwise, it returns a failure bit 0.
8) *Trapdoor:* $(r, Rk_u) \rightarrow TD$: This algorithm is run by the users. It selects a fresh random value $r \in \mathbb{Z}_q^*$, binds it with its registration key Rk_u, constructs the trapdoor TD, and returns the corresponding trapdoor.
9) *Test:* $(C, TD) \rightarrow \{0, 1\}$: The cloud server runs this algorithm by taking two pairs of ciphertext-trapdoor (C_A, TD_A) and (C_B, TD_B) from two different senders and performing the equality test. If the equality holds, it returns an output of 1; Otherwise, it returns 0.
10) *Decryption*$(dk, C) \rightarrow \mathcal{M}$: This algorithm uses the ciphertext C and decryption key dk as input. If the access policy of the receiver matches the required conditions $\mathcal{R} \models \mathbb{R}$, then it decrypts the message ($\mathcal{M}$); otherwise, it generates an error $\perp$.
11) *Rev*$(att_m, att_n) \rightarrow (uk_{1,u}, uk_{2,u}, Rk_{u,m}, ek_m, dk_m)$ The KGC and the CSP mutually execute this algorithm. KGC takes the user's old and new set of attributes (att_n,att_m) as an input and outputs two updated keys $uk_{1,u}$ and $uk_{2,u}$ and sends them to the cloud server. KGC also updates the partial encryption and decryption keys $(ek_{1,m}, dk_{1,m})$ and sends them to the related users. The cloud server uses $uk_{1,u}$ and $uk_{2,u}$ to update the user's registration keys $Rk_{u,m}$ and ciphertext components in the system.

IV. Overview of the Scheme

A. System Model

Fig. 4 illustrates our cloud-based IIoT system involving three types of entities: the cloud service provider (CSP), key generation center (KGC), and end devices (EDs) (receiver, sender). The KGC has a central role. It generates the system's public parameters, including the master secret key and master public key (see ①). The cloud server generates the user's registration keys (see ②). KGC creates encryption keys associated with an authenticated set of attributes for the sender (see ③), and issues decryption keys based on the authenticated set of attributes for the receiver (see ④). Senders collect sensitive industrial data from their IoT devices, then encrypt and embed the receiver access policy $\mathbb{R}$ in the ciphertext. The sender also generates a trapdoor using their registration key and sends it to the cloud along with the encrypted data (see ⑤). Receivers select the sender's access policy $\mathbb{S}$, embed it with their registration key, and create a Tag for their access policies. Then send it to the cloud (see ⑥). After delegating these policies to the cloud for verification. The cloud first verifies Tag–ciphertext matching without learning either policy; only after a successful match (see ⑦), it runs an equality test across ciphertexts belonging to the same senders (see ⑧). If the new ciphertext matches an already-stored message, the cloud discards the latest ciphertext and retains the older canonical ciphertext, thereby achieving deduplication. Once verification and equality tests are completed, authorized receivers can retrieve and decrypt the sensitive industrial data from the cloud to access the original information (see ⑨).

B. Threat Model

In this model, the KGC is considered a trusted authority responsible for system initialization and distributing all necessary public parameters, including the master public key and the encryption and decryption keys required for secure communication via the secure channel. However, both the sender and receiver might act maliciously. As the sender, an ED could attempt to impersonate others to produce authorized ciphertexts intended for someone else. At the same time, an ED (as the recipient) might try to decrypt sensitive industrial data to which they do not have authorized access. CSP is an "honest-but-curious" entity that securely stores shared data and performs matching. It can also attempt to eavesdrop and learn something about confidential industrial data. The encrypted data stays confidential, so unauthorized users can't access sensitive information. The system addresses the following security requirements, taking into account these potential threats.

1) *Confidentiality of data:* Industrial data collected by IoT devices is often sensitive and private. It commonly contains employee information, production schedules, operational details, trade secrets, and confidential data about equipment and facilities. Therefore, it must be encrypted and stored securely to prevent unauthorized access.
2) *Correctness:* Our RBAC-ET scheme is correct if two requirements are met. First, any legitimate user with the appropriate decryption key can successfully recover the plaintext from a ciphertext. Second, with the given valid trapdoors, the equality test always returns 1 whenever two ciphertexts are encrypted by the same message.
3) *Integrity of the access control policies:* Malicious actors may attempt to modify the access control policies to gain unauthorized access to data. The system must ensure that the access policies cannot be modified or tampered with.

C. Security Model

This section describes two security models: indistinguishability under a chosen ciphertext attack (IND-CCA2) and one-way against chosen ciphertext attack (OW-CCA2). The IND-CCA2 security definition ensures that the constructed cloud-enabled industrial IoT system can resist chosen ciphertext attacks, including adversarial behaviors such as eavesdropping and collusion attacks. The OW-CCA2 security ensures that the RBAC-ET scheme can resist chosen-ciphertext attacks, ciphertext-linking attacks, and partial-information-leakage attacks via equality testing. Our scheme considers two types of adversaries, namely:

1) *Type-I adversary (external):* $\mathcal{A}_1$, legitimate sender and receiver that does not possess any valid system secret keys or credentials.
2) *Type-II adversary (internal):* $\mathcal{A}_2$, a curious key generation center (KGC) or cloud server holding system master secrets or user registration keys, but prohibited from colluding with entities that already satisfy the access policy or with its counterpart authority.

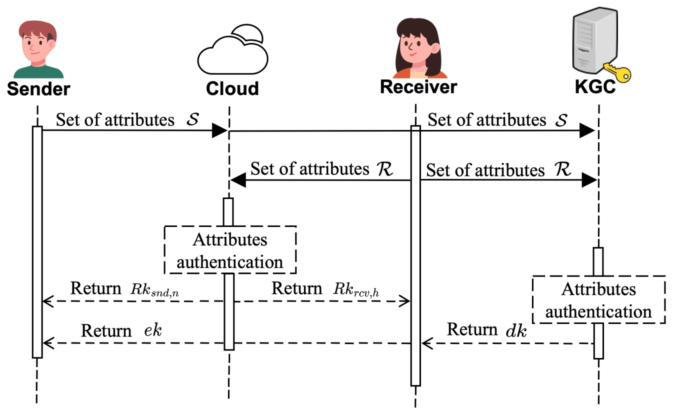
Definition 2. Let $\mathcal{O}$ represent different oracles: $\mathcal{O}_{EKGen}$, $\mathcal{O}_{DKGen}$, $\mathcal{O}_{Register}$, $\mathcal{O}_{Hash}$, $\mathcal{O}_{Enc}$, $\mathcal{O}_{Dec}$. We conducted the following experiment to implement the RBAC-ET scheme and achieve IND-CCA2 security.

```
Exp $^{\text{IND-CCA2}}_{\mathcal{RBAC}\text{-}\mathcal{ET},\mathcal{A}}(1^\psi)$
  $(\mathcal{S}^*, \mathbb{R}^*) \leftarrow \mathcal{A}(1^\psi)$;
  $(msk, mpk, pp) \leftarrow Setup(1^\psi)$;
  $(\mathcal{M}_0, \mathcal{M}_1) \leftarrow \mathcal{A}^{\mathcal{O}}$;   $f \in \{0,1\}$;
  $Rk^*_{snd,n} \leftarrow Register(\mathcal{S}^*)$;
  $ek^* \leftarrow EKGen(\mathcal{S}^*, msk)$;
  $C^* \leftarrow Enc(\mathcal{S}^*, \mathbb{R}^*, Rk^*_{snd,n}, ek^*, \mathcal{M}_f)$;
  $f' \leftarrow \mathcal{A}^{\mathcal{O}}(C^*)$;
  return 1 iff $f = f'$

Oracle $\mathcal{O}_{H_1}(att_{snd,n})$
  $\mathbb{G} \leftarrow H_1(att_{snd,n})$;
  return $\mathbb{G}$

Oracle $\mathcal{O}_{H_2}(att_{rcv,h})$
  $\mathbb{G} \leftarrow H_2(att_{rcv,h})$;
  return $\mathbb{G}$

Oracle $\mathcal{O}_{H_3}(\mathcal{M})$
  $\mathbb{Z}_q^* \leftarrow H_3(\mathcal{M})$;
  return $\mathbb{Z}_q^*$

Oracle $\mathcal{O}_{Register}(\mathcal{S}/\mathcal{R})$
  $Rk_u \leftarrow Register(\mathcal{S}/\mathcal{R})$;
  return $Rk_u$

Oracle $\mathcal{O}_{EKGen}(\mathcal{S}, msk)$
  Restriction: $\mathcal{S} \neq \mathcal{S}^*$
  $ek \leftarrow EKGen(\mathcal{S}, msk)$;
  return $ek$

Oracle $\mathcal{O}_{DKGen}(\mathcal{R}, msk)$
  Restriction: $\mathcal{R} \not\models \mathbb{R}^*$
  $dk \leftarrow DKGen(\mathcal{R}, msk)$;
  return $dk$

Oracle $\mathcal{O}_{Enc}(\mathcal{S}, \mathbb{R}, Rk_{snd,n}, ek, \mathcal{M})$
  $C \leftarrow Enc(\mathcal{S}, \mathbb{R}, Rk_{snd,n}, ek, \mathcal{M})$;
  return $C$

Oracle $\mathcal{O}_{Dec}(dk, C)$
  Restriction: $C \neq C^*$
  $\mathcal{M} \leftarrow Dec(dk, C)$;
  return $\mathcal{M}$
```

A RBAC-ET scheme is said to be IND-CCA2 secure if for an adversary $\mathcal{A}$, who can query $\mathcal{O}_{EKGen}$, $\mathcal{O}_{DKGen}$, $\mathcal{O}_{Register}$, $\mathcal{O}_{Hash}$, $\mathcal{O}_{Enc}$, $\mathcal{O}_{Dec}$ respectively, in a probabilistic polynomial time τ, the following advantage is negligible.

$$\text{Adv}^{\text{IND-CCA2}}_{\mathcal{RBAC}-\mathcal{ET},\mathcal{A}}(1^\psi) = \left|\Pr\left[\text{Exp}^{\text{IND-CCA2}}_{\mathcal{RBAC}-\mathcal{ET},\mathcal{A}}(1^\psi) = 1\right] - \tfrac{1}{2}\right|$$

Definition 3. Let $\mathcal{O}$ present different oracles: $\mathcal{O}_{EKGen}$, $\mathcal{O}_{DKGen}$, $\mathcal{O}_{Register}$, $\mathcal{O}_{Hash}$, $\mathcal{O}_{Enc}$, $\mathcal{O}_{Dec}$, $\mathcal{O}_{Trap}$. We performed the following experiment for the RBAC-ET scheme to achieve OW-CCA2



Sender
Cloud
Receiver
KGC
Set of attributes $\mathcal{S}$
Set of attributes $\mathcal{S}$
Set of attributes $\mathcal{R}$
Set of attributes $\mathcal{R}$
Attributes authentication
Return $Rk_{snd,n}$
Return $Rk_{rcv,h}$
Attributes authentication
Return ek
Return dk

