Attribute-Based Access Control Scheme for Secure Identity Resolution in Prognostics and Health Management

Yunhua He, *Member, IEEE*, Zihe Yan, and Tingli Yuan

***Abstract*—In modern industrial enterprises, the application of identity resolution systems contributes to improving efficiency and simplifying production management. With the development of the Industrial Internet of Things (IIoT), integrating identity resolution and prognostics and health management (PHM) has become a new trend. However, ensuring the confidentiality and integrity of enterprise identity data has become challenging due to flaws in identifier encoding design and the semi-trusted nature of identity resolution platforms. To address these issues, we propose a fine-grained access control scheme for the identity resolution system. Our scheme utilizes a novel identifier encoding method and attribute-based encryption algorithm, enabling flexible data classification and permission management for industry enterprises. Moreover, to combat potential malicious behaviors by users, such as unauthorized access or identity abuse, we leverage Blockchain technology to trace malicious users while safeguarding user privacy. The security of our scheme is formally proven under the decisional bilinear Diffie–Hellman (DBDH) assumption. Comparative experiments demonstrate the advantages of our proposal in terms of time costs and storage overhead over alternative schemes.**

***Index Terms*—Access control, attribute-based encryption (ABE), blockchain, identity resolution, industrial Internet, prognostics and health management (PHM).**

I. Introduction

INDUSTRIAL Internet of Things (IIoT) [1] is a creation of the in-depth integration of new-generation communication technologies and modern industrial technologies. In the context of Industry 4.0 [2], industrial enterprises utilize advanced sensors and data analytics technologies to comprehend and monitor the manufacturing processes. This enables the prediction of machine health trends, giving rise to prognostics and health management (PHM) [3] predictive maintenance systems. To optimize monitoring processes and enhance equipment monitoring efficiency, an increasing number of enterprises choose to employ identity resolution technology [4] as the data portal for PHM systems. This allows for rapid localization and data access of industrial equipment. With the continuous advancement of Internet of Things (IoT) technology, industrial data involved in PHM systems becomes more complex and richer. Some data even possesses a high level of confidentiality. Therefore, it is essential to implement appropriate access control during the identification resolution process to ensure the security of industrial data.

Various internationally accepted identifier encoding methods are referenced in current identity resolution system to improve communication, such as Handler, GS1, and Ecode [5]. However, the current focus of these identity methods lies primarily in improving the efficiency of resolution, with limited consideration for the privacy of identifiers. Only a few standards indicate the possibility of setting secure codes in identifier fields, yet they do not provide specific implementation methods, leaving it up to companies for customization. However, many enterprises, driven by their pursuit of interests, selectively neglect the customization of this field, which makes the reduction of the security of identifier coding and the exposure of identification data become a serious challenge [6]. Meanwhile, with more companies opting to use third-party identity resolution platforms (IRPs), they indirectly store their data in the cloud. Due to the immense value of industrial data, both these platforms and the identity resolution data on cloud services become prime targets for malicious attacks, posing a significant risk of data leakage for enterprises. Therefore, data security and permissions for IRPs become another critical concern.

To compensate for design flaws in identifier encoding, some emerging identifier encoding rules have been proposed, such as decentralized identifier (DID) and blockchain identifier (BID) [7]. These new encoding schemes aim to improve the effectiveness and security of identifying and resolving data in the context of the industrial Internet. However, due to their significant structural changes compared to existing identifier encoding methods, these new schemes may not be effectively compatible with the current practices. Their widespread adoption might require time and industry-wide acceptance.

In addressing data security and permissions for IRPs, various access control technologies have been proposed, among which ciphertext policy attribute-based encryption (CP-ABE) [8] has gained significant attention. CP-ABE provides fine-grained permission assignment by embedding policies in the ciphertext, allowing access only to users whose attributes

Manuscript received 29 January 2024; revised 12 March 2024; accepted 7 April 2024. Date of publication 10 April 2024; date of current version 26 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 6227074758; in part by the Beijing Natural Science Foundation under Grant M21029; and in part by the Opening Foundation of Yunnan Key Laboratory of Blockchain Application Technology under Grant 174009. *(Corresponding author: Yunhua He.)*

The authors are with the School of Information, North China University of Technology, Beijing 100144, China (e-mail: heyunhua@ncut.edu.cn; yanzihe@mail.ncut.edu.cn; ytinsli@gmail.com).

Digital Object Identifier 10.1109/JIOT.2024.3387079

satisfy the policy requirements. To further enhance security, revocable and traceable CP-ABE schemes [9] have been proposed, but they come with the drawback of complex computations, leading to substantial computational costs. Additionally, their tracing of malicious users relies on centralized institutions, which not only introduces the risk of single points of failure but also raises concerns about credibility [10].

The application of blockchain technology has brought new perspectives to address these issues [11]. In blockchain networks, participants, known as nodes, share data and collectively maintain a distributed ledger, utilizing consensus mechanisms to ensure data consistency. By uploading identifiers and data digests to the blockchain, the uniqueness of identifiers and the integrity of identity data can be guaranteed [12]. Additionally, combining blockchain with CP-ABE technology enables distributed permission management and access control, significantly reducing the reliance on centralized institutions. However, since data on the blockchain is shared and persisted, it also introduces the risk of data leakage. Therefore, ensuring data privacy while leveraging the advantages of blockchain becomes an important challenge [13].

In this article, we propose a novel attribute-based access control (ABAC) scheme for secure identity resolution in PHM. The primary goal of this scheme is to enhance the security of PHM data in enterprises, addressing issues, such as inadequate user privacy protection and inefficiency in tracking malicious users. To achieve these goals, first, a secure identifier encoding method is introduced, it enables fine-grained data partitioning and permission settings, ensuring that PHM data can only be accessed by authorized personnel. Second, an access control scheme based on CP-ABE is proposed built upon the proposed identifier encoding method, which features both revocability and traceability. Moreover, blockchain technology is incorporated in this article, significantly enhancing the efficiency and trustworthiness of malicious user trace, meeting the real-time requirements of PHM systems. Throughout this process, we also address user privacy concerns through cryptographic algorithms. The main contributions of this article are as follows.

1) *A Secure Identifier Encoding Method:* In our proposed scheme, we have devised an identifier encoding method that incorporates permission information directly into the identifier and utilizes encryption algorithms to conceal it. This approach enables fine-grained data classification and permission management, ensuring enhanced security and privacy for PHM data.
2) *A Revocable and Traceable Access Control Scheme:* Compared to other schemes, our proposed scheme not only supports the revocation of attributes and users but also enables the direct revocation of identifiers, thereby enhancing the enterprise's management capabilities over identifiers associated with PHM. We have also incorporated blockchain technology to achieve distributed auditing of user behavior, thereby improving transparency and accountability. Additionally, the adoption of an optimized tracing algorithm significantly enhances the efficiency of tracking malicious users, enabling quicker identification and response to potential security threats, while ensuring that user privacy remains uncompromised throughout this process.
3) *Security and Performance:* Our scheme has been demonstrated to be resistant to chosen plaintext attacks (CPAs). Auditability and traceability have also been proven. Performance evaluations show that our scheme outperforms other access control schemes in the IIoT domain in terms of time costs and storage overhead. This ensures that our solution is well-suited to meet the security requirements of enterprises regarding PHM data.

The remainder of this article is organized as follows: Related work is summarized in Section II, followed by preliminaries in Section III. In Section IV, we present the model of our proposed scheme and define the security model. Section V is the construction of our proposed scheme, including the secure identity encoding method. Section VI provides the analysis of the correctness and security of the proposed scheme. Section VII presents the simulation experiments. Finally, the conclusion is drawn in Section VIII.

II. Related Work

With the promotion and use of identity resolution systems, more and more enterprises choose to transform into intelligent factories [14]. They store data in the cloud and use the identity resolution system to associate entities and data, achieving improved management efficiency and reduced labor costs [4]. In this process, the security of the identity resolution system is crucial.

The implementation of access control offers a robust and effective solution to address this problem comprehensively. By carefully managing and partitioning access permissions, it ensures controlled sharing of resources while preventing unauthorized information flow [15]. This approach not only safeguards data security [16] but also establishes a trusted and flexible authorization mechanism [17]. By employing advanced access control models, such as ABAC [18], we can achieve fine-grained control, allowing only users with specific attributes that satisfy predefined policies to access the data. This provides a dynamic and adaptive approach to access management, successfully catering to the evolving demands of an open environment [19]. As a result, access control emerges as a pivotal element in ensuring the confidentiality, integrity, and availability of critical resources, significantly mitigating potential risks and alleviating the burden of exponential growth in permissions and roles [20].

In the realm of cryptography, attribute-based encryption (ABE) [8] has garnered considerable attention as a product of the convergence between cryptography and ABAC. Sahai and Waters [21] introduced the concept of ABE in their scheme. ABE can be categorized into two main variants: 1) key policy ABE (KP-ABE) [22] and 2) CP-ABE [23], based on the representation of access policies. Regarding the policy structure, CP-ABE can be further divided into three models: 1) tree-based [21]; 2) AND-gate-based [24]; and 3) linear secret sharing scheme (LSSS) [25]. The LSSS model extends the capabilities of the tree-based structure and demonstrates

optimal policy expressiveness. Currently, numerous CP-ABE schemes like [26] and [27] have been applied in securing IIoT systems.

To achieve more fine-grained user management, the concept of traceability was initially introduced by Hinek et al. [28] in ABE schemes based on labels, where users' privacy information is linked to their private keys to discourage key leakage. However, this method still cannot trace the specific malicious user. Subsequently, Yu et al. [29] proposed a KP-ABE scheme to prevent user collusion based on the decisional bilinear Diffie–Hellman (DBDH) assumption but does not ensure user anonymity. Li et al. [30] presented an anonymous and accountable CP-ABE scheme by embedding additional user-specific information into the issued private keys, thus implementing an accountability mechanism. Li et al. [31] introduced a multiauthority CP-ABE scheme with an accountability mechanism while reducing reliance on centralized trusted authorities. Dharminder et al. [32] proposed a distributed hierarchical cipher policy-based attribute encryption scheme to enhance the security of electric healthcare records, which can withstand IND-CCA and attribute collusion attacks in distributed environments. However, their scheme can only support user revocation, but cannot support attribute revocation. At the same time, their scheme does not specify how to locate and track malicious users. Katz and Schröder [33] introduced traceability in the context of predicate encryption. However, the additional overhead of introducing traceability in the system grows linearly with the number of users, making it infeasible for large user populations. Liu et al. [34] proposed a white-box traceable CP-ABE system that stores user identity information in an identity table to achieve traceability, but the size of the identity table grows with the number of users. Ning et al. [35] presented a white-box traceable scheme that supports both traceability and a large attribute space without the need for maintaining an identity table, and regardless of the number of users, the efficiency and storage space of key tracing remain constant. Based on this scheme, several access control schemes like [36] and [37] are proposed for security in IIoT. However, these schemes utilize numerous and complex bilinear pairing operations during the trace period and face the risk of a single point of failure.

The integration of blockchain technology with access control techniques has significantly enhanced the flexibility and privacy of access control. Yu et al. [38] proposed a blockchain-based access control scheme that is both revocable and traceable. However, their approach relies on certificate-based user control, resulting in additional storage overhead. In contrast, Ma et al. [39] presented an efficient access control solution for data sharing among intelligent factories in multidomain environments, with support for user revocation. Huo et al. [40] built a trusted platform for an identity resolution system on the blockchain, effectively addressing the risk of a single point of failure and improving overall efficiency. However, it should be noted that their approach may not have fully addressed security concerns.