Fig. 1. Key distribution in RBAC-ET

```
Exp $^{\text{OW-CCA2}}_{\mathcal{RBAC}\text{-}\mathcal{ET},\mathcal{A}}(1^\psi)$
  $(\mathcal{S}^*, \mathbb{R}^*) \leftarrow \mathcal{A}(1^\psi)$;
  $(msk, mpk, pp) \leftarrow Setup(1^\psi)$;
  $(\mathcal{M}_0, \mathcal{M}_1) \leftarrow \mathcal{A}^{\mathcal{O}}$;   $f \in \{0,1\}$;
  $Rk^*_{snd,n} \leftarrow Register(\mathcal{S}^*)$;
  $ek^* \leftarrow EKGen(\mathcal{S}^*, msk)$;
  $C^* \leftarrow Enc(\mathcal{S}^*, \mathbb{R}^*, Rk^*_{snd,n}, ek^*, \mathcal{M}_f)$;
  $f' \leftarrow \mathcal{A}^{\mathcal{O}}(C^*)$;
  return 1 iff $f = f'$

Oracle $\mathcal{O}_{H_1}(att_{snd,n})$
  $\mathbb{G} \leftarrow H_1(att_{snd,n})$;
  return $\mathbb{G}$

Oracle $\mathcal{O}_{H_2}(att_{rcv,h})$
  $\mathbb{G} \leftarrow H_2(att_{rcv,h})$;
  return $\mathbb{G}$

Oracle $\mathcal{O}_{H_3}(\mathcal{M})$
  $\mathbb{Z}_q^* \leftarrow H_3(\mathcal{M})$;
  return $\mathbb{Z}_q^*$

Oracle $\mathcal{O}_{Register}(\mathcal{S}/\mathcal{R})$
  $Rk_u \leftarrow Register(\mathcal{S}/\mathcal{R})$;
  return $Rk_u$

Oracle $\mathcal{O}_{EKGen}(\mathcal{S}, msk)$
  Restriction: $\mathcal{S} \neq \mathcal{S}^*$
  $ek \leftarrow EKGen(\mathcal{S}, msk)$;
  return $ek$

Oracle $\mathcal{O}_{DKGen}(\mathcal{R}, msk)$
  Restriction: $\mathcal{R} \not\models \mathbb{R}^*$
  $dk \leftarrow DKGen(\mathcal{R}, msk)$;
  return $dk$

Oracle $\mathcal{O}_{Enc}(\mathcal{S}, \mathbb{R}, Rk_{snd,n}, ek, \mathcal{M})$
  $C \leftarrow Enc(\mathcal{S}, \mathbb{R}, Rk_{snd,n}, ek, \mathcal{M})$;
  return $C$

Oracle $\mathcal{O}_{Trap}(C)$
  Restriction: $C \notin C^*$
  $TD \leftarrow Trapdoor(C)$;
  return $TD$

Oracle $\mathcal{O}_{Dec}(dk, C)$
  Restriction: $C \notin C^*$
  $\mathcal{M} \leftarrow Dec(dk, C)$;
  return $\mathcal{M}$
```

A RBAC-ET scheme is said to be OW-CCA2 secure if for an adversary $\mathcal{A}$, who can query $\mathcal{O}_{EKGen}$, $\mathcal{O}_{DKGen}$, $\mathcal{O}_{Register}$, $\mathcal{O}_{Hash}$, $\mathcal{O}_{Enc}$, $\mathcal{O}_{Dec}$, $\mathcal{O}_{Trap}$ respectively, in a probabilistic polynomial time τ, the following advantage is negligible.

$$\text{Adv}^{\text{OW-CCA2}}_{\mathcal{RBAC}-\mathcal{ET},\mathcal{A}}(1^\psi) = \left|\Pr\left[\text{Exp}^{\text{OW-CCA2}}_{\mathcal{RBAC}-\mathcal{ET},\mathcal{A}}(1^\psi) = 1\right] - \tfrac{1}{2}\right|$$

V. Proposed RBAC-ET

This section provides a brief overview of our RBAC-ET scheme and illustrates its implementation. Finally, we conclude with security proofs that demonstrate our scheme is secure and robust.

A. Notations

Let ψ denote the security parameter, and we define $\mathbb{Z}_q$ as a finite field of order q. The sender's attributes universe is represented by θ_{snd}, while the receiver's attributes universe is represented by θ_{rcv}. We define $\mathcal{S}$ as the sender's attribute set and $\mathcal{R}$ as the receiver's attribute set. The receiver's access policy is denoted by $\mathbb{R}$, and the sender's access policy is denoted by $\mathbb{S}$.

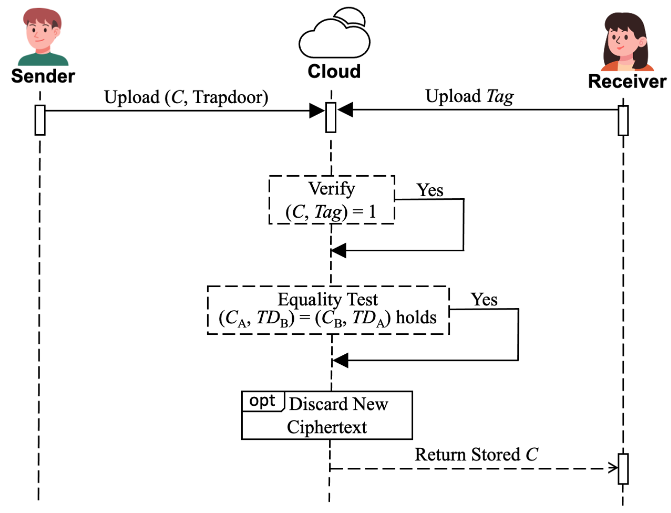


Fig. 2. Flow diagram of matching and equality test in RBAC-ET

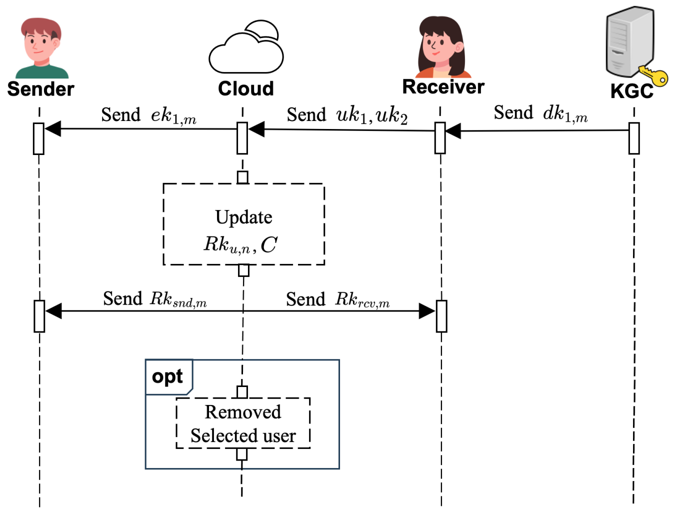


Fig. 3. Flow diagram of user revocation in RBAC-ET

B. Workflow of RBAC-ET

A revocable bilateral access control scheme with an equality test for the IIoT system consists of three primary phases: 1) key setup and distribution; 2) verification and equality test; 3) user revocation.

1) *Setup and Key Distribution:* In this phase, as shown in Fig. 1, the sender and receiver first submit their attribute sets $\mathcal{S}$ and $\mathcal{R}$ to the key generation center (KGC), which verifies them and generates an encryption key $ek_{\text{snd},n}$ for the sender and a decryption key $dk_{\text{rcv},h}$ for the receiver. The same attribute sets $\mathcal{S}$ and $\mathcal{R}$ are then sent to the cloud server. After the authentication of the attributes, the cloud returns the corresponding registration keys $Rk_{\text{snd},n}$ and $Rk_{\text{rcv},h}$ to the respective parties.
2) *Matching and Equality Test:* As shown in Fig. 2, the sender encrypts the collected IIoT data and embeds the receiver's access policy $\mathbb{R}$ inside the ciphertext before uploading it to the cloud. In contrast, the receiver independently chooses a sender access policy $\mathbb{S}$, binds it with a registration key to form a Tag, and submits this Tag to the cloud. The cloud first verifies that the Tag and the ciphertext match without revealing either policy, confirming that the sender and receiver satisfy each other's constraints. Only after a successful match does it perform an equality test across ciphertexts belonging to the same senders. If the new ciphertext is identical to an existing one, the cloud discards the latest ciphertext. It retains the older canonical ciphertext, enabling deduplication and preventing redundant storage.
3) *User Revocation:* As depicted in Fig. 3, Upon user attributes revocation or role changes, the KGC and cloud jointly update the corresponding keys: the KGC derives two update keys (uk_1, uk_2) for the cloud and issues partial keys $(ek_{1,m}, dk_{1,m})$ to affected senders and receivers; the cloud uses uk_1 to update registration keys to $Rk_{u,m}$ and send it to the affected user's, with uk_1 and uk_2, cloud updates the corresponding ciphertext components.

C. Concrete construction of RBAC-ET

$Setup(1^{\psi}) \rightarrow (msk, mpk, pp)$: KGC uses the security parameter ψ as input and does the following steps to generate some public parameters pp. It defines an elliptic curve $EC/\mathbb{F}_p$ over a prime finite field $\mathbb{F}_p$. Let g be the generator point of a cyclic group $\mathbb{G}$ over $\mathbb{F}_p$ with the order q. For the master secret key KGC, randomly picks a number $a \in \mathbb{Z}_q$ and calculates $\eta = a \cdot g$ as its master public key. It also picks three collision-resistant hash functions as $\mathcal{H}_1 : \theta_{snd} \rightarrow \mathbb{G}, \mathcal{H}_2 : \theta_{rcv} \rightarrow \mathbb{G}, \mathcal{H}_3 : \mathbb{G} \rightarrow \mathbb{Z}_q^*$. Finally, some public parameters are publicly announced for encryption and decryption, such as $pp = \{EC, \mathbb{G}, \mathbb{Z}_q^*, \eta, g, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$.

$Register(\mathcal{S}/\mathcal{R}) \rightarrow (Rk_u)$: Cloud server runs this algorithm, and on input a user attribute set $\mathcal{S}/\mathcal{R}$, this algorithm first authenticates $\mathcal{S}/\mathcal{R}$. It then generates a unique user identifier $\Theta_u \in \mathbb{Z}_q^*$ and associates it with the user attribute set. The algorithm outputs a user registration key Rk_u as

$$Rk_{snd,n} = \Theta_u \cdot \mathcal{H}_1(att_{snd,n}) + \Theta_u \cdot g$$

$$Rk_{rcv,h} = \Theta_u \cdot \mathcal{H}_2(att_{rcv,h}) + \Theta_u \cdot g$$

And send it to all the registered users.

$EKGen(\mathcal{S}, msk) \rightarrow (ek)$: This algorithm parses the set of sender's attributes, such as $\mathcal{S} = \{att_{snd,1}, att_{snd,2}, \ldots, att_{snd,b}\}$. For $n \in [b]$, a number is randomly picked $x \in \mathbb{Z}_q^*$, and calculates partial encryption keys as

$$ek_{1,n} = \mathcal{H}_1(att_{snd,n}) + ax \cdot g, \qquad ek_2 = a \cdot g$$

and finally, it returns the full encryption key $ek = (\{ek_{1,n}\}_{n\in[b]}, ek_2)$.

$DKGen(\mathcal{R}, msk) \rightarrow (dk)$: This algorithm parses the set of receiver's attributes, such as $\mathcal{R} = \{att_{rcv,1}, att_{rcv,2}, \ldots, att_{rcv,l}\}$. For $h \in [l]$, a number is randomly picked $\gamma \in \mathbb{Z}_q^*$, and calculates partial decryption keys as

$$dk_{1,h} = \mathcal{H}_2(att_{rcv,h}) + a\gamma \cdot g, \qquad dk_2 = a\gamma \cdot g + a \cdot g$$

and finally, it returns the full decryption key $dk = (\{dk_{1,h}\}_{h\in[l]}, dk_2)$.

$Enc(\mathcal{S}, \mathbb{R}, \mathcal{M}, ek, Rk_{snd,n}, mpk) \rightarrow (C)$: This algorithm randomly picks a number $k \in \mathbb{Z}_q$, and parses receiver access control structure $\mathbb{R} = (\mathbb{W}, \Omega)$, where $\mathbb{W} \in {\mathbb{Z}_q}^{l_{\mathbb{W}} \times n_{\mathbb{W}}}$ is a matrix and $\Omega : [l_{\mathbb{W}}] \rightarrow \theta_{rcv}$ is a mapping function. After that, it randomly picks $\vec{x} = (k, x_2, \ldots, x_{n_{\mathbb{W}}})^{\perp} \in \mathbb{Z}_q^{n_{\mathbb{W}} \times 1}$ and computes $\vec{\delta} = (\delta_1, \delta_2, \ldots, \delta_{l_{\mathbb{W}}}) = \mathbb{W}\vec{x}$. This algorithm also parses the set of sender's attributes $\mathcal{S} = \{att_{snd,1}, att_{snd,2}, \ldots, att_{snd,b}\}$. Then, it picks some random numbers such as $d, s, r, L, \vartheta \in \mathbb{Z}_q^*$, and takes a message $\mathcal{M}$ as an input where $\mathcal{M} \in \mathbb{G}$ and calculates $\Delta = (d + L) \cdot g$, and $m = \mathcal{H}_3(\mathcal{M})$. Let H be the receiver set given as $H = \{h | h \in [l_{\mathbb{W}}], \Omega(h) \in \mathcal{R}\}$, and finally, choose $\{\epsilon_h\}_{h\in H}$ such that $\sum_{h\in H} \epsilon_h \mathbb{W}_h = (1, 0, \ldots, 0)$. Where h is the index of the attribute $\Omega(h)$ in $\mathcal{R}$ such as $\Omega(h) = att_{rcv,h}$. For $n \in [b]$, and $h \in [l_{\mathbb{W}}]$ it calculates.

$$C_0 = d+L, \quad C_1 = \mathcal{M}+k\cdot g, \quad C_{2,h} = \epsilon_h \delta_h \cdot g + k\cdot\eta + ek_2,$$

$$C_{3,n} = \mathcal{H}_1(att_{snd,n}) + ek_{1,n} + ek_2$$

$$C_{4,h} = k \cdot \eta + \mathcal{H}_2(\Omega(h))$$

$$C_{5,n} = ek_{1,n} + ek_2 + Rk_{snd,n}, \quad C_6 = s \cdot \vartheta$$

$$C_7 = r \cdot m + s \cdot \vartheta$$

Finally, the ciphertext is

$$C = (C_0, C_1, C_{2,h}, C_{3,n}, C_{4,h}, C_{5,n}, C_6, C_7, \Delta).$$

$IntTag(\mathbb{S}, Rk_{rcv,h}) \rightarrow (Tag)$: Receiver runs this algorithm and parses the sender access structure $\mathbb{S} = (\mathbb{B}, \Lambda)$, where $\mathbb{B} \in \mathbb{Z}_q^{l_{\mathbb{B}} \times n_{\mathbb{B}}}$ is a matrix and $\Lambda : [l_{\mathbb{B}}] \rightarrow \theta_{snd}$ is a mapping function. Then it picks a vector $\vec{d} = (1, d_1, d_2, \ldots, d_{n_{\mathbb{B}}})^{\perp} \in \mathbb{Z}_q^{n_{\mathbb{B}} \times 1}$ and calculates:

$$\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{l_{\mathbb{B}}}) = \mathbb{B}\vec{d}.$$

It chooses $\{t_n\}_{n\in[N]}$. Where N is the sender's attribute set represented as $N = \{n | n \in [l_{\mathbb{B}}], \Lambda(n) \in \mathcal{S}\}$, and calculates, $\sum_{n\in N} t_n \mathbb{B}_n = (1, 0, \ldots, 0)$, then it picks some random numbers such as $\chi, \varrho \in \mathbb{Z}_q^*$. Finally, for $n \in [l_{\mathbb{B}}]$ it calculate a tag such as

$$X_n = (\chi \cdot \varrho)\mathcal{H}_1(\Lambda(n))$$

$$Y = \chi \cdot \varrho$$

$$Z_n = t_n \lambda_n \cdot g + Rk_{rcv,h}$$

$Tag \leftarrow (\{X_n, Z_n\}_{\in[l_{\mathbb{B}}]}, Y)$ and send it to the cloud for verification.

$Ver(Tag, C) \rightarrow (0, 1)$: This algorithm first remove the sender registration key $Rk_{snd,n}$ and the receiver registration key $Rk_{rcv,h}$ by computing $C'_{5,n} = C_{5,n} - Rk_{snd,n}$ and $Z'_n = Z_n - Rk_{rcv,h}$. If the following equation is valid, then it returns 1.

$$C_0 \cdot \sum_{n\in N} (Y \cdot C_{3,n} - X_n - Y \cdot C'_{5,n} + Z'_n) = \Delta$$

Otherwise, the algorithm will abort and return 0.

$Trapdoor : (r, Rk_u) \rightarrow (TD)$: The trapdoors for users A and B are calculated by this algorithm, as follows:

$$TD_A = r_A \cdot g + Rk_A \quad \text{and} \quad TD_B = r_B \cdot g + Rk_B,$$

Where r_A and r_B are the same numbers chosen during encryption by A and B, respectively.

$Test : (C, TD) \rightarrow \{0, 1\}$: For the given (CT_A, TD_A) and (CT_B, TD_B). The entity ET runs this algorithm as follows:

1) Compute: $Q_A = C_{7,A} - C_{6,A}$ and $Q_B = C_{7,B} - C_{6,B}$.
2) Compute: $TD'_A = TD_A - Rk_A$ and $TD'_B = TD_B - Rk_B$
3) ET checks if $Q_A \cdot TD'_B = Q_B \cdot TD'_A$ holds. If the equivalence holds, then the server will return 1. If not, the server will return 0.

$Dec(dk, C) \rightarrow (\mathcal{M})$: The receiver will run this algorithm to obtain the message $\mathcal{M}$ by computing the following equation:

$$C_1 - \sum_{h\in H} (dk_{1,h} + C_{2,h} - dk_2 - C_{4,h}) = \mathcal{M}$$

In this algorithm, the access control verification is checked by the decryption equation. In this equation, if the receiver has a valid set of attributes $\mathcal{H}_2(att_{rcv,h})$ that matches the receiver's access control policy $\mathcal{H}_2(\Omega(h))$ made by the sender, then the receiver can decrypt the ciphertext; otherwise, not. If satisfied, the receiver can get a message $\mathcal{M}$ from this algorithm.

$Rev(att_m, att_n) \rightarrow (uk_{1,u}, uk_{2,u}, Rk_{u,m}, ek_m, dk_m)$ KGC and CSP jointly run this algorithm and take a new attribute att_m as an input. This algorithm generate two update keys as $uk_{1,snd} = \mathcal{H}_1(att_{snd,m}) - \mathcal{H}_1(att_{snd,n})$ for sender, $uk_{1,rcv} = \mathcal{H}_2(att_{rcv,m}) - \mathcal{H}_2(att_{rcv,h})$ for receiver, and second key as $uk_2 = ek_{1,m} - ek_{1,n}$. It then updates the relevant partial encryption key as $ek_{1,m} = ek_{1,n} + uk_2$ and partial decryption key as $dk_{1,m} = dk_{1,n} + uk_{1,rcv}$, distributing them to the affected senders and receivers. The KGC sends uk_1 and uk_2 to the cloud server through a secure channel. With the uk_1, the cloud server updates the related registration key as $Rk_{u,m} = Rk_{u,n} + \Theta_u \cdot uk_1$ and sends it to the relevant users. With the uk_1 and uk_2, the cloud server updates the corresponding ciphertext components $C_{3,m} = C_{3,n} + uk_1 + uk_2, C_{5,m} = C_{5,n} + uk_2 + Rk_{u,m} - Rk_{u,n}$.

D. Correctness

If $\mathcal{S} \models \mathbb{S}$ holds, then a ciphertext can pass the verification algorithm if the ciphertext confirms the verification requirements.

$$C_0 \cdot \sum_{n\in N} (Y \cdot C_{3,n} - X_n - Y \cdot C'_{5,n} + Z'_n) = \Delta$$

$$(d + L) \sum_{n\in N} (\chi \cdot \varrho \cdot \mathcal{H}_1(att_{snd,n}) + \chi \cdot \varrho \cdot ek_{1,n} + \chi \cdot \varrho \cdot ek_2$$

$$-(\chi \cdot \varrho) \cdot \mathcal{H}_1(\Lambda(n)) - \chi \cdot \varrho \cdot ek_{1,n} - \chi \cdot \varrho \cdot ek_2 + t_n \lambda_n \cdot g)$$

$$= (d + L) \cdot g \sum_{n\in N} (\lambda_n \cdot t_n)$$

$$= (d + L) \cdot g = \Delta$$

If $\mathcal{R} \models \mathbb{R}$ holds, then according to the decryption algorithm, we can recover the message $\mathcal{M}$ as follows:

$$C_1 - \sum_{h\in H}(dk_{1,h} + C_{2,h} - dk_2 - C_{4,h}) = \mathcal{M}$$

$$\mathcal{M}+k\cdot g-\sum_{h\in H}(\mathcal{H}_2(att_{rcv,h})+a\gamma\cdot g+\epsilon_h\delta_h\cdot g+k\cdot\eta+ek_2-a\gamma\cdot g$$
$$-a\cdot g - k\cdot\eta - \mathcal{H}_2(\Omega(h)))$$

$$= \mathcal{M} + k\cdot g - \sum_{h\in H}(\delta_h\cdot\epsilon_h)\cdot g$$
$$= \mathcal{M} + k\cdot g - k\cdot g = \mathcal{M}$$

If a user is removed during the Rev phase from the system, the cloud server deletes the user's registration key, which makes it impossible to recover C_5' and Z', stopping the Match process. If a user needs a new registration key, the cloud server generates and sends it to them. For attribute updates, the KGC and the cloud server utilize uk_1 and uk_2 to perform the computation.

$$Rk_{u,m} = Rk_{u,n}+\Theta_u\cdot uk_{1,u} = \Theta_u\cdot\mathcal{H}(att_{u,m})+\Theta_u\cdot g = Rk_{u,m}$$
$$ek_{1,m} = ek_{1,n} + uk_2 = ek_{1,n} + ek_{1,m} - ek_{1,n} = ek_{1,m}$$
$$dk_{1,m} = dk_{1,n} + uk_{1,rcv} = \mathcal{H}_2(att_{rcv,m}) + a\gamma\cdot g = dk_{1,m}$$
$$C_{3,m} = C_{3,n} + uk_{1,snd} + uk_{2,snd}$$
$$= \mathcal{H}_1(att_{snd,m}) + ek_{1,m} + ek_2 = C_{3,m}$$
$$C_{5,m} = C_{5,n} + uk_{2,snd} + Rk_{m,snd} - Rk_{n,snd}$$
$$= ek_{1,m} + ek_2 + Rk_{m,snd} = C_{5,m}$$

For the equality test, it checks:

$$Q_n = C_{7,n} - C_{6,n}$$
$$= r_n\cdot m_n + s_n\cdot\vartheta_n - s_n\cdot\vartheta_n$$
$$= r_n\cdot m_n$$
$$Q_A\cdot TD_B' = (r_A\cdot r_B\cdot m_A\cdot g)$$
$$Q_B\cdot TD_A' = (r_A\cdot r_B\cdot m_B\cdot g)$$

Equality holds if $m_A = m_B$.

In short, the algorithm efficiently updates ciphertext parts and the keys using the original values. Users do not have to participate in the revocation process, which makes the system more practical when there are numerous revocation transactions.