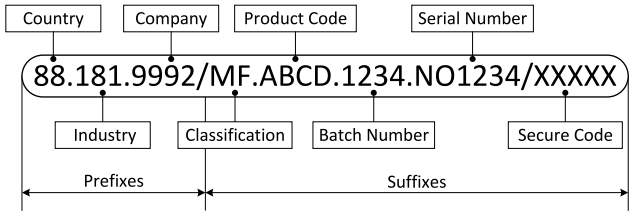

Fig. 1. Typical industrial Internet identity.

III. Preliminaries

A. Identity Resolution System in IIoT

Identity resolution system is a critical technology in the field of IIoT. It is used to resolve and process identifiers for industrial devices, systems, and data. In the context of PHM, a large number of devices, sensors, and systems need to be interconnected and communicate with each other. The purpose of the identification resolution system is to ensure accurate identification, localization, and accessibility of these entities.

These identifiers can be based on various standards and protocols, such as international IoT standards and industrial automation standards. The identity resolution system maintains a registration and mapping table for identifiers, which records the identity of each entity along with its associated information and attributes.

A typical Industrial Internet identity is shown in Fig. 1. The comma symbol ("/") symbol divides the identity into a prefix part and a suffix part, which locate the enterprise and a specific resource belonging to the enterprise, respectively. The comma symbol (".") further distinguishes different identity fields, each representing a distinct meaning, and ultimately they combine to form a complete identity.

Through the identification resolution system, various applications in the IIoT can accurately resolve and locate target entities. For example, when a system needs to retrieve data from a specific sensor, it can utilize the identification resolution system to obtain the identity of that sensor and establish a communication connection based on it. This approach simplifies communication and integration between devices, enhancing the scalability and flexibility of the system.

B. Bilinear Pairing

Let G and G_T be two multiplicative cyclic groups of prime order p. Let g be a generator of G and e be a bilinear map, $e : G \times G \rightarrow G_T$. The bilinear map e has the following properties.

1) *Bilinearity:* $e(P^a, Q^b) = e(P, Q)^{ab} = e(P^b, Q^a)$ and $a, b \in Z_p$.
2) *Nondegenerate:* The $e(g, g) \neq 1$, where the 1 is the identity of group G_T and g is a generator of G.
3) *Computable:* The bilinear pairing function e should be efficiently computable.

C. Linear Secret-Sharing Schemes and Bilinear Map

Let $P = \{P_1, P_2, \ldots, P_l\}$ be a set of participants. We say a secret shared scheme $\mathbb{A}$ on PP is linear when and only if the following two conditions are satisfied.

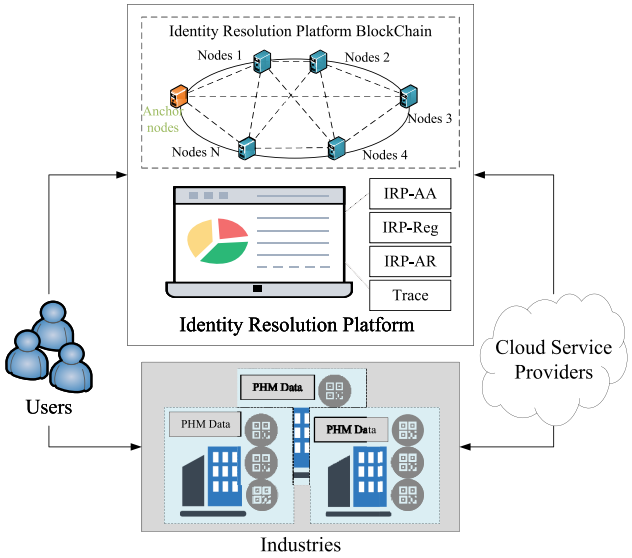

Fig. 2. Architecture of our proposed scheme.

1) The shares of secret value s for each party form a vector over Z_p.
2) The secret sharing structure $\mathbb{A}$ involves a sharing generation matrix M with l rows and n columns. Let ρ be a mapping from $\{1, 2, \ldots, l\}$ to P. Each row M_i in M corresponds to participant $\rho(i)$, where $i \in [1, l]$. Given a column vector $\vec{v} = (s, y_2, \ldots, y_n)^T \in Z_p^n$, where $s \in Z_p$ represents the secret value to be shared and $y_2, \ldots, y_n$ are randomly chosen values from Z_p, the vector $M \cdot \vec{v}$ represents the l shares of $\mathbb{A}$, where the ith share $\lambda_i = (M_i \cdot \vec{v}_i)$ belongs to part $\rho(i)$.

An LSSS possesses the property of linear reconstruction. Assuming an LSSS $\mathbb{A}$ represents an access structure, let S be an authorized attribute set, and define $I = \{i, \rho(i) \in S\}$, where I is a subset of $\{1, 2, \ldots, l\}$. Then, there exists a set of constants $\{\omega_i \in Z_p\}_{i \in I}$ for i in I such that $s = \sum_{i \in I} \omega_i \lambda_i$, and these constants can be found in polynomial time. For any unauthorized set, it is not possible to find a set of constants that satisfies the condition. Therefore, by verifying the equation $\sum_{i \in I} M_i \omega_i = (1, 0, \ldots, 0)$, one can determine whether the attribute set satisfies the access policy.

IV. System Model

In this section, we will introduce the architecture of our proposed secure identity resolution access control scheme based on attributes and then provide the security model.

A. Secure Identity Resolution Access Control Scheme Based on Attributes

Fig. 2 illustrates the architecture of the secure identity resolution access control scheme proposed by us, which consists of four main entities: 1) users; 2) IRP; 3) industries; and 4) cloud service providers (CSPs).

1) *IRP:* As the core of our scheme, the IRP serves as the primary gateway for users to initiate identifier uploads and resolution processes. Its pivotal role involves customizing access control strategy for enterprises while ensuring the utmost security of identification data and preserving user privacy.
2) *Users:* Users of the system comprise identity resolution visitors and identity data owners, both of whom are required to register on the platform to access the respective services. To ensure user credibility, their information will be uploaded to the platform's blockchain. In the event of any malicious behavior being detected, appropriate measures will be taken to trace and revoke the user's access privileges.
3) *CSP:* A CSP is considered an honest but curious entity in the system. It is responsible for storing product data acquired by the enterprise through identity resolution and conducting match tests on behalf of the platform. If the match test yields a successful result, the CSP carries out outsourced decryption. Conversely, if the match test fails, the CSP terminates the execution of the algorithm to maintain security and privacy.
4) *Industrial Enterprises:* The enterprises act as data owners, enabling them to upload their data to the cloud using cloud storage services and create unique identifiers for their PHM data through the IRP. They retain absolute management control over the data.

It is noted that our proposed IRP has five main components.

1) The IRP Blockchain (IRP-BC) is responsible for identity information management and system parameter storage.
2) The IRP attribute authority (IRP-AA) manages user attributes and generates secure parameters.
3) The IRP registrar (IRP-Reg) is responsible for identity data classification and policy binding.
4) The IRP resolution authority (IRP-RA) interfaces with enterprises to resolve subidentities and retrieve corresponding data.
5) The *Trace* module is designed for tracing malicious users, design details will be introduced in the next section.

B. Security Model

In our scheme, the IRP is built on a consortium blockchain where each node is honest and curious, meaning that they will both correctly execute the instructions we give them and try to snoop on our stored identity resources. It is assumed that the communication channel between entities is secure, which can be achieved through the SSL/TSL protocol. In the following, we define a security model through a CPA game between an adversary and a challenger.

1) *Setup:* The challenger $\mathcal{B}$ runs the setup algorithm *Setup* and passes the public key PK to the adversary $\mathcal{A}$.
2) *Phase 1:* Adversary $\mathcal{A}$ asks Challenger $\mathcal{B}$ for the private keys corresponding to the polynomially finite set of attributes $S_1, S_2, \ldots, S_{q_1}$. Challenger $\mathcal{B}$ runs the *KeyGen* algorithm to send these private keys to adversary $\mathcal{A}$.
3) *Challenge:* In this phase, adversary $\mathcal{A}$ selects two equal-length plaintext messages m_0 and m_1 from the message space that will be challenged, and submits them to the

challenger together with an access structure M^* to be challenged. In particular, the set of attributes in stage 1, $S_1, S_2, \ldots, S_{q_1}$ are not satisfied by this access structure. The challenger flips a random coin $b \in \{0, 1\}$ and returns to adversary $\mathcal{A}$ the ciphertext CT^* formed by encrypting m_b under the access structure M^*.

4) *Phase 2:* Repeat the private key query process of Phase 1, where none of the private keys correspond to the set of attributes asked for $S_{q_1+1}, S_{q_1+2}, \ldots, S_{q_1}$ satisfies the access structure M^* to be challenged.
5) *Guess:* In this phase, adversary $\mathcal{A}$ outputs a guess for b $b' \in \{0, 1\}$.

In this attack game, the probability of adversary $\mathcal{A}$ winning is defined as

$$\left|Pr\left[b' = b\right] - 1/2\right|.$$

Definition 1: Our proposed scheme is safe if no adversary can win the above game by a non-negligible margin in polynomial time.

V. Design of Our Proposed Scheme

In this section, we will give a detailed description of the proposed scheme. We first introduce the secure identifier encoding method that has been specifically designed for this system, then provide an overview of our scheme, then the construction of our scheme.

A. Secure Identifier Encoding Method

We propose a novel and secure identifier encoding method for protecting PHM data security. Our approach preserves the existing identification rules while introducing a new security code field that enables the implementation of enhanced security measures. Specifically, we adopt a data division strategy to segment the original identification data into multiple subsets, each aligned with the security level requirements of the respective enterprise. Each subset is assigned a subidentifier that extends and builds upon the original identifier. These subidentities play a vital role in defining the operational privileges, content, and validity period of the associated data set. Moreover, they are bound to a well-defined access policy. All data sets are encrypted via a symmetric encryption key, and only users who comply with the access strategy from our scheme can obtain the symmetric encryption key. This stringent access control mechanism guarantees that data access is confined to authorized entities, ensuring the highest level of security for PHM applications.

For the sake of easier understanding, we first provide the corresponding definitions of our scheme, which are denoted as follows.

Definition 2 *[Identity (ID)]:* In the context of the industrial Internet, identity is a symbol used to uniquely identify a data resource. It can be a string, number, or any other form of data. Identity enables quick localization and access to the corresponding data. The enterprise combines ID with a specific identification carrier, enabling engineer to easily access product information via scanning devices. In our scheme, we define the complete data resource associated with an identity as identity data, represented as $Data_{ID}$. ID and the digest of $Data_{ID}$ will be uploaded to the blockchain.