E. Security Proofs

The following theorems show that we can achieve both IND-CCA2 and OW-CCA2 security through the random oracle model. In this section, our security analysis demonstrates that the RBAC-ET scheme is secure under the Elliptic Curve Decisional Diffie-Hellman (ECDDH) Problem and Elliptic Curve Discrete Logarithm problem (ECDLP) assumptions in the random oracle model.

Theorem 1: Assume $\mathcal{A}_1$ adversary can win the IND-CCA2 game in polynomial time with non-negligible advantage ϵ, then in polynomial time the challenger $\mathcal{C}$ can solve the Elliptic Curve Decisional Diffie-Hellman (ECDDH) Problem with non-negligible advantage ϵ'.

Proof: Suppose that $\mathcal{A}_1$ is a Type-I adversary against the IND-CCA2 security that can break our proposed scheme with a non-negligible advantage ϵ. Then, we can build a simulator $\mathcal{C}$ to break the ECDDH problem. Given an instance $(A = \alpha g, B = \beta g, Z)$, the goal is to distinguish whether $Z = \alpha\beta g$, or a random point in $\mathbb{G}$. The security game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_1$ is as follows:

Init: $\mathcal{C}$ receives the challenging sender's attribute set $\mathcal{S}^*$ and receiver's attribute set $\mathcal{R}^*$ from $\mathcal{A}_1$.

Setup: $\mathcal{C}$ sets up the public parameters for the adversary. The crucial step is to embed part of the ECDDH challenge into these parameters. $\mathcal{C}$ sets the master public key $\eta = A = \alpha g$. Note that $\mathcal{C}$ does not know the corresponding master secret key $a = \alpha$. The public parameters are announced as $pp = \{EC, \mathbb{G}, \mathbb{Z}_q^*, \eta, g, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$.

Hash Oracles: $\mathcal{C}$ will simulate the three hash functions $(\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3)$ as random oracles, maintaining lists $(\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3)$ to ensure that identical hash queries receive consistent responses.

Phase 1: $\mathcal{A}_1$ selects the challenging sender's policy $\mathbb{S}^*$ and receiver's policy $\mathbb{R}^*$. $\mathcal{C}$ constructs two disjoint sets: challenge attributes set Ψ and non-challenge attributes set Υ. $\mathcal{A}_1$ adaptively queries oracles, with $\mathcal{C}$ ensuring queried attributes do not trivially decrypt the challenge ciphertext.

Now, the challenger responds the adversary $\mathcal{A}_1$ on various hash queries as follows:

- $H_1 - Query$: When $\mathcal{A}_1$ queries for the sender attribute $att_{snd,n}$. Given an input tuple $(att_{snd,n})$, the challenger $\mathcal{C}$ first checks whether a matching entry $(att_{snd,n}, U_n)$ already exists in the list $\mathcal{L}_1$. If such an entry is found, $\mathcal{C}$ returns the corresponding value U_n. Otherwise, $\mathcal{C}$ randomly selects $U_n \in \mathbb{G}$, stores the tuple $(att_{snd,n}, U_n)$ in $\mathcal{L}_1$, and then returns U_n.
- $H_2 - Query$: When $\mathcal{A}_1$ queries for the receiver attribute $att_{rcv,h}$. Given an input tuple $(att_{rcv,h})$, the challenger $\mathcal{C}$ first checks whether a matching entry $(att_{rcv,h}, V_n)$ already exists in the list $\mathcal{L}_2$. If such an entry is found, $\mathcal{C}$ returns the corresponding value V_n. Otherwise, $\mathcal{C}$ randomly selects $V_n \in \mathbb{G}$, stores the tuple $(att_{rcv,h}, V_n)$ in $\mathcal{L}_2$, and then returns V_n.
- $H_3 - Query$: Given an input tuple $(\mathcal{M})$, the challenger $\mathcal{C}$ first checks whether a matching entry $(\mathcal{M}, y)$ already exists in the list $\mathcal{L}_3$. If such an entry is found, $\mathcal{C}$ returns the corresponding value y. Otherwise, $\mathcal{C}$ randomly selects $y \in \mathbb{Z}_q^*$, stores the tuple $(\mathcal{M}, y)$ in $\mathcal{L}_3$, and then returns y.

Phase 2: $\mathcal{A}_1$ adaptively initiate some queries to $\mathcal{C}$, and $\mathcal{C}$ responds to these queries as follows.

- $\mathcal{O}_{\text{Register}}(.)$: When $\mathcal{A}_1$ requests a registration key, the simulator $\mathcal{C}$ can compute it directly. $\mathcal{C}$ generates a unique user identifier $\Theta_u \in \mathbb{Z}_q^*$ and computes the registration keys as:

$$Rk_{snd,n} = \Theta_u\cdot\mathcal{H}_1(att_{snd,n}) + \Theta_u\cdot g$$

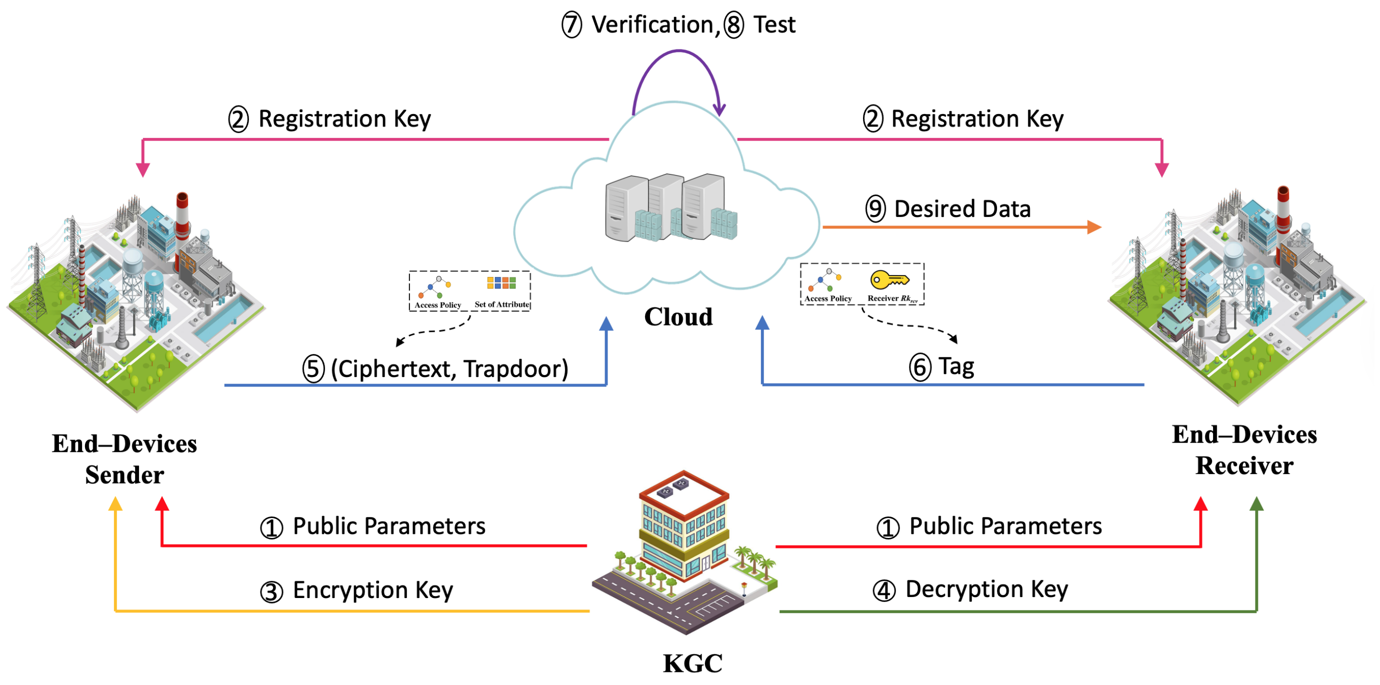


Fig. 4. System model of RBAC-ET

$$Rk_{rcv,h} = \Theta_u \cdot \mathcal{H}_2(att_{rcv,h}) + \Theta_u \cdot g$$

and return $Rk_{snd,n}$, $Rk_{rcv,h}$ to $\mathcal{A}_1$.

- $\mathcal{O}_{EKGen}(.)$: When the adversary $\mathcal{A}_1$ requests an encryption key ek for a sender attribute $att_{snd,n}$, the simulator's response depends on whether the attribute belongs to the challenge set Ψ or the non-challenge set Υ.
  a) if $att_{snd,n} \in \Psi$, $\mathcal{C}$ must embed the challenge element $A = \alpha \cdot g$ into encryption key, it randomly picked $y \in \mathbb{Z}_q^*$, and compute ek as:
  $$ek_{1,n} = \mathcal{H}_1(att_{snd,n}) + y \cdot A, \qquad ek_2 = A$$
  $\mathcal{C}$ returns the encryption key ek^* to $\mathcal{A}_1$.
  b) if $att_{snd,n} \in \Upsilon$, $\mathcal{C}$ picks two random numbers $x, y \in \mathbb{Z}_q^*$, where x is a known secret, and compute:
  $$ek_{1,n} = \mathcal{H}_1(att_{snd,n}) + xy \cdot g, \qquad ek_2 = x \cdot g$$
  and returns ek to $\mathcal{A}_1$.
- $\mathcal{O}_{DKGen}(.)$: When the adversary $\mathcal{A}_1$ requests a decryption key dk for a receiver attribute $att_{rcv,h}$, the simulator's response depends on whether the attribute belongs to the challenge set Ψ or the non-challenge set Υ.
  a) if $att_{rcv,h} \in \Psi$, The simulator must embed the challenge element $A = \alpha \cdot g$ into decryption key, it randomly picked $c \in \mathbb{Z}_q^*$, and compute dk as:
  $$dk_{1,h} = \mathcal{H}_2(att_{rcv,h}) + c \cdot A, \qquad dk_2 = c \cdot A + A$$
  $\mathcal{C}$ returns the decryption key dk^* to $\mathcal{A}_1$.
  b) if $att_{rcv,h} \in \Upsilon, \mathcal{C}$ picks two random numbers $c, d \in \mathbb{Z}_q^*$, where d is a known secret, and compute:
  $$dk_{1,h} = \mathcal{H}_2(att_{rcv,h}) + cd \cdot g, \qquad dk_2 = cd \cdot g + d \cdot g$$
  and returns dk to $\mathcal{A}_1$.
- $\mathcal{O}_{Enc}(., ., ., ., ., .)$: $\mathcal{A}_1$ queries the encryption oracle. The oracle randomly picks a number $k \in \mathbb{Z}_q^*$, and parses receiver access control structure $\mathbb{R} = (\mathbb{W}, \Omega)$, and randomly picks $\vec{x} = (k, x_2, \ldots, x_{n_\mathbb{W}})^\perp \in \mathbb{Z}_q^{n_\mathbb{W} \times 1}$ and computes $\vec{\delta} = (\delta_1, \delta_2, \ldots, \delta_{l_\mathbb{W}}) = \mathbb{W}\vec{x}$. Then, it picks some random numbers such as $d, s, r, L, \vartheta \in \mathbb{Z}_q^*$, and calculate $\Delta = (d + L) \cdot g$, and $m = \mathcal{H}_3(\mathcal{M})$.
  a) For the challenge sender attribute set $\mathcal{S}^*$ and receiver policy $\mathbb{R}^*$, the simulator is given the ECDDH challenge $(A = \alpha g, B = \beta g, Z)$. It sets $\eta = A$ and implicitly sets the message-blinding random value $k = \beta$ by using B in place of $k \cdot g$. The challenge ciphertext C^* is constructed as follows:
  $$C_0^* = d^* + L^*, \quad C_1^* = \mathcal{M}^* + B,$$
  $$C_{2,h}^* = \epsilon_h^* \delta_h^* \cdot g + Z + A,$$
  $$C_{3,n}^* = \mathcal{H}_1(att_{snd,n}) + ek_{1,n}^* + ek_2^*$$
  $$C_{4,h}^* = Z + \mathcal{H}_2(\Omega(h))$$
  $$C_{5,n}^* = ek_{1,n}^* + ek_2^* + Rk_{snd,n}^*, \quad C_6^* = s^* \cdot \vartheta^*$$
  $$C_7^* = r^* \cdot m^* + s^* \cdot \vartheta^*$$
  and return the challenge ciphertext C^* to $\mathcal{A}_1$
  b) Else, the simulator $\mathcal{C}$ performs the encryption by faithfully running the standard `Enc` algorithm, and return the ciphertext C to $\mathcal{A}_1$
- $\mathcal{O}_{Dec}(.)$: C responds to decryption queries for any ciphertext $C \neq C^*$ by using its ability to generate the necessary decryption keys through the $\mathcal{O}_{\text{DKGen}}$ oracle, and

compute:

$$C_1 - \sum_{h \in H} (dk_{1,h} + C_{2,h} - dk_2 - C_{4,h}) = \mathcal{M}$$

and return $\mathcal{M}$ to $\mathcal{A}_1$.