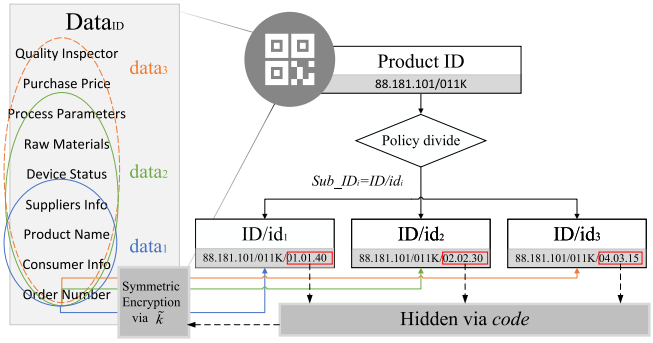

Fig. 3. Example of the proposed secure identity encoding method.

Definition 3 (Subidentity (id)): A subidentity is a symbol used to further divide and categorize $Data_{ID}$. It plays a role similar to the *SecureCode* in Fig. 1 as a part of the suffixes in the identity ID, forming a complete identity in the format of ID/id. The formal construction of *id* is shown below, with three fields comprised: 1) *Authority*; 2) *Content*; and 3) *Time*, separated by periods (.)

$$\text{id} = (\text{Authority.Content.Time}).$$

Here, the *Authority* field represents the user's operational privileges, while the *Content* field denotes the range of content the user can access, which corresponds to a subordinate data resource belonging to $Data_{ID}$. The *Time* field signifies the validity period of user permissions. We define the respective subordinate data resource as data_{id}, which is a subset of $Data_{ID}$. To bolster security, all data_{id} are encrypted using symmetric algorithms in each session, and the symmetric key's validity period is aligned with the *Time* field of the corresponding *id*.

An simplified example is illustrated in Fig. 3, where "88.181.101/011K" represents a product produced by the company "88.181.101" with the identity "011K." The subidentities, including "01.01.40" as id_1, "02.02.30" as id_2, and "04.03.15" as id_3, represent three partitions of the data set. In this example, data_1 corresponds to data accessible to consumers, data_2 represents data accessible to PHM maintenance personnel, and data_3 represents data accessible to enterprise managers. Also, it is important to note that the specific field formats may vary and be more complex depending on different enterprises or application scenarios, and the examples given here are only for ease of understanding.

To ensure timely detection of potential issues by PHM maintenance personnel, we designed the *Authority* field of id_2 as a medium level 02 and set the *Content* field of id_2 to 02, corresponding to information in data_2, and the *Time* field of id_2 to a relatively long duration of 30 min. This means that specialized PHM data, such as equipment status, can be accessed by them for product maintenance, while ordinary consumers are not authorized to view this information. At the same time, they are restricted from accessing more private information, such as the purchase price of products. Following

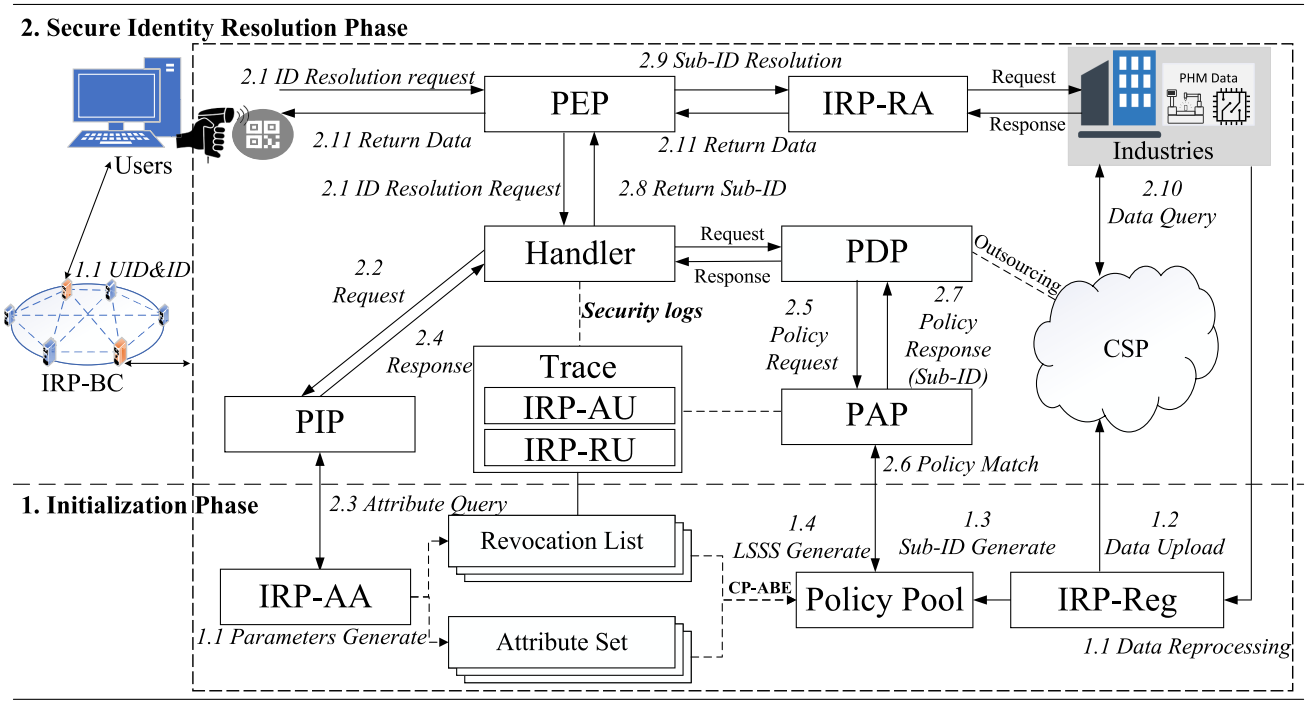

Fig. 4. Implementation process of our proposed scheme.

similar design principles, we apply the same approach for id_1 and id_3.

This encoding method restricts consumer access to user order information and basic product details. It also ensures that neither the industrial producers nor the distributors can obtain the user's personal information through the same identifier.

Furthermore, we believe that the client does not need to know the specific permissions but only needs to process the returned data. Therefore, we hide the *id* via a *code*.

Definition 4 *(Code):* As the following shows, *code* is the result of hashing the ID with a random number *nonce*, which is different for each session

$$\text{code} = \text{Hash}(\text{id}||\text{nonce}). \tag{1}$$

It serves as a secret value and is encrypted via our scheme, where an attribute-based access scheme is applied for its security.

In brief, we encrypt the symmetric key of data_{id} with *code*, and only users who meet the access strategy bound to the *id* can compute and get *code*, then get the symmetric key, and obtain the data_{id} eventually.

We acknowledge that the process of how users are matched with the resources they require has not been explicitly discussed in our approach. This issue involves the user identity authentication process, while our primary focus lies in securing the data. Therefore, for our discussion, we assume that the user has already completed the authentication process and is attempting to access specific data_{id}.

B. Overview of Our Scheme

As an ABAC scheme, our solution retains the traditional modules of ABAC, including the policy information point (PIP), policy enforcement point (PEP), policy administration point (PAP), policy decision point (PDP), and the trusted IRP-AA. Based on this foundation, we have made the following improvements.

1) We decoupled the PEP from other modules and introduced a Handler to process the context, reducing the burden on the PEP.
2) We introduced the resolution authority module of the IRP (IRP-RA), which is responsible for resolving the identities and returning the data to the user.
3) We defined the Trace module for tracing and revoking malicious users, it is composed of an audit component IRP-AU and a regulation component IRP-RU.
4) Finally, we added the IRP-Reg and Policy Pool modules to register and manage *sub-id* and the corresponding policy for finer-grained data partitioning. It is noted that the *Policy Pool* is an abstract concept used to describe the collection of policies stored on the CSP.

Fig. 4 illustrates the two phases of workflow of our scheme, including initialization and Secure Identity Resolution. The blockchain serves as a foundational component providing storage and communication services for the system.

Phase-1-Initialization:

1) Users upload their identity information to IRP-BC for registration and obtain a globally unique identifier. At the same time, enterprises register identifiers for each product and obtain its identifier. The product's identifier maps to the complete information of the product, including its PHM data.
2) IRP-AA contacts PIP to generate private key parameters based on the user's stored identity information and returns them to the user. It also generates public-private

key parameters for enterprises and provides them to the respective enterprises.

3) IRP-Reg assigns subidentities and corresponding policies to each product identity based on the security requirements of the enterprise. It encrypts the PHM data using the enterprise's public key parameters and uploads the data to the CSP. The policies are uploaded to the Policy Pool.

Phase-2-Secure Identity Resolution:

1) The PEP receives the identifier resolution request *ID* from the user and forwards it to the Handler. The Handler then calls the PIP to query the user's identity information and private key, then forwards the information to the PDP.
2) The PDP receives the user information sent by the Handler and calls the PAP to query the encrypted data data_{id} associated with the user's identity and the corresponding access policies. It outsources them to the CSP for policy matching and sends the decision result back to the PEP through the Handler.
3) If the user fails to satisfy the access policies, the identity resolution request is denied. Otherwise, the PEP initiates a subidentifier resolution request to the IRP-RA.
4) The IRP-RA, based on the subidentifier, retrieves the ciphertext stored in the CSP through the enterprise and returns the ciphertext and decryption key to the PEP.
5) The PEP returns the received ciphertext and key from the IRP-RA to the user while recording the user's operations in the security log. In the event of malicious behavior by the user, the Trace module is employed to trace the user and revoke their access privileges.

C. Construction of Our Scheme

Our proposed scheme is composed of nine phases: 1) registration; 2) system setup; 3) T-system setup; 4) data encryption; 5) key generation; 6) match test; 7) data decryption; 8) malicious user trace; and 9) revocation.

1) Registration $\rightarrow$ $(UID, ID, ID/id)$*:* The blockchain system creates and sends user addresses to users using a pseudorandom number generator (PRNG). The IRP generates an unique identity *UID* for each user, and an exclusive identity *ID* for each product. The ID Register module generates SubIDs *id* for each *ID* according to the access control policies set by the industry enterprise, then keeps them in Policy Pool. For the sake of simplification, the following content will prefer to focus on one specific access control policy corresponding to a relevant identity ID/id.

2) $SystemSetup(\lambda) \rightarrow (PP, MSK)$*:* The IRP-AA chooses two multiplicative cyclic groups of prime order p, G, and G_T, where g is the generator of G. The IRP-AA defines a bilinear map $e : G \times G \rightarrow G_T$ and three hash function $H_1 : \{0,1\}^* \rightarrow \{0,1\}^{\text{key}}$, $H_2 : \{0,1\}^* \rightarrow G$, and $H_3 : \{0,1\}^* \rightarrow G_T$. The IRP-AA selects different $g_l, h \in G$ and chooses a random value $V_{P(id)} \in Z_p^*$ as ID's version parameters, then randomly selects $V_x \in Z_p^*$ for each attributes x in access control strategy, which is an attribute set denoted as U. We denote the attribute public key as $PK_x = g^{V_x}$, and set $U_x = H_2(x) \times g^{V_x}$. Denote the ID public key as $PK_{P(id)} = g^{V_{P(id)}}$, the PK_x and $PK_{P(id)}$ will be further used for identity data encryption and attribute revocation. IRP-AA publishes public parameters on IRP-BC as follows:

$$PP = \left\{ g, g^a, g_l, h, g^{\frac{1}{\beta}}, e(g,g)^\alpha, PK_x, PK_{P(id)}, H_1, H_2, H_3 \right\}$$

and keeps $MSK = (g^\alpha, \beta, a)$ in secret.

3) $T\text{-}SystemSetup(\lambda) \rightarrow (TPP, TMS)$*:* To generate key pairs for components in Trace module, the IRP-AU randomly chooses $\sigma \in Z_p^*$ as its secret key sk_a, and calculate $pk_a = h^\sigma \bmod q$ as its public key. Similarly, the IRP-RU randomly chooses $\pi \in Z_p$, as its secret key sk_r, and calculate $pk_r = h^\pi \bmod q$ as its public key. The public key parameters of trace system is denoted as $TPP = \{pk_a, pk_r\}$, and the master secret key is $TMK = \{\sigma, \pi\}$.

4) $Encrypt(PP, data_{id}, (M, \rho)_{id}) \rightarrow CT_{id}$*:* According to the access strategy of identity ID/id, the IRP-AA constructs a $l \times n$ weighted access control matrix M. We define U_{id} as a set of attributes involved in access strategy. The IRP-AA randomly chooses $r_i \in Z_p$ for each attribute i in $U_{P(id)}$. The IRP-AA encrypts the sensitive $data_{\text{id}}$ with a symmetric encryption algorithm $DCT = SymEnc_{\tilde{k}}(data_{\text{id}})$, and sends the symmetric key $\tilde{k}$ to industry enterprise in a secure channel.

The implement of LSSS encryption for encrypting the symmetric key is composed with following steps.

Step 1: IRP-AA randomly chooses a *nonce* and calculates $code = H_3(id||nonce)$ for $data_{\text{id}}$, computes $r = H_1(code)$, and keeps them in secure.

Step 2: The secret value s is defined as $s = H_1(\tilde{k}, code)$, hence we get the secret vector $\vec{v} = (s, v_2, v_3, \ldots, v_n) \in Z_p^*$. Calculate $\lambda_i = \vec{v} \cdot M_i$, M_i as the $i-th$ row vector in M.

Step 3: Inspired by (k,k) secret sharing scheme, we calculate $C_2 = (\tilde{k}) \ \oplus r$ to hide $\tilde{k}$.

Step 4: The complete ciphertext CT_{id} is denoted as follows. The cloud server uses a hash function to process the ciphertext and sends the index of the ciphertext to the blockchain, expressed as $index(CT_{\text{id}}) = Hash(CT_{\text{id}})$. A transaction is simultaneously generated and recorded on the blockchain

$$CT_{\text{id}} = \Bigg\{ M, \quad C = code \cdot e(g,g)^{\alpha s}$$
$$C_1 = g^{\frac{s}{\beta}}, C_2 = \left(\tilde{k}\right) \ \oplus r, V = g^{aV_{P(id)}}$$
$$\forall i \in U_{P(id)}, D_i = g^{\frac{r_i}{\beta}}, C_i = g_l^{\frac{\lambda_i}{\beta}} (U_i \cdot g^{V_{P(id)}})^{\frac{-r_i}{\beta}} \Bigg\}.$$

5) $KeyGen(PP, MSK, UID, S) \rightarrow SK$*:* When the user registers, the IRP-AA collects its attribute information and stores them in PIP, letting S be the user's attribute set. We define $c = IDEnc(UID) \in Z_p^*$ as the user's identity parameter. IRP-AA randomly chooses $\mu \in Z_p^*$ and calculate $K = {g_l}^{\beta\alpha} g^{(a+c)\mu}$, $K_1 = c$, $L = g^{(a+c)\mu}$, $L_1 = g^{(a+\mu)V_{P(id)}}$, and $L_2 = g^{\mu V_{P(id)}}$. For each attribute x in S, IRP-AA calculates $K_x = {U_x}^{(a+c)\mu} g^{V_{P(id)}(a+c)\mu}$. The complete user's private key is

$$SK = \{K, K_1, L, L_1, L_2, \{K_x\}_{x \in S}\}.$$

Then, IRP-AA generates the user's public key as $pk_u = g_l^{K_1}$ and uploads it to IRP-BC.

6) MatchTest$(PP, CT) \rightarrow \perp$ *or Continue:* Two equations are designed for match test, one is to check whether the identity ID is valid, the other one is to verify whether the user's attribute set S satisfies the access strategy for $data_{\mathrm{id}}$. The execution of these two algorithms is outsourced to CSP for optimizing the computational overhead of users.

The implementation of MathTest is composed of the following steps.

Step 1: Since the outsourced cloud servers may not be entirely trustworthy, the IRP-AA randomly chooses a secret $z \in Z_p$ to generate the transform private key TSK for the user. The TSK is denoted as follows:

$$TSK = \left\{ \left(K' = g^{\frac{\beta\alpha}{z}} g_l^{\frac{(a+c)\mu}{z}}, K_1' = \frac{c}{z} \right. \right.$$
$$L' = g^{\frac{(a+c)\mu}{z}}, L_1' = g^{\frac{(a+\mu)V_{P(id)}}{z}}, L_2' = g^{\frac{\mu V_{P(id)}}{z}}$$
$$\left. \left. \forall x \in S, K_x{}' = U_x^{\frac{(a+c)\mu}{z}} g^{\frac{V_{P(id)}(a+c)\mu}{z}} . \right) \right\}$$

Step 2: The CSP invoke (2) to check the validity of identity ID. If the equation holds, it indicates that the version parameter $V_{P(id)}$ embedded in the user's SK is consistent with the latest and that the ID is still valid

$$L_1' = L_2' \cdot V^{\frac{1}{z}}. \tag{2}$$

Step 3: If the identity is valid, the CSP will first calculate trans-ciphertext T, which is denoted as the following (3). It is noted that only the user whose attribute set S satisfies the access policy can keeps the equation hold. In this step, Lagrange interpolation formula is used to find coefficients $\{\omega_i | i \in I\}$ such that $\sum_{i \in I} \omega_i \lambda_i = s$, where $I = \{i : \rho(i) \in S\} \subset \{1, 2, \ldots, l\}$. Otherwise, it outputs $\perp$

$$T = \frac{e(C_1, K')}{e\left(\prod_{i \in I} C_i^{\omega_i}, L'\right) \prod_{i \in I} e(D_i^{\omega_i}, K'_{\rho(i)})} = e(g, g)^{\frac{\alpha s}{z}}. \tag{3}$$

CSP returns T to IRP-AA.

7) Decrypt$(PP, CT_{id}, SK) \rightarrow Data_{id}$*:* The IRP-RA invokes C from CT_{id} to calculates $code = C/T^{-z}$. Then, calculate and return symmetric key $\tilde{k}$ to the user. The $\tilde{k}$ is denoted as

$$\tilde{k} = C_2 \oplus H_1(code) = \left(\tilde{k}\right) \oplus H_1(code) \oplus H_1(code) = \tilde{k}.$$

The user decrypts DCT with $\tilde{k}$ and obtains the plaintext $data_{\mathrm{id}}$.

8) Malicious User Trace: If an user inadvertently or intentionally causes key leakage, they must be traced and added to the revocation list. To achieve this, we propose a privacy-preserving trace mechanism. As illustrated in Fig. 5, it mainly consists of four stages: 1) Key Sanity Check; 2) Logs Upload; 3) Logs Audit; and 4) User Trace.

Stage-1-Key Sanity Check: The key integrity check serves as the first phase of the tracking process and occurs before the user proceeds with authorized access in our scheme. The client, which can be considered as the user terminal, sends

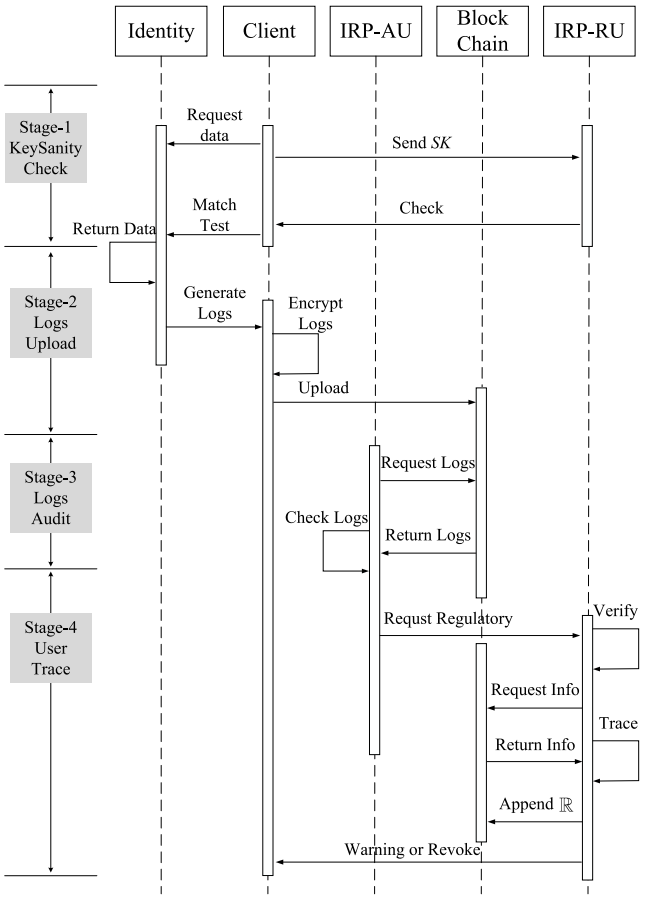

Fig. 5. Workflow of the proposed privacy-preserving trace mechanism.

the user's private key to IRP-RU for an integrity check with the *KeySanityCheck* algorithm. This algorithm consists of the following four parts.

1) $K_1 \in Z_p, K, L, L_1, L_2, K_x \in G$.
2) $e(g^a \cdot L_2, g) = e(g, L_1) \neq 1$.
3) $e(g^{\frac{1}{\beta}}, K) = e(g, g)^{\alpha} \cdot e(g^{\frac{1}{\beta}}, L) \neq 1$.
4) $\exists x \in S$, we have $e(L_2, g^{K_1} g^a) e(H_2(x), L) e(g, g^{V_x}) = e(g, K_x) \neq 1$.

If any of the parts do not hold, meaning the user's private key fails to pass the *KeySanityCheck* algorithm, it indicates that the key is not well-formed and thus not generated by IRP-AA. In this case, the key can be considered invalid or unrecognized. Otherwise, if the key is valid, the system will continue to perform the process to determine if the user's private key matches the resource's access policy.

Stage-2-Logs Upload: If the user's attributes satisfy the access policy, the user will access the resource and generate secure log entries. These logs will be recorded by their clients. To ensure user privacy, the client randomly selects $\eta \in Z_p$, computes a temporary key $Ek_u = g_l^{\eta}$, and calculates $X = e(pk_u, g_l^{\eta})$. The user's one-time address is then computed as $Addr_p = g^{H_1(x)}$, and the transaction address is outputted as $(Addr_p, Ek_u)$. The secret key of transaction address is denoted as $Addr_s = H_1(e(Ek_u, g_l^{K_1}))$.

Next, the client inputs the public parameters PP and IRP-AU's public key, selects random numbers $\theta, \delta \in Z_p$, and computes $c_1 = h^{\theta} \bmod q$, $c_2 = \bar{o}pk_a^{\theta}$, $c_3 = g_l^{\delta} \bmod p$, and

$c_4 = pk_a pk_r^{\delta} \bmod p$. Here, $\bar{o}$ represents the user's operation parameters in logs, such as read, write, delete, etc. Then, the user uploads encrypted secure logs (c_1, c_2) and (c_3, c_4) along with the transaction address to IRP-BC.