Challenge: $\mathcal{M}_1$ and $\mathcal{M}_2$ are two equal-length but dissimilar messages chosen by $\mathcal{A}_1$. Upon receiving the messages $\mathcal{M}_1$ and $\mathcal{M}_2$, $\mathcal{C}$ flips a random coin $f \in \{0,1\}$ and encrypts $\mathcal{M}_f$ using the $\mathcal{O}_{\text{Enc}}$ oracle under $(ek^*, \mathcal{S}^*, \mathbb{R}^*)$, as described above to compute the challenge ciphertext and returns the ciphertext C^* to $\mathcal{A}_1$.

Phase 3: $\mathcal{A}_1$ issues additional queries (excluding trivial decryption of the challenged ciphertext C^*). $\mathcal{C}$ responds to all the queries with the same constraint as in Phase 1.

Guess: The adversary $\mathcal{A}$ outputs a bit guess $f' \in \{0,1\}$. The simulator $\mathcal{C}$ outputs "ECDDH tuple" iff $f' = f$; otherwise, it outputs "random tuple".

Analysis of Adversary's Success Probability:

Case 1: $Z = \alpha\beta g$. In this case, the challenge ciphertext C^* is valid and correctly constructed. The simulated components $C_2, h = \epsilon_h \delta_h \cdot g + \alpha\beta g + ek_2$ and $C_4, h = \alpha\beta g + H_2(\Omega(h))$ are consistent with a real encryption where $k = \beta$ and $\eta = A$. By the premise of the theorem, $\mathcal{A}_1$ has a non-negligible advantage ϵ in this game. Thus, its probability of guessing correctly is:

$$\Pr[f' = f \mid Z = \alpha\beta g] = \frac{1}{2} + \epsilon$$

Case 2: $Z = zg$ is a random group element. When Z is a random element from $\mathbb{G}$, the term $C_1 = \mathcal{M}_f + B$ perfectly hides the message $\mathcal{M}_f$ due to the randomness of $B = \beta g$. The components $C_{2,h}$ and $C_{4,h}$ are randomized using Z, which provides no information about the relationship between A and B.

The adversary's view is computationally independent of the challenge bit f. Therefore, adversary A_1 cannot distinguish which message was encrypted and can do no better than random guessing. Its probability of success in this case is:

$$\Pr[f' = f \mid Z = zg] = \frac{1}{2}$$

The advantage of $\mathcal{C}$, denoted ϵ', is given by:

$$\begin{aligned}\epsilon' &= |\Pr[\mathcal{C} \to 1 | Z = \alpha\beta g] - \Pr[\mathcal{C} \to 1 | Z = zg]| \\ &= \left| \left(\frac{1}{2} + \epsilon\right) - \frac{1}{2} \right| = \epsilon.\end{aligned}$$

Since ϵ is non-negligible, the advantage ϵ' is also non-negligible.

Analysis of Computational Effort: The time complexity of the algorithm is bound by

$$t' = \mathcal{O}\Big(t + \sum_{i=1}^{3} q_{H_i} \cdot T_h + (q_{\text{Reg}} + q_{\text{EKGen}} + q_{\text{DKGen}} + q_{\text{Enc}} + q_{\text{Dec}}) \cdot N_{\text{att}} \cdot (T_{\text{sm}} + T_{\text{pa}})\Big)$$

where

- t: Time taken by the adversary A_1 (external to the simulation).
- q_{H_i}: Number of queries to the hash function H_i, where $i = 1, 2, 3$.
- T_h: Time cost of a single hash function evaluation.
- q_{Reg}, q_{EKGen}, q_{DKGen}, q_{Enc}, q_{Dec}: Number of registration, encryption key generation, decryption key generation, encryption, and decryption oracle queries.
- N_{att}: Number of attributes processed per query.
- T_{sm}: Time for one scalar multiplication on an elliptic curve.
- T_{pa}: Time for one point addition on an elliptic curve.

Theorem 2: The proposed RBAC-ET scheme is secure against a Type-II adversary $\mathcal{A}_2$ (either a curious cloud server or a curious (KGC) under the ECDDH assumption in the random oracle model.

Proof: For a provided ECDDH problem instance $(g, A = \alpha g, B = \beta g, Z)$, the engagement between $\mathcal{C}$ and $\mathcal{A}_2$ mirrors that described in Theorem 1. However, the subsequent stages of their interaction introduce distinct modifications. The proof handles two cases.

Case 1: $\mathcal{A}_2$ is a curious key generation center

In this scenario, $\mathcal{A}_2$ is a curious KGC with access to the master secret key $msk = a = \alpha$. However, it is restricted from invoking the $\mathcal{O}_{\text{Register}}$ oracle when either $(\mathbb{S}^* \cap \mathcal{S}^*) \nsubseteq (\mathbb{S} \cap \mathcal{S})$ or $(\mathbb{R}^* \cap \mathcal{R}^*) \nsubseteq (\mathbb{R} \cap \mathcal{R})$, holds. Notably, $\mathcal{A}_2$ does not possess the valid registration key Rk_u, which makes it impossible to recover C_5' and Z', stopping the match process. Interaction between simulator $\mathcal{C}$ and $\mathcal{A}_2$ is given as:

Setup: $\mathcal{C}$ initializes the system by generating public parameters $pp = \{EC, \mathbb{G}, \mathbb{Z}_q^*, \eta, g, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$ and provides them to $\mathcal{A}_2$.

Queries: $\mathcal{A}_2$ adaptively queries oracles such as $\mathcal{O}_{\mathcal{H}_i}$, $\mathcal{O}_{\text{EKGen}}$, $\mathcal{O}_{\text{DKGen}}$, $\mathcal{O}_{\text{Encrypt}}$, and $\mathcal{O}_{\text{Decrypt}}$, subject to the restriction on $\mathcal{O}_{\text{Register}}$.

Challenge: In the challenge phase, without the valid registration key Rk_u, $\mathcal{C}$ cannot compute the ciphertext component $C'_{5,n}$ and Z'_n, stopping the match process. As a result, it cannot decrypt C^* to recover $\mathcal{M}^*$. The advantage of $\mathcal{A}_2$ in distinguishing $\mathcal{M}_f$ is negligible due to its inability to compute C^*.

Case 2: $\mathcal{A}_2$ is a curious cloud server

In this scenario, $\mathcal{A}_2$ is a curious cloud server capable of forging arbitrary registration keys Rk_u. However, it cannot forge the encryption keys ek or decryption keys dk. Interaction between $\mathcal{C}$ and $\mathcal{A}_2$ is given as:

Setup: $\mathcal{C}$ provides the public parameters pp to $\mathcal{A}_2$, as in the previous case.

Queries: $\mathcal{A}_2$ queries oracles similarly to a Type-I adversary $\mathcal{A}_1$, subject to the restriction on $\mathcal{O}_{\text{EKGen}}$ and $\mathcal{O}_{\text{DKGen}}$.

Challenge: $\mathcal{M}_1$ and $\mathcal{M}_2$ are chosen by $\mathcal{A}_1$, and sent to $\mathcal{C}$. $\mathcal{C}$ flips a coin $f \in \{0,1\}$, $\mathcal{M}^* \in \{0,1\}$, and encrypts M_f by calling $\mathcal{O}_{\mathsf{Enc}}(\mathcal{S}^*, \mathbb{R}^*)$ and returns the ciphertext C^* to $\mathcal{A}_1$.

Guess: Despite its ability to forge Rk_u, $\mathcal{C}$ cannot access dk. This limitation prevents it from decrypting the target C^*. In the guess phase, mirroring the challenges faced by $\mathcal{A}_1$. The advantage of $\mathcal{A}_2$ in distinguishing $\mathcal{M}_f$ is negligible due to its inability to compute C^*.

In both cases, the advantage of $\mathcal{A}_2$ in the security game is equivalent to the advantage of the simulator $\mathcal{C}$, which is

ϵ, in solving the ECDDH problem. This establishes that the proposed scheme is secure against Type-II adversaries under the hardness of the ECDDH problem.

Theorem 3: Assume the ECDDH and ECDLP assumptions hold in the group $\mathbb{G}$. Then, the RBAC-ET can achieve trapdoor indistinguishability and is OW-CCA2 secure in the random oracle model.

Proof: In comparison to Theorem 1, the adversary $\mathcal{A}$ in this game also has access to the trapdoor information associated with the ciphertext. We must therefore demonstrate that this trapdoor information does not help the adversary in guessing the plaintext, meaning the trapdoor does not disclose any information about the message. We build a simulator $\mathcal{C}$ that can solve the ECDDH problem by interacting with an adversary $\mathcal{A}$ that has a non-negligible advantage ϵ in the OW-CCA2 game.

Setup: Given an ECDDH challenge instance $(g, A = \alpha g, B = \beta g, Z)$, where the goal is to distinguish whether $Z = \alpha\beta g$ or a random point in $\mathbb{G}$, $\mathcal{C}$ will interact with $\mathcal{A}$ as follows, $\mathcal{C}$ sets master public key $\eta = A = \alpha g$ (without knowing the master secret key $a = \alpha$). Initializes empty hash lists $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ for $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$. Publishes public parameters $pp = \{EC, \mathbb{G}, \mathbb{Z}_q^*, \eta, g, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$. $\mathcal{C}$ selects two disjoint sets: challenge attributes set Ψ and non-challenge attributes set Υ.

Phase 1: The adversary $\mathcal{A}_1$ adaptively queries the following oracles.

- $\mathcal{H}_1 - Query$: When $\mathcal{A}$ queries for sender attribute $att_{snd,n}$. If $(att_{snd,n}, U_n)$ exists in $\mathcal{L}_1$, return U_n. Otherwise, select $U_n \in \mathbb{G}$ randomly, store $(att_{snd,n}, U_n)$ in $\mathcal{L}_1$, and return U_n.
- $\mathcal{H}_2 - Query$: When $\mathcal{A}$ queries for receiver attribute $attr_{cv,h}$. If $(attr_{cv,h}, V_n)$ exists in $\mathcal{L}_2$, return V_n Otherwise, select $V_n \in \mathbb{G}$ randomly, store $(attr_{cv,h}, V_n)$ in $\mathcal{L}_2$, and return V_n
- $\mathcal{H}_3 - Query$: Given input $\mathcal{M}$. If $(\mathcal{M}, y)$ exists in $\mathcal{L}_3$, return y. Otherwise, select $y \in \mathbb{Z}_q^*$ randomly, store $(\mathcal{M}, y)$ in $\mathcal{L}_3$, and return y.