It is worth noting that due to the nature of the blockchain, in our scheme, users are unable to modify their operation logs, as they are encrypted and uploaded in real-time to the blockchain by the client.

Stage-3-Logs Audit: The IRP-AU audits the data on the IPR-BC. It takes as input its own private key sk_a, the ciphertext (c_1, c_2), and obtains the user's operational parameters $\bar{o}' = (c_2/c_1^{sk_a}) \bmod q$. If $\bar{o}'$ is consistent with what the policy (note that the policy for auditing is often individually set by the industry enterprises) calls for, then it takes no action. If the user exhibits malicious behavior, albeit possibly unintentional, the IRP-AU will identify the user as malicious and then proceeds to send a trace request to the IRP-RU.

As could be observed, in our scenario, regardless of whether an user is malicious or not, their identity information is not available to the Auditor, which enhances the privacy of the user's privileges.

Stage-4 User Trace: Once the malicious user is detected, the IRP-AU signs the transaction address with its private key and sends it to the IRP-RU. It randomly selects $\varpi \in Z_p$, then sets $\bar{m} = H_1(Addr_p, Ek_u, (c_3, c_4))$ and compute $\bar{t} = h^{\varpi} \bmod q$, $\bar{s} = \varpi^{-1}(\bar{m} - \bar{t} \cdot sk_a) \bmod (q-1)$. $\bar{R} = (m, t, s)$ is the request with the IRP-AU's signature.

When receiving a trace request from IRP-AU, IRP-RU first verifies the identity of IRP-AU by using the public key pk_a. If the equation $pk_a^{\varpi} \varpi^{\bar{s}} = h^{\bar{m}} \bmod q$ holds, the request is accepted, and the trace procedure continues. Otherwise, the request is discarded.

To identify malicious users, the IRP-RU utilizes the transaction address $Addr_p$, the user's public key pk_u and the ciphertext of the user's public key (c_3, c_4) as inputs. Then, calculates $pk'_u = (c_4/c_3^{sk_r}) \bmod p$ and $Addr'_p = g_l^{H(e(Ek_u, pk_u))}$. The UID of a malicious user can be obtained by $IDDec(K_1)$.

Depending on the level of harm caused by the user's malicious actions, IRP-RU issues varying warnings to the user or may proceed to directly revoke their privileges if deemed necessary.

9) Revocation: Our solution has the capability of implementing fine-grained revocation, which can be divided into three specific types: 1) malicious user revocation; 2) attribute revocation; and 3) product identity revocation.

Type-1-Malicious User Revocation: The malicious user trace process mentioned above can be seen as a discovery process for identifying users who may need to be revoked. If an user's behavior is deemed too harmful, we will not only issue a warning but also add them to the revocation list$\mathbb{R}$.

Type-2-Attribute Revocation: When revoking attribute *att* from the system (corresponding to attribute version V_{att}), IRP-AA first generates a new random version V'_{att} for the attribute and then updates the current attribute key PK_{att} to PK'_{att}.

IRP-AA generates an upgrade key UUK for all users who possess the attribute att, which is calculated as

$$UUK = PK_x^{(V_x{}' - V_x)(a+c)\mu}$$

then sends it to the respective IRP-AA.

The IRP-AA updates the key component K_{att} associated with the revoked attribute of each user's private keys to $\tilde{K}_x = K_{\text{att}} \cdot UUK$ and keeps them in PIP while keeping other parts unchanged.

Since the ciphertext components c are different for each user, their respective UUK will also be different. This ensures that no user can use someone else's upgrade key to upgrade their key.

Meanwhile, IRP-AA updates the ciphertext component C_{att} for CT_{id} containing the attribute att. The upgrade key UCK for ciphertext is denoted as

$$UCK = PK_{\rho(i)}{}^{-(V_{\rho(i)}{}' - V_{\rho(i)})r_i}.$$

The re-encrypted ciphertext component is denoted as $\tilde{C}_{\text{att}} = C_{\text{att}} \cdot UCK$.

Type-3-Product ID Revocation: In practical production environments, it is highly possible for a company to revoke the identifiers of outdated products. However, it may not be feasible to physically destroy a barcode or QR code, which would result in significant manual effort.

In our proposed scheme, we address this challenge by introducing the ID version attribute $V_{P(id)}$ for each product to control the versions of product identities. In a normal identification scenario, $V_{P(id)}$ is an element on Z_p. However, when a company wants to revoke the identification information of a particular product, they can simply set $V_{P(id)} = 0 \in Z_p$ and update the corresponding $PK_{P(id)}$. As a result, the identification of all products within the same batch will become unresolvable.

By utilizing this approach, the company can effectively revoke the identification codes of specific products without the need for physical destruction. This provides a practical solution to manage revocations cost-effectively while maintaining control over the product identification process.

VI. Analysis

A. Correctness Analysis

The correctness of our proposed scheme is verified as follows.

First, assuming the *ID* user visited is valid, we verify the correctness of Step 2 in the *MatchTest* algorithm by performing (2)

$$L'_1 = g^{\frac{(a+\mu)V_{P(id)}}{z}} = g^{\frac{\mu V_{P(id)}}{z}} \cdot g^{\frac{aV_{P(id)}}{z}} = L'_2 \cdot V^{\frac{1}{z}}.$$

Then, assuming that the user's attributes satisfy the corresponding access strategy, we verify the correctness of Step 3 in the *MatchTest* algorithm by performing (3)

$$T = \frac{e(C_1, K')}{e\left(\prod_{i\in I} C_i^{\omega_i}, L'\right) \prod_{i\in I} e\left(D_i^{\omega_i}, K'_{\rho(i)}\right)}$$
$$= \frac{e\left(g^{\frac{s}{\beta}}, g^{\frac{\beta\alpha}{z}} g_l^{\frac{(a+c)\mu}{z}}\right) \cdot \left(\prod_{i\in I} e\left(g^{\frac{r_i\omega_i}{\beta}}, U_{\rho(i)}{}^{\frac{(a+c)\mu}{z}} g^{\frac{V_{P(id)}(a+c)\mu}{z}}\right)\right)^{-1}}{e\left(\prod_{i\in I}\left(g_l^{\frac{\lambda_i}{\beta}} \cdot \left(U_{\rho(i)} \cdot g^{V_{P(id)}}\right)^{-\frac{r_i}{\beta}}\right)^{\omega_i}, g^{\frac{(a+c)\mu}{z}}\right)}$$

$$= \frac{e(g,g)^{\frac{s\alpha}{z}} \cdot e(g,g_l)^{\frac{(a+c)\mu s}{z\beta}} \cdot \left(\prod_{i\in I} e\left(g^{\frac{r_i\omega_i}{\beta}}, \left(U_{\rho(i)} \cdot g^{V_{P(id)}}\right)^{\frac{(a+c)\mu}{z}}\right)\right)^{-1}}{e\left(\prod_{i\in I} g_l^{\frac{\lambda_i\omega_i}{\beta}}, g^{\frac{(a+c)\mu}{z}}\right) \cdot e\left(\prod_{i\in I}\left(U_{\rho(i)} \cdot g^{V_{P(id)}}\right)^{-\frac{r_i\omega_i}{\beta}}, g^{\frac{(a+c)\mu}{z}}\right)}$$

$$= \frac{e(g,g)^{\frac{s\alpha}{z}} \cdot e(g,g_l)^{\frac{(a+c)\mu s}{z\beta}}}{e(g,g_l)^{\frac{(a+c)\mu}{z\beta}\sum_{i\in S}(\omega_i\lambda_i)}} = \frac{e(g,g)^{\frac{s\alpha}{z}} \cdot e(g,g_l)^{\frac{(a+c)\mu s}{z\beta}}}{e(g,g_l)^{\frac{(a+c)\mu s}{z\beta}}}$$

$$= e(g,g)^{\frac{\alpha s}{z}}.$$

The symmetric key $\tilde{k}$ can be decrypted by computing $code = C/.T^{-z}$, $\tilde{k} = C_2 \oplus H_1(code) = (\tilde{k}) \oplus H_1(code) \oplus H_1(code) = \tilde{k}$. Hence, the correctness of our scheme can be verified.

B. Security Analysis

Theorem 1: If the assumption of the decisional q-Parallel BDHE [41] holds, then our proposed scheme is secure under the selective access policy and CPA model.

Proof: Let's assume the existence of an adversary $\mathcal{A}$, who can selectively break the proposed scheme with a non-negligible advantage $\varepsilon = Adv_{\mathcal{A}}$ in the aforementioned security model, where the challenge matrix is denoted as M^* $(l^* \times n^*)$. In this case, we can construct a simulator $\mathcal{B}$ that can break the decisional q-Parallel BDHE assumption with non-negligible advantage.

Initialization: Simulator $\mathcal{B}$ takes the inputs y and T from the decisional q-Parallel BDHE assumption. For the sake of differentiation, we set $\vec{y} = (g, g^S, g^{\tau}, g^{\tau^2}, \ldots, g^{\tau^q}, g^{\tau^{q+2}}, \ldots, g^{\tau^{2q}})$ and $T = e(g,g)^{s\tau^{q+1}}$. The adversary $\mathcal{A}$ submits an access policy (M^*, ρ^*) where the challenge matrix M^* has a column size of n^*.

Setup: $\mathcal{B}$ In this phase, $\mathcal{B}$ randomly chooses $\alpha', a \in Z_p^*$ and calculates $e(g,g)^{\alpha} = e(g,g)^{\alpha'} e(g^{\tau}, g^{\tau^q})$, which implies $\alpha = \alpha' + \tau^{q+1}$. The $\mathcal{B}$ selects a random oracle U and establishes a list. When U_x is called, if U_x already exists in the list, $\mathcal{B}$ directly returns the result. If U_x does not exist in the list, $\mathcal{B}$ randomly selects $U_x' \in Z_p$. Let X be the set of i that satisfy $\rho^*(i) = x$. Then, $\mathcal{B}$ sets $g_l = g^{\tau}$, $U_x = g^{U_x'} \prod_{i\in X} g^{\tau M_{i,1}^*/b_i} \cdot g^{\tau^2 M_{i,2}^*/b_i}, \ldots g^{\tau^{n^*} M_{i,n^*}^*/b_i}$. If X is empty (denoted by $\emptyset$), $\mathcal{B}$ sets $U_x = g^{U_x'}$. Since U_x' is randomly chosen, the above parameters are randomly distributed.

Phase 1: In this phase, simulator $\mathcal{B}$ answers the adversary's queries about the private key for attribute set S, where S does not satisfy the access matrix M^*.

The simulator first selects random values $r, c \in Z_p$, and $V_{P(id)} \in Z_p^*$, and finds a vector $\omega = (\omega_1, \omega_2, \ldots, \omega_{n^*}) \in Z_p^{n^*}$ such that $\omega_1 = -1$ and for all i satisfying $\rho^*(i) \in S, \omega \cdot M_i^* = 0$. According to the definition of LSSS, such a vector must exist. Note that if such a vector does not exist, the length of the vector $(1, 0, \ldots, 0)$ is the length of the attribute set S. The simulator sets

$$L = g^{(a+c)\mu} = g^{\beta r} \prod_{i=1}^{n^*} \left(g^{\beta(\tau^{q+1-i})}\right)^{\omega_i}$$

where $\mu = (\beta/a + c)(r + \omega_1\tau^q + \omega_2\tau^{q-1} + \cdots + \omega_{n^*}\tau^{q-n^*+1})$.