Phase 2: $\mathcal{A}_1$ adaptively initiate some queries to $\mathcal{C}$, and $\mathcal{C}$ responds to these queries as follows.

- $\mathcal{O}_{\mathsf{Register}}(.)$: Generates a unique user identifier $\Theta_u \in \mathbb{Z}_q^*$. Computes registration keys:

$$Rk_{snd,n} = \Theta_u \cdot \mathcal{H}_1(att_{snd,n}) + \Theta_u \cdot g$$
$$Rk_{rcv,h} = \Theta_u \cdot \mathcal{H}_2(attr_{rcv,h}) + \Theta_u \cdot g$$

  Returns $Rk_{snd,n}$, $Rk_{rcv,h}$ to $\mathcal{A}$

- $\mathcal{O}_{\mathsf{EKGen}}(\cdot)$: When the adversary $\mathcal{A}$ requests an encryption key ek for a sender attribute $att_{snd,n}$, the simulator generates it as.
  a) If $att_{snd,n} \in \Psi$ (challenge attribute):

$$ek_{1,n} = \mathcal{H}_1(att_{snd,n}) + y \cdot A, \qquad ek_2 = A$$

  $\mathcal{C}$ returns the encryption key ek^* to $\mathcal{A}$.
  b) If $att_{snd,n} \in \Upsilon$ (non-challenge attribute):

$$ek_{1,n} = \mathcal{H}_1(att_{snd,n}) + xy \cdot g, \qquad ek_2 = x \cdot g$$

  $\mathcal{C}$ returns the encryption key ek to $\mathcal{A}$.

- $\mathcal{O}_{\mathsf{DKGen}}(\cdot)$: When the adversary $\mathcal{A}$ requests a decryption key dk for a receiver attribute $att_{rcv,h}$, the simulator generates it as.
  a) If $att_{rcv,h} \in \Psi$ (challenge attribute):

$$dk_{1,h} = \mathcal{H}_2(att_{rcv,h}) + c \cdot A, \qquad dk_2 = c \cdot A + A$$

  $\mathcal{C}$ returns the decryption key dk^* to $\mathcal{A}$.
  b) If $att_{rcv,h} \in \Upsilon$ (non-challenge attribute):

$$dk_{1,h} = \mathcal{H}_2(att_{rcv,h}) + cd \cdot g, \qquad dk_2 = cd \cdot g + d \cdot g$$

  $\mathcal{C}$ returns the decryption key dk to $\mathcal{A}$.

- $\mathcal{O}_{\mathsf{Enc}}(.,.,.,)$: When $\mathcal{A}$ requests an encryption of a message $\mathcal{M}_n$, for non-challenge attributes $(att_{snd,n} \in \Upsilon)$, $\mathcal{C}$ encrypts $\mathcal{M}_n$ by faithfully running the standard `Enc` algorithm.
- $\mathcal{O}_{\mathsf{Trap}}(.)$: For ciphertext $C \neq C^*$ with the same value r stored during encryption. $\mathcal{C}$ rturns trapdoor $TD = r \cdot g + Rk$ to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{Dec}(.)}$: For any ciphertext $C \neq C^*$, $\mathcal{C}$ can correctly return the message $\mathcal{M}$ by running the full decryption algorithm, as it can generate all necessary keys and look up all hash values in its lists.

Challenge: $\mathcal{A}$ outputs challenge attribute and policy sets $(\mathcal{S}^*, \mathbb{R}^*)$ along with two different challenge messages $\mathcal{M}_0$ and $\mathcal{M}_1$ and sends them to the simulator. $\mathcal{C}$ picks a random bit $f \in \{0, 1\}$ and encrypts $\mathcal{M}_f$ using the ECDDH challenge: For the encryption, the simulator is given the ECDDH challenge $(A = \alpha g, B = \beta g, Z)$. It sets $\eta = A$ and implicitly sets the message-blinding random value $k = \beta$ by using B in place of $k \cdot g$. The challenge ciphertext is constructed as follows: $C_0^* = d^* + L^*$, $C_1^* = \mathcal{M}^* + B$, $C_{2,h}^* = \epsilon_h^* \delta_h^* \cdot g + Z + A$, $C_{3,n} = \mathcal{H}_1(att^*{}_{snd,n}) + ek_{1,n}^* + ek_2^*$, $C_{4,h}^* = Z + \mathcal{H}_2(\Omega(h))$, $C_{5,n}^* = ek_{1,n}^* + ek_2^* + Rk_{snd,n}^*$, $C_6^* = s^* \cdot \vartheta^*$, and $C_7^* = r^* \cdot m^* + s^* \cdot \vartheta^*$. The critical component for the equality test is $C_7^* = r^* \cdot m^* + s^* \cdot \vartheta^*$. $\mathcal{C}$ sends C^* to $\mathcal{A}$

Phase 2: $\mathcal{C}$ continues responding to $\mathcal{A}$'s queries with the restriction that it cannot provide decryption or trapdoors for C^*.

Guess: The adversary $\mathcal{A}$ outputs a bit guess $f' \in \{0, 1\}$. The simulator $\mathcal{C}$ outputs "ECDDH tuple" iff $f' = f$; otherwise, it outputs "random tuple".

Analysis of Adversary's Success Probability:

Case 1 ($Z = \alpha\beta g$): The challenge distribution is identical to a real execution of the scheme. By the premise, adversary $\mathcal{A}$ has a non-negligible advantage ϵ. Therefore, its probability of guessing the bit correctly is:

$$\Pr[f' = f \mid Z = \alpha\beta g] = \frac{1}{2} + \epsilon.$$

Case 2 ($Z = zg$): When Z is a random group element, the term $C_1^* = \mathcal{M}_f + B$ perfectly hides the message $\mathcal{M}_f$ due to the randomness of $B = \beta g$. The equality-test component $Q^* = r^* \cdot m^*$ reveals no information about m^* because recovering the random value r^* from its public value $r^* \cdot g$ requires solving the ECDLP. The adversary's view is computationally independent of the bit f, so it can do no better than random guessing. Its success probability is:

$$\Pr[f' = f \mid Z = zg] = \tfrac{1}{2}$$

Therefore, the ECDDH advantage is

$$\begin{aligned}\epsilon' &= |\Pr[\mathcal{C} \to 1 | Z = \alpha\beta g] - \Pr[\mathcal{C} \to 1 | Z = zg]| \\ &= \left| \left(\frac{1}{2} + \epsilon \right) - \frac{1}{2} \right| = \epsilon.\end{aligned}$$

Since ϵ is non-negligible, ϵ' is also non-negligible. This demonstrates that an adversary with a non-negligible advantage in the OW-CCA2 game can be used to solve the ECDDH problem. The proof also establishes trapdoor indistinguishability, as the hardness of the ECDLP assumption computationally hides the equality test values.

Analysis of Computational Effort: According to the above simulation, we obtain that the time complexity of the algorithm is bound by

$$\begin{aligned} t' = \; \mathcal{O}\Big(t + \sum_{i=1}^{3} q_{H_i} \cdot T_h + (q_{\text{Reg}} + q_{\text{EKGen}} + q_{\text{DKGen}} \\ + \; q_{\text{Enc}} + q_{\text{Trap}} + q_{\text{Dec}}) \cdot N_{\text{att}} \cdot (T_{\text{sm}} + T_{\text{pa}})\Big)\end{aligned}$$

where q_{Trap} is the number of trapdoor oracle queries.

VI. Performance Evaluation

In this section, we analyze RBAC-ET's performance from two perspectives: theoretical complexity and experimental evaluation. The theoretical complexity encompasses communication, space complexity, and computational cost, providing a foundational understanding of our scheme's efficiency and scalability. Experimentally, RBAC-ET is benchmarked against six existing schemes: PBAC-FG [29], FTA-BAC [32], CRO-ABME [36], FS-IBEET [14], PriBAC [35], and PVPKEET [17]. We focus on running times to highlight the relative strengths and potential limitations of our scheme.

A. Theoretical Complexity

Table II presents the comprehensive comparison of the computational costs of RBAC-ET against the six benchmark schemes PBAC-FG, FTA-BAC, CRO-ABME, FS-IBEET, PriBAC, and PVPKEET across nine fundamental cryptographic operations: `Setup`, `Register`, `EKGen`, `DKGen`, `Enc`, `Ver`, `Trapdoor`, `Test`, and `Dec`. RBAC-ET achieves the lowest computational cost during the `Setup` phase, requiring only a single group multiplication (M_G) operation. In contrast, PBAC-FG, FTA-BAC, CRO-ABME, FS-IBEET, PriBAC, and PVPKEET incur higher costs due to the utilization of bilinear pairing and multiple exponentiation operations. Notably, scheme PriBAC exhibits the highest setup cost because it exhibits a linear relationship with the total number of system attributes (n). During the `Register` phase, CRO-ABME grows linearly with the number of sender's attributes, and RBAC-ET shows very low computation cost compared to the CRO-ABME scheme. During the `EKGen` phase, FTA-BAC grows linearly with the number of sender's attributes and incurs the highest cost. Our scheme has a much lower computational cost than all the other schemes. The computational cost of PBAC-FG, CRO-ABME, and PriBAC increases relative to the attributes of the sender and the associated cryptographic operations. The computational complexity of the `DKGen` algorithm presented in our scheme scales linearly with respect to receiver attributes, whereas FTA-BAC requires additional computational resources depending on the sender and receiver access policy and the number of attributes. Schemes PBAC-FG, CRO-ABME, and PriBAC have relatively similar computational costs. In `Enc` phase, PVPKEET demonstrates the lowest computational cost because this scheme does not provide a bilateral access control mechanism. FTA-BAC incurs the highest encryption cost due to numerous and complex bilinear pairing and exponential operations, while PBAC-FG, CRO-ABME, FS-IBEET, and PriBAC exhibit intermediate costs. Our scheme demonstrates the lowest computational cost among all bilateral access control schemes, as it does not utilize any bilinear pairings or exponential operations. We also learn the computational cost of the `Ver` algorithm. The bilinear mapping grows proportionally according to the attributes in both the sender's and receiver's policies in PBAC-FG, FTA-BAC, CRO-ABME, and PriBAC. Our scheme is only linearly related to the number of sender attributes in this phase. Now, we will discuss the `Equality Test` algorithm computation cost. The FS-IBEET and PVPKEET schemes' costs grow linearly due to bilinear exponentiation operations. Our scheme shows the lowest computation cost for the equality text algorithm. It's important to highlight that the computational cost of the `Dec` algorithm in all existing schemes is higher than in ours. PBAC-FG and FTA-BAC schemes exhibit the highest decryption costs, with computational complexity scaling linearly with respect to the attributes in the receiver's policy. Lastly, we will discuss the computational cost of the `revocation` algorithm. The execution time grows proportionally with the number of attributes requiring updates for the CRO-ABME scheme. Our scheme exhibits the lowest computational cost because it does not utilize any bilinear pairings or exponential operations.

Table III illustrates the communication and space complexities, highlighting that the KGC space complexity is constant $\mathcal{O}(1)$ for PBAC-FG, FTA-BAC, and our scheme, because the authority keeps only master/public parameters; it becomes linear $\mathcal{O}(n)$ in the attribute universe for PriBAC and CRO-ABME, which maintain per-attribute or per-user state. Furthermore, our scheme's space complexity for the sender, cloud, and receiver is comparable to that of PBAC-FG while being demonstrably more efficient than FTA-BAC, PriBAC, and CRO-ABME. This efficiency is a direct result of our scheme's simpler and more optimized access policy structure, which avoids the complex constructions found in competing schemes. Regarding communication complexity, our scheme achieves linear complexity in KGC and user interactions, outperforming both FTA-BAC and PriBAC. This improvement is because our design ensures the privacy of policy content by not incorporating it into the encryption and decryption keys generated by the KGC. In sender and cloud communications, our scheme exhibits a complexity comparable to PBAC-FG, PriBAC, and CRO-ABME, and is superior

TABLE II
COMPARISON OF COMPUTATIONAL COST ACROSS RELATED SCHEMES.

Schemes	Computational Complexity									
	KGC			Cloudlet/Edge Server			Sender		Receiver	
PBAC-FG [29]	Setup	EKGen	DKGen	Verification			Enc		Dec	
	$e+2e_0$ $+2e_1$	$(\lvert\mathcal{S}\rvert+1)e_0$ $+\lvert\mathcal{S}\rvert h+M_G$	$3\lvert\mathbb{R}\rvert e_0$ $+\lvert\mathbb{R}\rvert h+M_G$	$(2\lvert\mathbb{S}\rvert+1)e+\lvert\mathbb{S}\rvert e_1+(\lvert\mathbb{S}\rvert+1)h+M_G$			$(\lvert\mathcal{R}\rvert+\lvert\mathcal{S}\rvert+4)e_0+e_1+$ $(\lvert\mathcal{R}\rvert+\lvert\mathcal{S}\rvert+1)h+4M_G$		$2\lvert\mathbb{R}\rvert e+\lvert\mathbb{R}\rvert e_1+M_G$	
FTA-BAC [32]	Setup	EKGen	DKGen	Verification			Enc		Dec	
	$2e_1$	$(\lvert\mathcal{S}\rvert+2)e_0$ $+\lvert\mathcal{S}\rvert h+M_G$	$(4\lvert\mathcal{R}\rvert+2)e_0+(\lvert\mathbb{R}\rvert$ $+\lvert\mathbb{S}\rvert)h+2M_G$	$(2\lvert\mathbb{S}\rvert+2\lvert\mathbb{R}\rvert)e+(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert)h$			$(2\lvert\mathbb{R}\rvert+\lvert\mathbb{S}\rvert+4)e_0+(2\lvert\mathbb{R}\rvert$ $+\lvert\mathbb{S}\rvert+2)h+2M_G$		$2\lvert\mathbb{R}\rvert e+\lvert\mathbb{R}\rvert e_1+M_G$	
CRO-ABME [36]	Setup	EKGen	DKGen	Register	Verification		Enc		PkGen	Dec
	$(n+2)e_0$	$\lvert\mathbb{S}\rvert e_0$	$\lvert\mathbb{R}\rvert e_0$	$(\lvert\mathcal{S}\rvert+\lvert\mathcal{R}\rvert)e_0$	$(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert+1)e+(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert)e_0$		$4e+(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+6)e_0+2h$		$(\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert+1)e_0$	$2e_1+2M_G+2h$
FS-IBEET [14]	Setup	KeyGen		Equality Test			Enc	Trapdoor	Dec	
	$e+2e_0$	$2e_0+h$		$4e+2e_0+2h$			$(7+T)e_0+(5+T)h+2M_G$	$(2+T)e+Te_0+Th$	$2e+2e_0+M_G$	
PriBAC [35]	Setup	EKGen	DKGen	Verification			Enc		TrGen	Dec
	$(2n+2)e_0$	$\lvert\mathcal{S}\rvert e_0$	$\lvert\mathcal{R}\rvert e_0$	$(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert+1)e+(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert)e_0$			$4e+(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+6)e_0+2h$		$(\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert+1)e_0$	$(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert)e+2h$
PVPKEET [17]	Setup	UserKeyGen	CloudKeyGen	Equality Test			Enc		Dec	
	e_0	e_0	e_0	$2e+2e_0+2M_G$			$4e_0+2M_G+h$		e_0+M_G	
Our	Setup	EKGen	DKGen	Register	Verification	Equality Test	Enc	Trapdoor	Tag	Dec
	M_G	$\lvert\mathcal{S}\rvert h+3M_G$	$\lvert\mathcal{R}\rvert h+5M_G$	$(\lvert\mathcal{S}\rvert+\lvert\mathcal{R}\rvert)h+$ $(\lvert\mathcal{S}\rvert+\lvert\mathcal{R}\rvert+1)M_G$	$2M_G$	$2M_G$	$(\lvert\mathcal{S}\rvert+\lvert\mathbb{R}\rvert+1)h$ $+(\lvert\mathbb{R}\rvert+8)M_G$	M_G	$\lvert\mathbb{S}\rvert h+(2\lvert\mathbb{S}\rvert$ $+4)M_G$	M_G

[1] n: The maximum number of user attributes in the system; T: The number of 0 in the time slot; $|\mathcal{S}|/|\mathcal{R}|$: The number of a sender's/receiver's attributes; $|\mathbb{S}|/|\mathbb{R}|$: The number of attributes in a sender's/receiver's policy.

[2] $\mathbb{G}/\mathbb{G}_T$: additive group $\mathbb{G}$ of prime order p; e: Bilinear maps; e_o: Exponentiations in $\mathbb{G}$; e_1: Exponentiations in $\mathbb{G}_T$; h: Hashes to $\mathbb{G}$; M_G: A scalar multiplication operation in $\mathbb{G}$.

TABLE III
SPACE AND COMMUNICATION COMPLEXITY ACROSS DIFFERENT ACCESS CONTROL SCHEMES.

Schemes	Space Complexity				Communication Complexity		
	KGC	Sender	Cloud	Receiver	KGC and User	Cloud and Sender	Cloud and Receiver
PBAC-FG [29]	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert)$
FTA-BAC [32]	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)$	$\mathcal{O}(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathbb{S}\rvert+\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathbb{S}\rvert+\lvert\mathbb{R}\rvert)$
PriBAC [35]	$\mathcal{O}(n)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{R}\rvert)+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert)$
CRO-ABME [36]	$\mathcal{O}(n)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{R}\rvert)+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)+\lvert\mathbb{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{R}\rvert)+\lvert\mathbb{R}\rvert)$
Ours	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathcal{S}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathcal{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{R}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert+\lvert\mathbb{S}\rvert+\lvert\mathcal{R}\rvert+\lvert\mathbb{R}\rvert)$

n: The maximum number of user attributes in the system; $|\mathcal{S}|/|\mathcal{R}|$: The number of a sender's/receiver's attributes; $|\mathbb{S}|/|\mathbb{R}|$: The number of attributes in a sender's/receiver's policy.

to FTA-BAC. Notably, during receiver-cloud interactions for data retrieval, our scheme's communication complexity is comparable to that of PBAC-FG and PriBAC, reflecting a robust protocol that ensures secure and policy-compliant data access. These results collectively underscore the effectiveness of our scheme, offering a highly scalable and efficient solution for resource-constrained industrial Internet of Things devices.

B. Experimental Evaluation

To verify the practicality of RBAC-ET, we compared its efficiency with that of state-of-the-art works, including PBAC-FG, FTA-BAC, CRO-ABME, FS-IBEET, PriBAC, and PVPKEET, which are closely related to outsourced ME-based schemes. We conducted our experimental simulation on a PC running the x86_64 macOS Big Sur 11.4 operating system, which has an i7 4770HQ 2.20 GHz CPU and 16 GB of memory. We perform elliptic curve point multiplication operations using the standard MIRACL cryptography library. For bilinear pairing operations, we use Java OpenJDK 17.0.2 and the JPBC library. We carried out a series of experiments covering the following: `Setup`, `Register`, `EKGen`, `DKGen`, `Enc`, `Ver`, `Trapdoor`, `Test`, and `Dec`.

Fig. 5 presents a detailed comparison of the schemes, highlighting those with the maximum and minimum computational running times.

Fig. 5(a) shows that PriBAC and CRO-ABME take more running time for system setup than the remaining schemes and grow linearly with the size of the attribute universe (n). PBAC-FG, FTA-BAC, FS-IBEET, and PVPKEET exhibit relatively stable and constant time regardless of the number of attributes. In contrast, our proposed scheme shows the minimum constant running time for system setup.

Fig. 5(b) presents the time for the user registration algorithm. CRO-ABME grows linearly with the number of sender attributes, and RBAC-ET shows very low computation cost compared to the CRO-ABME scheme.

Fig. 5(c) presents the time for encryption key generation. Our scheme has the lowest runtime for the encryption key generation due to its optimized use of cryptographic operations. PBAC-FG and FTA-BAC take significantly more time because they require additional computations involving a large number of sender attributes $|\mathcal{S}|$. CRO-ABME and PriBAC exhibit comparatively lower running times than PBAC-FG and FTA-BAC, but still require more time than our scheme.

Fig. 5(d) depicts that RBAC-ET has the minimum running time for the decryption key generation among all existing schemes. FTA-BAC and PBAC-FG exhibit the highest running time due to their reliance on a higher number of expensive bilinear pairing, group multiplication, and exponentiation operations. CRO-ABME and PriBAC have intermediate running times.

Fig. 5(e) illustrates that FTA-BAC takes more running time for encryption than the remaining schemes because it is associated with a large number of attributes in the sender's and receiver's access policies $|\mathbb{S}|$ and $|\mathbb{R}|$. PVPKEET shows the lowest running time because it does not provide a bilateral access control mechanism. Our scheme exhibits the minimum running time among all bilateral access control schemes, as it utilizes no bilinear pairing and exponentiation operations during message encryption. PBAC-FG, CRO-ABME, and PriBAC have similar running times.

Fig. 5(f) depicts the time required for access policy matching verification in our system. PBAC-FG, CRO-ABME, and PriBAC take the intermediate time and increase proportionally regarding the attributes in the sender's $|\mathbb{S}|$ and receiver's $|\mathbb{R}|$ access policies. RBAC-ET has better performance than others. This efficiency is due to avoiding extensive bilinear pairing and exponentiation operations. FTA-BAC has the highest running time in this phase.

Fig. 5(g) presents the equality test running time vs the number of times executed. RBAC-ET has a constant running time for the equality test algorithm compared to the FS-IBEET and PVPKEET schemes. The running time of the FS-IBEET and PVPKEET schemes grows linearly due to the bilinear exponentiation operations.

Fig. 5(h) presents the decryption running time vs the number of attributes and policies. RBAC-ET achieves a constant running time for decryption, compared to all existing schemes. The running time also remains constant in FS-IBEET, CRO-ABME, and PVPKEET schemes. The FTA-BAC and PBAC-FG schemes take the highest running time, with computational complexity scaling linearly with the attributes in the receiver's access policy $|\mathbb{R}|$.

Fig. 5(i) presents the revocation algorithm's running time vs the number of users to update. RBAC-ET achieves a constant running time for user updates compared to the CRO-ABME scheme. The CRO-ABME scheme takes the highest running time, with computational complexity scaling linearly with the number of attributes requiring updates.

Overall, RBAC-ET outperforms all the existing schemes: PBAC-FG, FTA-BAC, CRO-ABME, FS-IBEET, PriBAC, and PVPKEET. Our scheme achieves superior computational efficiency, minimizes communication and storage overheads across all nine phases, and remains almost constant in all phases. This advantage is primarily due to the elimination of extensive bilinear pairing and exponentiation operations, making our scheme a more effective and scalable solution for resource-constrained industrial Internet of Things devices.