It can be observed that $g_l{}^{(a+c)\mu}$ includes the algebraic term $g^{-\beta\tau^{q+1}}$ by defining μ in this way, but this term can be eliminated by multiplying with $g^{\beta\alpha} = g^{\beta\alpha'} g^{\beta\tau^{q+1}}$ in the process of constructing K.

Simulator $\mathcal{B}$ calculates K as follows:

$$K = g^{\beta\alpha} g_l{}^{(a+c)\mu} = g^{\beta\alpha'} g_l{}^{\beta r} \prod_{i=2}^{n^*} g_l{}^{\beta(\tau^{q+2-i})^{\omega_i}}.$$

To compute K_x for $x \in S$, if there exists no i satisfying $\rho^*(i) = x$, then K_x can be defined as $K_x = L^{U_x^*}$. In the case where $x \in S$ and x appears in the access structure, it must be ensured that terms in the form of $g^{\beta\tau^{q+1}}$ can be simulated. Additionally, since $\omega \cdot M_i^* = 0$, these terms can be eliminated.

Let X be the set of i's satisfying the condition $\rho^*(i) = x$. The simulator creates K_x as follows:

$$K_x = U_x{}^{(a+c)\mu} g^{V_{P(id)}(a+c)\mu} = L^{U_x^*} L^{V_{P(id)}} \prod_{i\in X}\prod_{j=1}^{n^*} \left(g^{(\tau^j/b_i)\beta r} \cdot \prod_{k=1,k\neq j}^{n^*} \left(g^{\tau^{q+j-k+1}/b_i}\right)^{\beta\omega_k}\right)^{M_{i,j}^*}.$$

Challenge: The adversary submits two messages m_0 and m_1 to $\mathcal{B}$. Simulator $\mathcal{B}$ randomly selects $b \in \{0,1\}$ and computes $C = R^* \times T \times e(g^s, g^{\alpha'})$, $C_1 = g^{\frac{s}{\beta}}$, $C_2 = m_b \oplus H_1(R^*)$, and $V = e(g,g)^{aV_{P(id)}}$. Then, $\mathcal{B}$ randomly selects values $t_2, t_3, \ldots, t_n$ and uses the vector $\vec{v} = (s, s\tau + t_2, \ldots, s\tau^{n-1} + t_n) \in Z_p^{n^*}$ to share the secret. Additionally, the simulator chooses random values $r_1', r_2', \ldots, r_l' \in Z_p^*$. For $i = 1, 2, \ldots, n$, T_i is defined as the set of all $k \neq i$ such that $\rho^*(i) = \rho^*(k)$. The ciphertext components in the challenge are constructed as

$$D_i = g^{\frac{r_i}{\beta}} = g^{-\frac{r_i'}{\beta}} g^{-\frac{sb_i}{\beta}}$$

$$C_i = g_l{}^{\frac{\lambda_i}{\beta}} \left(U_{\rho(i)} \cdot g^{V_{P(id)}}\right)^{\frac{-r_i}{\beta}} = U_{\rho(i)}^{r_i'} g^{V_{P(id)} r_i'} \left(\prod_{j=2}^{n^*} (g_l^{\tau})^{M_{i,j}^* y_j'/\beta}\right) \left(g^{b_i \cdot s}\right)^{V_{P(id)} - z_{\rho_{(i)}^*}} \cdot \left(\prod_{k\in T_i}\prod_{j=1}^{n^*} \left(g^{\tau^j \cdot s \cdot (b_i/b_k)}\right)^{M_{k,j}^*/\beta}\right).$$

Phase 2: Identical to *Phase 1*.

Guess: $\mathcal{B}$ makes a guess about b to $\mathcal{A}$ as b'. If the guess b' is equal to b, the $\mathcal{B}$ will export 1, indicating that T is a valid component in the q-parallel BDHE game. On the other hand, if b' is not equal to b, T will be considered a completely random element, and $\mathcal{B}$ will export 0.

Theorem 2: Our proposed scheme is auditable, if in the case that both the user and the IRP-AU are honest, the IRP-AU can audit the security logs encrypted by the user's client and can check the user's operation records.

TABLE I
COMPARISONS OF SECURITY PROPERTIES BETWEEN SCHEMES

Schemes	Attribute Revocation	Traceability	Out Dec	Blockchain	Access Structure	Auditability	Identifier Revocation
Xiong *et al.*[42]	✓	×	✓	✓	LSSS	×	×
Hahn *et al.*[43]	×	✓	×	×	AND	×	×
Li *et al.*[36]	✓	✓	✓	×	AND	×	×
Yu *et al.*[38]	✓	✓	✓	✓	TREE	×	×
Ours	✓	✓	✓	✓	LSSS	✓	✓

Proof: As to the ciphertext (c_1, c_2), where $c_1 = h^{\theta} \bmod q$ and $c_2 = \bar{o} pk_a^{\theta}$, IPR-AU utilize its secret key and calculate

$$\begin{aligned} & c_2 \left(c_1^{sk_a}\right)^{-1} \bmod q \\ &= \bar{o} \cdot pk_a^{\theta} \left(h^{\theta \cdot sk_a}\right)^{-1} \bmod q \\ &= \bar{o} h^{\theta \cdot sk_a} h^{-\theta \cdot sk_a} \bmod q \\ &= \bar{o}. \end{aligned}$$

Therefore, by inputting the ciphertext (c_1, c_2) encrypted by the IRP-AU's public key pk_a, the IPR-AU can use sk_a to decrypt the secure logs for $\bar{o}$ and judge whether malicious behaviors happened. That is, our scheme is auditable.

Theorem 3: Our proposed scheme is traceable, If the pk_u is calculated by the user's private key, and the public key of the transaction address is encrypted by the public key of the IRP-RU to generate the ciphertext, the IRP-RU can trace the UID of a malicious user.

Proof: The user public key pk_u, calculated from the user's private key, is given by $pk_u = g^{K_1}$. The user's identity can be derived from the transaction address of the transaction recipient. In this case, the user's transaction address $(Addr_p, Ek_u)$ is defined as $Addr_p = g^{H_1(x)}$, where $X = e(pk_u, g_l^{\eta})$, and $Ek_u = g^{\eta}$. The proof of correctness is as follows:

Let $X' = e(Ek_u, pk_u)$, we have

$$\begin{aligned} Addr_p{}' &= g^{H_1(X')} = g^{H_1(e(Ek_u, pk_u))} = g^{H_1\left(e(g^{\eta}, g_l^{K_1})\right)} \\ Addr_p &= g^{H_1(X)} = g^{H_1(e(pk_u, g_l{}^{\eta}))} = g^{H_1\left(e(g_l^{K_1}, g^{\eta})\right)}. \end{aligned}$$

According to bilinear properties, we can easily get $Addr_p = Addr_p{}'$. Hence, we can get the transaction address.

As to the ciphertext (c_3, c_4), where $c_3 = g_l^{\delta} \bmod p$ and $c_4 = pk_a pk_r^{\delta} \bmod p$, IPR-RU utilize its secret key and calculate

$$\begin{aligned} & c_4 \left(c_3^{sk_r}\right)^{-1} \bmod q \\ &= pk_u pk_r^{\delta} \left(g_l{}^{\delta \cdot sk_r}\right)^{-1} \bmod q \\ &= pk_u g_l{}^{\delta \cdot sk_r} g_l{}^{-\delta \cdot sk_r} \bmod q \\ &= pk_u. \end{aligned}$$

C. Evaluation

Therefore, by inputting the ciphertext (c_3, c_4) encrypted with the regulatory authority's public key pk_r, the regulatory authority can decrypt it and obtain the public key of the malicious user, thus tracing its UID, demonstrating the traceability of our proposed scheme.

TABLE II
SIMULATION EXPERIMENT PARAMETERS

Parameter	Value	Parameter	Value	Parameter	Value
n	$5 \sim 50$	q	$64bytes$	$trial$	40
E_1	$9.2ms$	E_T	$2.1ms$	P	$18.4ms$
χ_1	$128bytes$	χ_2	$128bytes$	χ_3	$20bytes$

VII. EVALUATION AND SIMULATION EXPERIMENTS

In this section, we will compare our scheme with four other access control schemes in the IIoT domain, denoted as Xiong et al. [42], Hahn et al. [43], Li et al. [36], and Yu et al. [38]. We begin by conducting a functional evaluation of these five schemes, and the results are presented in Table I. Subsequently, we perform simulation experiments on all five schemes using the parameters listed in Table II.

The comparison result in Table I indicates that while Li et al. [36], Yu et al. [38], and our scheme all support attribute revocation, user tracing, and outsourced decryption, Li et al. [36]'s scheme does not utilize blockchain technology, which introduces a risk of single point of failure. Xiong et al. [42]'s scheme, although it adopts blockchain technology and implements attribute revocation, lacks the capability for user tracing. Hahn et al. [43]'s scheme also achieves user traceability but performs moderately in other aspects. Additionally, the solutions of Hahn et al. [43] and Li et al. [36] both use AND-gate access structures, which significantly reduces the expression ability of the strategy. Yu et al. [38] adopts a TREE structure similar to Dharminder et al. [32], which has better strategy expression capabilities. However, there is still a certain gap compared with Xiong et al. [42] and our solution that adopts the LSSS structure.

In contrast, our scheme takes advantage of blockchain technology, which not only supports a privacy-preserving secure log auditing mechanism but also enables real-time monitoring of malicious behaviors. In the subsequent discussion, we will provide the time overhead for each stage of the malicious user trace period.

A. Simulation Experiments

To implement our simulation, we utilized a Java-based pairing encryption package [44], which is built on top of a C library for pairing-based encryption. We employed an elliptic curve of type A, represented by the equation $Y^2 = X^3 + X$, with a group order of 160 bits. The hardware specifications used in this experiment are as follows: CPU: 12th Gen Intel Core i7-12700H 2.30 GHz, RAM: 16 GB. The experiment was conducted on a laptop running Windows 11.

TABLE III
PERFORMANCE COMPARISON OF TIME COSTS BETWEEN SCHEMES

Schemes	System setup	Encryption	Key generation	Decrpytion
Xiong *et al.*[42]	$(2l+4)E_1 + E_T$	$(3l+2)E_1$	$(4\vartheta+5)E_1$	$6P + (d+1)E_1 + E_T$
Hahn *et al.*[43]	$(l+2)E_1$	$2E_1 + lP$	$(2\vartheta+3)E_1$	$(2d+1)P + (2d+1)E_T$
Li *et al.*[36]	$(2l+2)E_1 + 2P$	$(2l+6)E_1 + 2E_T$	$(2\vartheta+5)E_1$	$4P + (d+2)E_T$
Yu *et al.*[38]	$(l+2)E_1 + (l+2)E_T + P$	$(2l+6)E_1 + 2E_T$	$(3\vartheta+6)E_1$	$6P + (5d+4)E_T$
Ours	$(l+5)E_1 + E_T + P$	$(2l+1)E_1$	$(2\vartheta+5)E_1$	$3P + (d+2)E_T$

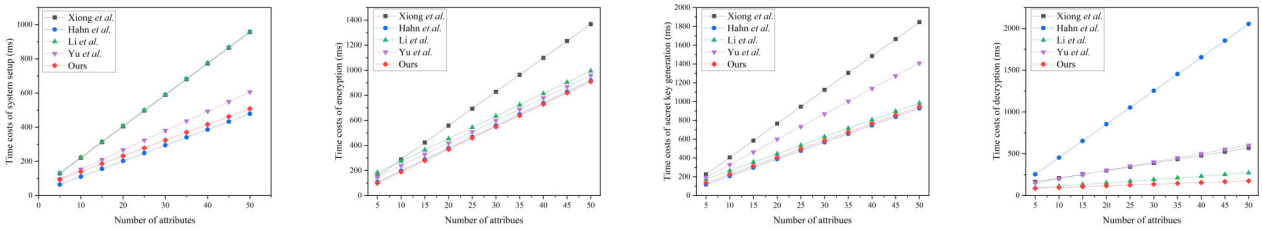

Fig. 6. Time costs. (a) System setup. (b) Encryption. (c) Key generation. (d) Decryption.

The experiments were conducted to compare the time costs and storage overhead between our schemes and others. The results are presented in Table III. It is noted that we let the n represents the number of system attributes, l represents the number of attributes in the access policy, ϑ represents the number of attributes in the user's secret key, and d represents the number of compliant attributes. In storage overhead context, the χ_1, χ_2, and χ_3 are elements taken from groups G, G_T, and Z_p, respectively. In the time costs context, the E_1 and E_T represent exponentiation operations in groups G and G_T, P represents a single bilinear paring operation, respectively. We varied the size of the attribute set from 5 to 50 with a step of 5. The result of each experiment is the average execution time based on 40 trials, ensuring accuracy. It is worth noting that the selection of this data set is based on real-world scenarios, as an excessive number of attributes may render it impractical.

We first compare the time costs of different schemes. In terms of system setup time, we initially compared the time curves as the number of attributes gradually increased for the five schemes. As shown in Fig. 6(a), the system setup time for all five schemes increases linearly as the number of attributes grows from 5 to 50. Among them, due to Li's scheme attempts to maintain a large revocation list and Xiong's scheme incorporates mechanisms of partially hidden policies, both initialized with two exponentiation operations for each attribute. This results in a significantly higher growth rate in their setup time compared to the other schemes. Yu's scheme introduces the concept of domain, with additional computation overhead to ensure domain confidentiality, resulting in higher time costs compared to our scheme and Hahn's scheme. For our scheme, an additional exponentiation on group operation and a pairing operation are performed to protect product identifiers. Thus, although our growth rate aligns with Hahn's, our scheme incurs slightly higher time costs, but overall, it remains within an acceptable range.

Then, we compare the encryption time curves with the increasing number of attributes for the five schemes. As shown in Fig. 6(b), the encryption time for all five schemes increases linearly as the number of attributes grows from 5 to 50. Among them, Xiong et al. [42] exhibits the fastest growth rate and requires the longest encryption time, which is caused by their utilization of a Pederson commitment to achieve the verification of the key. The other four schemes, including ours, have almost same growth rates, and the differences primarily stem from the use of additional exponentiation operations to achieve ciphertext constructions for specific scenarios.

Then, we compare the users' secret key generation curves with the increasing number of attributes for the five schemes. From Fig. 6(c), it can be observed that as the number of attributes increases from 5 to 50, the decryption time for Xiong et al. [42] and Yu et al. [38] grows rapidly, while our scheme exhibits significantly lower overhead, nearly comparable to Li et al. [36] and Hahn et al. [43]. This is because our scheme introduces an additional step for identity version verification, making it slightly more complex compared to Li et al. [36] and Hahn et al. [43]. However, in Xiong et al. [42], additional pairing operations are introduced for decrypting semi-hidden policies, and in Hahn et al. [43], the user's identity is exponentiated to every attribute value to support traceability. These factors contribute to the significant overhead in their schemes.

Next, we compare the decryption time curves with the increasing number of attributes for the five schemes. As depicted in Fig. 6(d), although the decryption time for all four schemes increases linearly with the number of attributes, Hahn et al. [43] demonstrates exceptionally rapid growth. This is because Hahn et al. [43] involves additional pairing operations for revocation in each domain, while the other schemes rely on constant pairing operations. Meanwhile, Yu et al. [38] and Xiong et al. [42] also introduce domain parameters in their schemes, raising the operation complexity of match test period for each access strategy.

In terms of the time costs during the *Revocation* period, we conducted experiments and analysis around the three proposed revocation types, and the time expenditure analysis

TABLE IV
COMPARISON OF TIME COSTS OF REVOCATION

Scheme	Type-1 User Revocation				Type-2 Attribute Revocation	Type-3 Product ID Revocation
	Key Sanity Check	Logs Upload	Logs Audit	User Trace		
Li *et al.*[36]	$4P$	-	-	-	$2nE_1 + P + E_1$	-
Yu *et al.*[38]	$2P$	-	-	-	$3nE_1 + E_T$	-
Ours	$9P$	$6E_1 + 2P$	$2E_1$	$5E_1 + P$	$2nE_1$	E_1

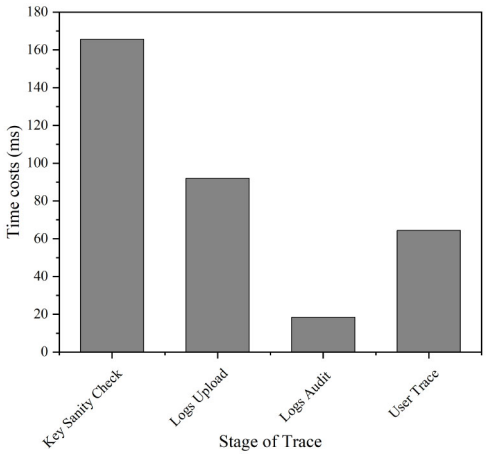

Fig. 7. Time costs of malicious user trace.

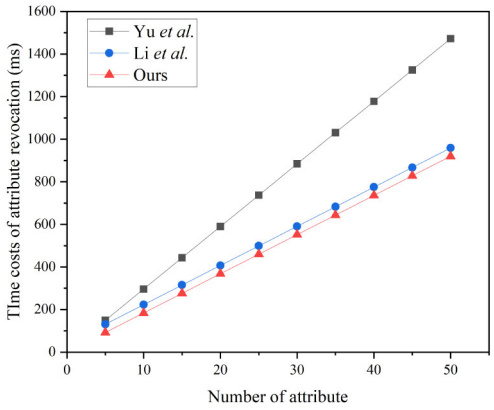

Fig. 8. Time costs of attribute revocation.

is presented in Table IV. In *Type-1-User Revocation*, it can be observed that although all three schemes employ the *Key Sanity Check* algorithm, the costs vary due to differences in the private key structures of the three schemes. However, these costs are fixed. Additionally, since our scheme supports auditing of user behavior, it involves additional steps compared to their schemes. The overall time costs for the malicious user trace are shown in Fig. 7, indicating that despite our scheme requiring additional user public keys and introducing auditing and supervision processes, the overall time cost remains within an acceptable range. This aligns with the PHM system's requirements for data security and real-time performance.

Regarding *Type-2-Attribute Revocation*, as shown in Fig. 8, the time costs for all three schemes linearly increase with the number of attributes. Yu's scheme exhibits a significantly higher growth rate due to the additional attribute revocation for each domain. Li's scheme, utilizing pairing and exponentiation operations to maintain its revocation list, incurs slightly higher time costs compared to ours.

In *Type-3-Product Identity Revocation*, as it only requires changing the identity version number to render the identity unresolvable, a single group exponentiation operation is sufficient, and the cost is negligible.

TABLE V
COMMUNICATION OVERHEAD OF ENTITIES

Entity	Communication Overhead
User	3.1ms
Identity Resolution Platform	193.3ms
Cloud Service Provider	452.7ms
Industrial Enterprise	118.2ms

In Table V, we present the total time overhead of different entities in our model during a complete identification and resolution process using our scheme. The access control policy here is set to be composed of five attributes, and it is assumed that user attributes comply with this policy. It can be observed that, due to the outsourcing of complex bilinear operations to the CSP, the communication cost for users is very low, almost negligible. The time overhead for the IPR platform includes all components and involves multiple steps, making it slightly higher. The time overhead for enterprises is mainly related to communication with the cloud. In the context of PHM maintenance, the communication costs of various entities are entirely acceptable.

Table VI and Fig. 9 illustrate the comparison of storage overhead. As shown in Fig. 9(a), with the increase in the number of attributes, the storage overhead of the public keys for all five schemes exhibits linear growth. However, our scheme demonstrates the lowest growth rate among them. Hahn et al. [43]'s scheme shows the highest growth rate due to the additional exponentiation used for obfuscation purposes. On the other hand, Li et al. [36]'s scheme requires three exponentiations for each attribute to distinguish its validity.

In Fig. 9(b), it is evident that the storage overhead of private key for users exhibits linear growth. Hahn et al. [43], Li et al. [36], and Yu et al. [38]'s schemes share a similar growth rate due to their utilization of complex bilinear operations to embed user information and revocation attributes into the user's private key. In contrast, our scheme leverages blockchain technology, resulting in a lower growth rate for user storage overhead. Additionally, Xiong et al. [42]'s scheme incurs extra computational overhead due to the combination of symmetric encryption. Similarly, our scheme also considers symmetric encryption but utilizes XOR operations to conceal the key.

Furthermore, according to Fig. 9(c), with the increase in the number of attributes, our scheme shares a consistent growth

TABLE VI
COMPARISON OF STORAGE OVERHEAD

Stage	Public key parameters	Secret key parameters	Cipertexts
Xiong *et al.*[42]	$(2n+11)\chi_1$	$(5\vartheta+2)\chi_1$	$(6l+3)\chi_1+1\chi_2$
Hahn *et al.*[43]	$(6n+2)\chi_1$	$(2\vartheta+3)\chi_1+1\chi_3$	$3l\chi_1+11\chi_2$
Li *et al.*[36]	$(3n+4)\chi_1+1\chi_2$	$(2\vartheta+2)\chi_1+1\chi_2$	$(l+5)\chi_1+1\chi_2+\chi_3$
Yu *et al.*[38]	$(2n+2)\chi_1+1\chi_2$	$(2\vartheta+4)\chi_1$	$(2l+6)\chi_1+1\chi_2$
Ours	$(n+6)\chi_1+1\chi_2$	$(\vartheta+4)\chi_1+1\chi_3$	$(2l+1)\chi_1+1\chi_2+1\chi_3$

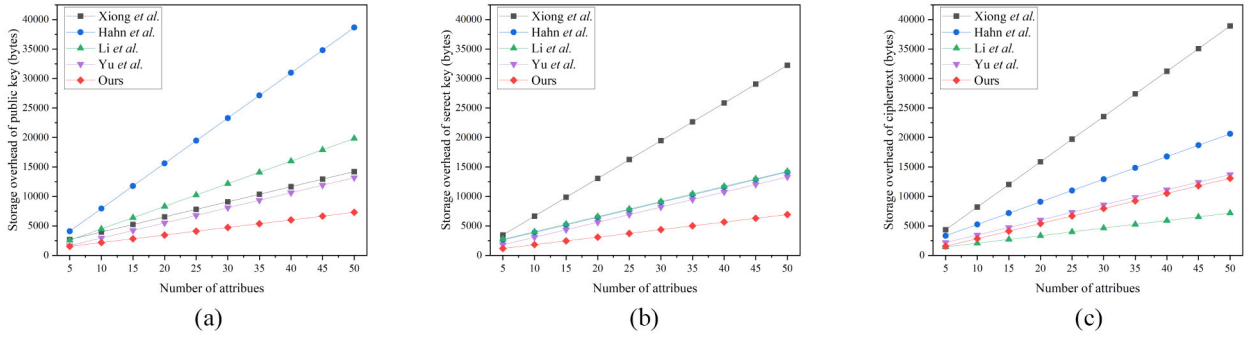

Fig. 9. Storage overhead. (a) Public key. (b) Secret key. (c) Ciphertext.