VII. Conclusion

This paper presented a lightweight, revocable bilateral access control scheme with an equality test to address critical security and efficiency challenges in cloud-enabled IIoT environments. Our proposed RBAC-ET scheme offers several desirable features, including fine-grained bilateral access control, policy confidentiality, equality test functionality, robust user revocation mechanism, and lightweight encryption and decryption. Crucially, it introduces two capabilities previously unaddressed in concert: an efficient equality test for duplicate ciphertext identification and a robust revocation mechanism that ensures both forward and backward secrecy. To the best of our knowledge, RBAC-ET is the first scheme to integrally combine policy-hiding privacy with lightweight, revocable bilateral access control for industrial IoT scenarios. A key innovation in our proposed scheme is the shift from computationally expensive bilinear pairing and exponentiation operations to efficient elliptic curve scalar multiplication operations, which significantly reduces the computational overhead for encryption and decryption. This makes RBAC-ET uniquely practical for deployment on resource-constrained IIoT devices. We have evaluated RBAC-ET through both theoretical analysis and experimental simulations. The results demonstrate that our scheme provides enhanced security functionality while maintaining computational performance comparable to, and often superior to, existing state-of-the-art schemes.

For future work, we aim to further strengthen privacy preservation by transitioning from current adversary assumptions to unconditional privacy. This will be achieved by integrating succinct zero-knowledge proof systems, thereby enhancing the robustness and security of our access control scheme.

References

[1] P. Yao, B. Yan, T. Yang, Y. Wang, Q. Yang, and W. Wang, "Security-enhanced operational architecture for decentralized industrial internet of things: A blockchain-based approach," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 11073–11086, 2024.

[2] J. Cui, F. Wang, Q. Zhang, Y. Xu, and H. Zhong, "Anonymous message authentication scheme for semitrusted edge-enabled iiot," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, pp. 12921–12929, 2021.

[3] L. Wei, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-ta model for fog-based vanets," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 422–436, 2023.

[4] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, "Secure data storage and searching for industrial iot by integrating fog computing and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4519–4528, 2018.

[5] A. Wu, Y. Zhang, J. Zhu, Q. Zhao, and Y. Zhang, "Hierarchal bilateral access control with constant size ciphertexts

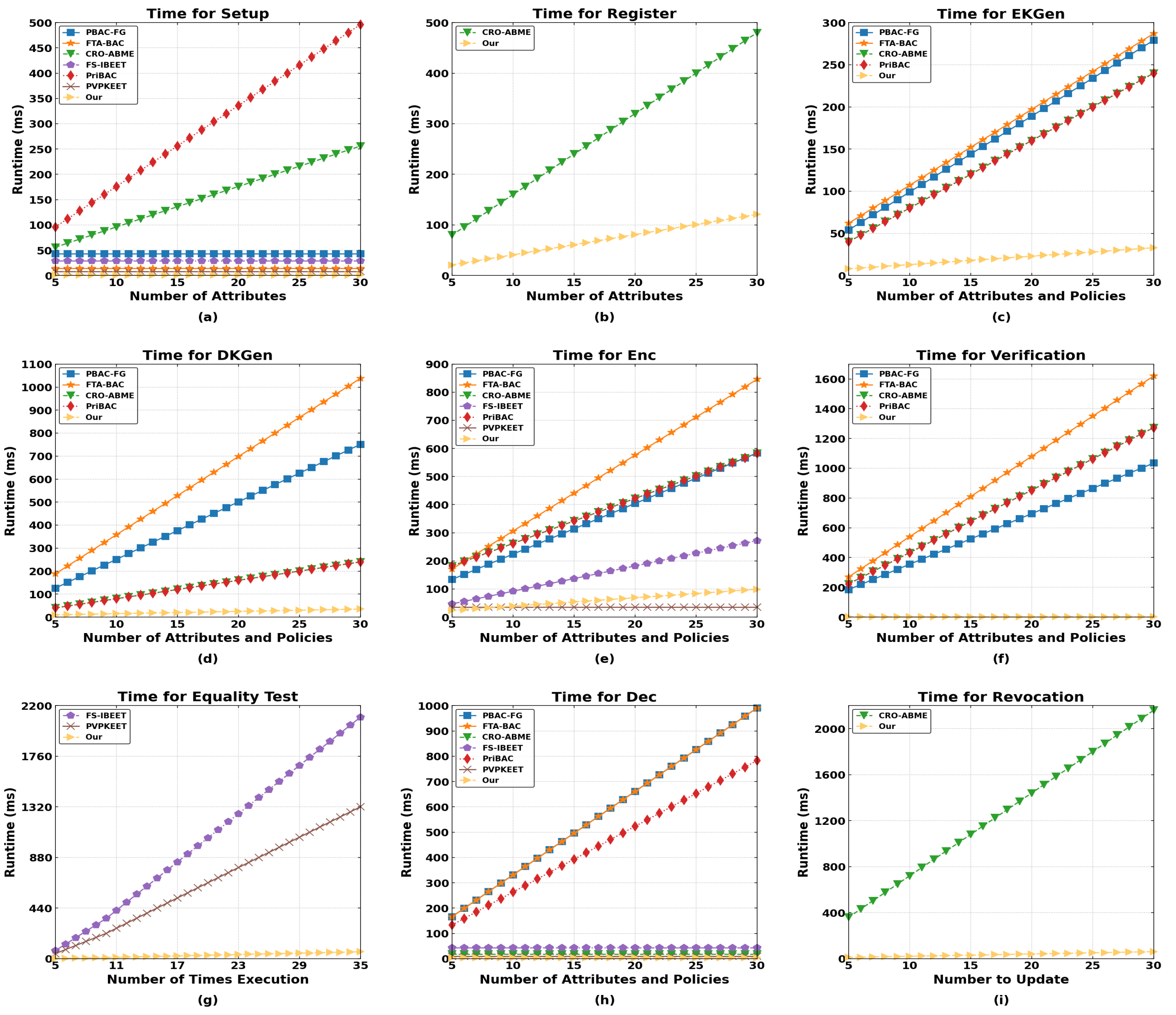


Fig. 5. Experimental performances of the algorithm running time

for mobile cloud computing," *IEEE Transactions on Cloud Computing*, vol. 12, no. 2, pp. 659–670, 2024.

[6] Y. Zhao, H. Yu, Y. Liang, A. Brighente, M. Conti, J. Sun, and Y. Ren, "Secure source identification scheme for revocable instruction sharing in vehicle platoon," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 3671–3684, 2024.

[7] D. Han, N. Pan, and K.-C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 316–327, 2022.

[8] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 478–487, 2020.

[9] J. Wei, X. Chen, J. Wang, X. Huang, and W. Susilo, "Securing fine-grained data sharing and erasure in outsourced storage systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 2, pp. 552–566, 2023.

[10] H. Deng, Z. Qin, Q. Wu, R. H. Deng, Z. Guan, Y. Hu, and F. Li, "Achieving fine-grained data sharing for hierarchical organizations in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1364–1377, 2023.

[11] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient data access control with fine-grained data protection in cloud-assisted iiot," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2886–2899, 2021.

[12] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and*

Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17, pp. 293–310, Springer, 2014.

[13] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Generation Computer Systems*, vol. 115, pp. 486–496, 2021.

[14] Z. Li, J. Lai, J. Han, L. Chen, and X. Yang, "Forward secure equality test for secure data sharing in healthcare systems," *IEEE Internet of Things Journal*, vol. 12, no. 14, pp. 27861–27870, 2025.

[15] Q. Zhang, J. Zhang, N. Yang, and Y. Tian, "Secure and lightweight signcryption scheme with group equality test for heterogeneous wban," *IEEE Internet of Things Journal*, vol. 12, no. 13, pp. 25314–25332, 2025.

[16] X. Zheng, D. Zheng, and Y. Zhang, "Revocable identity-based matchmaking encryption with equality test for smart healthcare," *Sensors*, vol. 25, no. 15, 2025.

[17] W. Li, W. Susilo, C. Xia, L. Huang, F. Guo, and T. Wang, "Secure data integrity check based on verified public key encryption with equality test for multi-cloud storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5359–5373, 2024.

[18] Y. Hou, Y. Cao, H. Xiong, J. Kang, C. H. Foh, and C. Dong, "Heterogeneous broadcast signcryption scheme with equality test for iovs," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 19550–19564, 2024.

[19] S. Chen, J. Li, Y. Zhang, and J. Han, "Efficient revocable attribute-based encryption with verifiable data integrity," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 10441–10451, 2024.

[20] Y. Ming, H. Wang, C. Wang, H. Liu, J. Feng, and K. Li, "Registration-based bilateral fine-grained access control in vehicular social networks," *IEEE Internet of Things Journal*, vol. 12, no. 15, pp. 31986–31997, 2025.

[21] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Security and Communication Networks*, vol. 2017, no. 1, p. 8405879, 2017.

[22] G. Ateniese, D. Francati, D. Nunez, and D. Venturi, "Match me if you can: Matchmaking encryption and its applications," in *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pp. 701–731, Springer, 2019.

[23] B. Chen, T. Xiang, M. Ma, D. He, and X. Liao, "Cl-me: Efficient certificateless matchmaking encryption for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 15010–15023, 2021.

[24] Z. Yan, H. Qu, X. Zhang, J.-L. Xu, and X.-J. Lin, "Identity-based proxy matchmaking encryption for cloud-based anonymous messaging systems," *Journal of Systems Architecture*, vol. 142, p. 102950, 2023.

[25] J. Sun, G. Xu, T. Zhang, X. Yang, M. Alazab, and R. H. Deng, "Privacy-aware and security-enhanced efficient matchmaking encryption," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4345–4360, 2023.

[26] Y. Bao, R. Lu, S. Zhang, X. Cheng, H. Lian, Y. Guan, and W. Qiu, "Lightweight and bilateral controllable data sharing for secure autonomous vehicles platooning service," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 13969–13984, 2023.

[27] C. Zhang, M. Zhao, Y. Xu, T. Wu, Y. Li, L. Zhu, and H. Wang, "Achieving fuzzy matching data sharing for secure cloud-edge communication," *China Communications*, vol. 19, no. 7, pp. 257–276, 2022.

[28] S. Xu, J. Ning, Y. Li, Y. Zhang, G. Xu, X. Huang, and R. H. Deng, "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1064–1077, 2022.

[29] J. Sun, Y. Yuan, M. Tang, X. Cheng, X. Nie, and M. U. Aftab, "Privacy-preserving bilateral fine-grained access control for cloud-enabled industrial iot healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6483–6493, 2022.

[30] X. Hu, L. Wang, L. Gu, and Y. Ning, "A bilateral access control data sharing scheme for internet of vehicles," *IEEE Internet of Things Journal*, vol. 11, no. 22, pp. 36748–36762, 2024.

[31] Q. Huang, C. Wang, and B. Lu, "An efficient and verifiable encrypted data filtering framework over large-scale storage in cloud edge," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 8248–8262, 2024.

[32] T. Wu, X. Ma, C. Zhang, X. Liu, G. Yang, and L. Zhu, "Toward fine-grained task allocation with bilateral access control for intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 14814–14828, 2024.

[33] J. Li, E. Zhang, J. Han, Y. Zhang, and J. Shen, "Ph-mg-abe: A flexible policy-hidden multigroup attribute-based encryption scheme for secure cloud storage," *IEEE Internet of Things Journal*, vol. 12, no. 2, pp. 2146–2157, 2025.

[34] S. Yao, R. V. J. Dayot, I.-H. Ra, L. Xu, Z. Mei, and J. Shi, "An identity-based proxy re-encryption scheme with single-hop conditional delegation and multi-hop ciphertext evolution for secure cloud data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3833–3848, 2023.

[35] M. Zhao, C. Zhang, T. Wu, J. Ni, X. Liu, and L. Zhu, "Revocable and privacy-preserving bilateral access control for cloud data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5389–5404, 2024.

[36] Q.-A. Huang and Y.-W. Si, "Certificateless and revocable bilateral access control for privacy-preserving edge–cloud computing," *IEEE Internet of Things Journal*, vol. 12, no. 8, pp. 10333–10348, 2025.

[37] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.