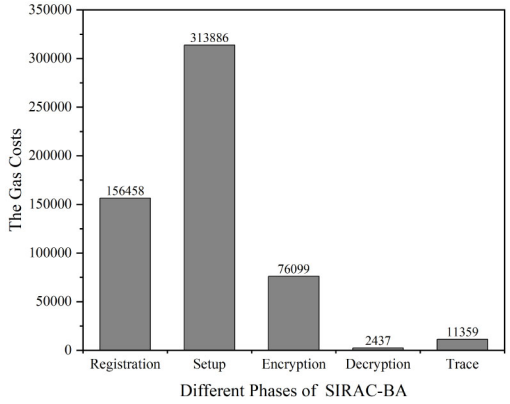

Fig. 10. Gas cost of our scheme.

rate with Yu et al. [38]'s scheme, which outperforms Xiong and Hahn et al. [43]'s schemes. However, it slightly lags behind Li et al. [36]'s scheme. This difference is attributed to Li et al. [36]'s approach of transferring certain ciphertext parameters to the public key during system initialization.

Finally, we use Ethereum platform to evaluated the cost of the blockchain communication in different phases of our proposed scheme, and summarized the gas consumption in Fig. 10. In this experimental evaluation, during the *Registration* phase, we simulated the generation of five *UID* for users and five id_s for subidentifiers using a PRNG. In the *Setup* phase, we uploaded the public key parameters to the blockchain platform. In the *Encryption* phase, we stored only index of the policy-related ciphertext components on the blockchain to enhance storage and computation efficiency for miners. In the *Decryption* phase, these metadata were used in outsourced decryption tests with the transformation key in the corresponding transaction. In the *Trace* operation, IRP-BC needed to store the disclosed decryption key to trace malicious users and record proof of abnormal behavior. Therefore, we assessed the corresponding gas costs on the Ethereum blockchain in these five phases.

In conclusion, based on the results from Figs. 6 and 9, our scheme performs well in terms of storage overhead and time costs, especially when the number of attributes is large, compared to the other referenced schemes in IIoT.

VIII. CONCLUSION

As a critical infrastructure in the convergence application of PHM and the IIoT, enhancing the security of identity resolution systems is of paramount importance. In this article, we presented an ABAC scheme for secure identity resolution in PHM, which achieves fine-grained permission granting and multitype revocation. Our scheme incorporates a novel identity encoding method using CP-ABE for data categorization and permission granting. To combat collusion attacks, user identity information is embedded in their private keys. Leveraging blockchain technology, our scheme enables privacy-preserving log auditing and efficient revocation of malicious users during the trace process. Additionally, outsourced decryption enhances decryption efficiency. Formal security proofs demonstrate that our scheme is secure under the IND-CPA model and possesses traceability and auditability. When compared to other access control schemes in the IIoT domain, our approach demonstrates significant advantages in terms of storage overhead and time costs. By integrating our encoding method and access control scheme, we provide theoretical guidance for the information transformation of industrial enterprises. As a direction for future work, user identity authentication for identity resolution can be a focal point of research.

REFERENCES

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, pp. 239–242, Jun. 2014.
[3] X. Jia, B. Huang, J. Feng, H. Cai, and J. Lee, "A review of PHM data competitions from 2008 to 2017: Methodologies and analytics," in *Proc. Annu. Conf. Progn. Health Manag. Soc.*, 2018, pp. 1–10.
[4] Y. Ren, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Potential identity resolution systems for the Industrial Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 391–430, 1st Quart., 2020.
[5] Y. Ren et al., "Survey of identity resolution system in Industrial Internet of Things," *J. Commun.*, vol. 40, no. 11, pp. 138–155, 2019.
[6] R. Akter, M. Golam, V.-S. Doan, J.-M. Lee, and D.-S. Kim, "IoMT-Net: Blockchain-integrated unauthorized UAV localization using lightweight convolution neural network for Internet of Military Things," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6634–6651, Apr. 2023.
[7] O. Dib and K. Toumi, "Decentralized identity systems: Architecture, challenges, solutions and future directions," *Ann. Emerg. Technol. Comput.*, vol. 4, no. 5, pp. 19–40, 2020.
[8] Z. Qiao, S. Liang, S. Davis, and H. Jiang, "Survey of attribute based encryption," in *Proc. 15th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, 2014, pp. 1–6.
[9] J. Zhang, J. Ma, Y. Yang, X. Liu, and N. N. Xiong, "Revocable and privacy-preserving decentralized data sharing framework for fog-assisted Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10446–10463, Jul. 2022.
[10] X. Xiang, J. Cao, and W. Fan, "Decentralized authentication and access control protocol for blockchain-based e-health systems," *J. Netw. Comput. Appl.*, vol. 207, Nov. 2022, Art. no. 103512.
[11] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[12] A. Al Omar, M. S. Rahman, A. Basu, and S. Kiyomoto, "MediBchain: A blockchain based privacy preserving platform for healthcare data," in *Proc. Int. Conf. Secur., Priv. Anon. Comput., Commun. Storage*, 2017, pp. 534–543.
[13] Z. Qu, Z. Zhang, B. Liu, P. Tiwari, X. Ning, and K. Muhammad, "Quantum detectable Byzantine agreement for distributed data trust management in blockchain," *Inf. Sci.*, vol. 637, Aug. 2023, Art. no. 118909.
[14] E. Hozdić, "Smart factory for industry 4.0: A review," *Int. J. Modern Manuf. Technol.*, vol. 7, no. 1, pp. 28–35, 2015.
[15] R. S. Sandhu and P. Samarati, "Access control: Principle and practice," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 40–48, Sep. 1994.
[16] P. Samarati and S. C. de Vimercati, "Access control: Policies, models, and mechanisms," in *Proc. Int. School Found. Secur. Anal. Design*, 2000, pp. 137–196.
[17] P. B. Prince and S. P. J. Lovesum, "Privacy enforced access control model for secured data handling in cloud-based pervasive health care system," *SN Comput. Sci.*, vol. 1, no. 5, p. 239, 2020.
[18] E. Yuan and J. Tong, "Attributed based access control (ABAC) for web services," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2005, p. 569.
[19] D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control," *ACM Comput. Surv.*, vol. 49, no. 4, pp. 1–45, 2017.
[20] Z. Xu and S. D. Stoller, "Mining attribute-based access control policies," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 533–545, Sep./Oct. 2015.
[21] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy (SP)*, 2007, pp. 321–334.
[22] N. Attrapadung, B. Libert, and E. De Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proc. Int. Workshop Public Key Cryptogr.*, 2011, pp. 90–108.
[23] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptogr.*, 2011, pp. 53–70.
[24] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.
[25] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Ann. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2011, pp. 568–588.
[26] Q. Li, Q. Zhang, H. Huang, W. Zhang, W. Chen, and H. Wang, "Secure, efficient, and weighted access control for cloud-assisted Industrial IoT," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 16917–16927, Sep. 2022.
[27] J. Feng, H. Xiong, J. Chen, Y. Xiang, and K.-H. Yeh, "Scalable and revocable attribute-based data sharing with short revocation list for IIoT," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4815–4829, Mar. 2023.
[28] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti, "Attribute-based encryption with key cloning protection," Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2008/478, 2008. [Online]. Available: https://eprint.iacr.org/2008/478
[29] S. Yu, K. Ren, W. Lou, and J. Li, "Defending against key abuse attacks in KP-ABE enabled broadcast systems," in *Proc. Int. Conf. Secur. Priv. Commun. Syst.*, 2009, pp. 311–329.
[30] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *Proc. Int. Conf. Inf. Secur.*, 2009, pp. 347–362.
[31] J. Li, Q. Huang, X. Chen, S. S. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur.*, 2011, pp. 386–390.
[32] D. Dharminder, P. K. Dadsena, and D. Mishra, "Construction of system friendly attribute based fully distributed access control architecture for e-healthcare," *Multimedia Tools Appl.*, vol. 82, no. 17, pp. 26937–26953, 2023.
[33] J. Katz and D. Schröder, "Tracing insider attacks in the context of predicate encryption schemes," in *Proc. ACITA*, 2011, p. 6.
[34] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Trans. Inf. Forensics Security*, vol. 8, pp. 76–88, 2012.
[35] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1274–1288, 2015.
[36] Q. Li, B. Xia, H. Huang, Y. Zhang, and T. Zhang, "TRAC: Traceable and revocable access control scheme for mHealth in 5G-enabled IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3437–3448, May 2022.
[37] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, and M. Imran, "Lightweight searchable encryption protocol for Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4248–4259, Jun. 2021.
[38] K. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7669–7678, Nov. 2021.
[39] R. Ma, L. Zhang, Q. Wu, Y. Mu, and F. Rezaeibagha, "BE-TRDSS: Blockchain-enabled secure and efficient traceable-revocable data-sharing scheme in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 10821–10830, Nov. 2023.
[40] R. Huo et al., "A blockchain-enabled trusted identifier co-governance architecture for the Industrial Internet of Things," *IEEE Commun. Mag.*, vol. 60, no. 6, pp. 66–72, Jun. 2022.
[41] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2005, pp. 440–456.
[42] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.-H. Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Gener. Comput. Syst.*, vol. 97, pp. 453–461, Aug. 2019.
[43] C. Hahn, H. Kwon, and J. Hur, "Efficient attribute-based secure data sharing with hidden policies and traceability in mobile health networks," *Mobile Inf. Syst.*, vol. 2016, Jul. 2016, Art. no. 6545873.
[44] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2011, pp. 850–855.

Yunhua He (Member, IEEE) received the Ph.D. degree in computer science from Xidian University, Xi'an, China, in 2016.

He was a Visiting Scholar with the Department of Computer Science, George Washington University, Washington, DC, USA, from 2014 to 2016. He has been serving as an Associate Professor with the School of Information Science and Technology, North China University of Technology, Beijing, China. His current research interests include blockchain technology, IoT security and privacy, and industrial internet security.

Zihe Yan received the bachelor's degree from Nanjing Tech University in China, Nanjing, China, in 2021. He is currently pursuing the master's degree in cyberspace security with the North China University of Technology, Beijing, China.

His current research interests include blockchain security and industrial Internet of Things.

Tingli Yuan received the bachelor's degree from Qingdao University, Qingdao, China, in 2021. He is currently pursuing the master's degree in cyberspace security with the North China University of Technology, Beijing, China.

His current research interests include blockchain security and federated learning.