Efficient and Secure Data Storage in 5G Industrial Internet Collaborative Systems

Wang Jigang[1,2], Liu Dong[1,2], Wan Changsheng[3], Lu Ping[1,2]
(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. Southeast University, Nanjing 210096, China)

DOI: 10.12142/ZTECOM.202601007
https://kns.cnki.net/kcms/detail/34.1294.TN.20260306.1414.002.html, published online March 6, 2026
Manuscript received: 2024-06-26

Abstract: Security and access control for data storage in 5G industrial Internet collaborative systems are facing significant challenges. The characteristics of 5G networks, such as low latency and high speed, facilitate data transmission in the industrial Internet but also increase vulnerability to attacks like theft and tampering. Moreover, in 5G industrial Internet collaborative system environments, data flows across multiple entities and links, which necessitates a flexible access control model to meet specific data access requirements. Traditional role-based and attribute-based access control mechanisms are difficult to apply in such dynamic application scenarios. To address these challenges, we propose a novel data storage solution for 5G industrial Internet collaborative systems. Similar to existing approaches, it provides integrity and confidentiality protection for transmitted data. In terms of security, only authenticated data owners and users can obtain file decryption keys, preventing malicious attackers from data forgery. Regarding access control, decryption is permitted only to authorized data users, safeguarding against unauthorized file access. Furthermore, by introducing an attribute-based encryption mechanism, only data users with specific attributes can decrypt files. In terms of efficiency, our approach utilizes bilinear and modular exponentiation operations solely during the authentication process. For handling substantial data loads, lightweight cryptographic algorithms are employed. Consequently, our solution achieves higher efficiency compared with other known methods. Experimental results demonstrate the feasibility of our approach in real-world applications.

Keywords: 5G industrial Internet collaborative systems; data storage; identity-based authentication; access control

Citation (Format 1): Wang J G, Liu D, Wan C S, et al. Efficient and secure data storage in 5G industrial internet collaborative systems [J]. ZTE Communications, 2026, 24(1): 45 - 55. DOI: 10.12142/ZTECOM.202601007
Citation (Format 2): J. G. Wang, D. Liu, C. S. Wan, et al., "Efficient and secure data storage in 5G industrial internet collaborative systems," *ZTE Communications*, vol. 24, no. 1, pp. 45 - 55, Mar. 2026. doi: 10.12142/ZTECOM.202601007.

1 Introduction

In recent years, 5G and beyond technology has been widely adopted and integrated into various fields[1]. As a foundation of the digital economy, 5G industrial Internet collaborative systems play a crucial role. Traditional industrial networks were closed, making it difficult to manage a large number of terminal devices and users and resulting in poor network scalability. With the integration of 5G, industrial networks can efficiently manage numerous industrial terminal devices and users through 5G network elements. Simultaneously, users and devices can easily access the industrial Internet via 5G networks. This transition has opened up industrial networks, allowing users and devices to use the industrial Internet more conveniently, thereby significantly improving production efficiency. Furthermore, by using 5G, industrial networks can seamlessly access external cloud servers and acquire the ability to store large amounts of data at a low cost.

Regardless of the specific technology adopted, a typical data storage scheme for 5G industrial Internet collaborative systems consists of four entities (Fig. 1): the data owner, who stores data in the 5G edge cloud; the data user, who accesses the data; and an authentication server, which provides entity authentication.

Considering the assumptions and requirements for secure data storage in 5G industrial Internet collaborative systems, there is a need to propose a novel data storage scheme based on identity authentication and access control. The design requirements focus on the following five aspects:

1) Industrial data confidentiality[2]. Since the edge cloud is often provided by telecommunications operators, industrial network users may be concerned about data leakage, which could endanger normal production processes, lead to economic losses, and even result in legal disputes. Therefore, the data owner must encrypt the data stored in the cloud.

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. IA20230628015 and the State Key Laboratory of Particle Detection and Electronics under Grant No. SKLPDE-KF-202314.

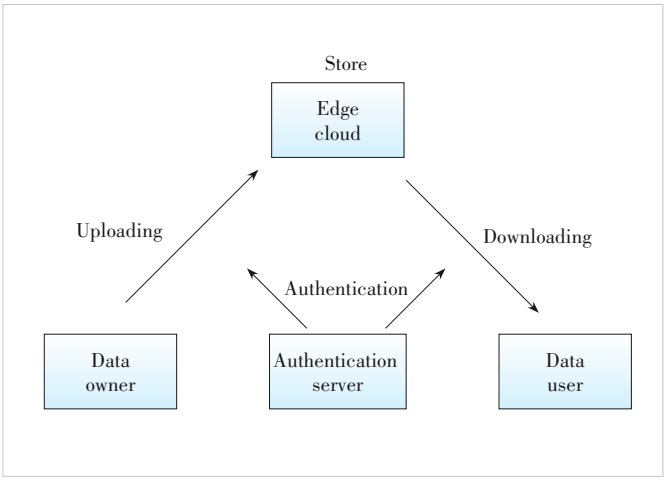

Figure 1. System model and interaction among four entities in a typical data storage scheme for 5G industrial Internet collaborative systems

2) Industrial data integrity[2]. Similar to the confidentiality requirement, the data owner also faces the risk that data stored in the cloud may be tampered with by other users. As a result, the data owner must sign the data stored in the cloud.

3) Authentication of the data owner and the data user[3]. When the data owner authorizes the data user, both parties need to undergo an identity authentication process. Otherwise, malicious attackers may impersonate the data user to illegally access cloud data, or conversely, impersonate the data owner to provide false data. However, since the data stored in the cloud is already encrypted and protected for integrity, this scheme does not require authentication of the edge cloud server.

4) Authorization of the data user and implementation of access control[4]. The data user must obtain authorization from the data owner to access data on the edge cloud. Otherwise, unauthorized malicious attackers may illegally access cloud data, causing economic losses to the enterprise. In addition, only data users with specific attributes should be allowed to access cloud data.

5) Identity-based key management[5]. Since digital certificates require a public key infrastructure and issuing certificates for a large number of terminal devices and users is costly, an identity-based key management mechanism can be used to reduce deployment costs and facilitate the management of a large number of terminal devices and users.

Obviously, designing a secure data storage scheme for 5G industrial Internet collaborative systems is challenging, with diverse and multifaceted design requirements, necessitating careful consideration of each process to be implemented. In this paper, we make the following four contributions:

1) We propose a lightweight and secure data storage system framework for 5G industrial Internet collaborative systems. Compared to other well-known methods, our approach only uses bilinear pairing and modular exponentiation operations during the authentication process, while employing lightweight cryptographic algorithms for handling large amounts of data, resulting in high efficiency.

2) We propose an identity-based authentication scheme, ensuring that only authenticated data owners and data users can obtain the file encryption key, thereby preventing malicious attackers from data forgery.

3) We introduce a lightweight data access control scheme, allowing only authorized data users to decrypt files, preventing illegal attackers from stealing files, while also enabling attribute-based encryption mechanisms for specific data users.

4) We analyze the security of our scheme in the random oracle model, and evaluate the efficiency of newly designed protocols.

The remaining content of this paper is organized as follows. Section 2 reviews relevant literature and research work. Section 3 describes the data storage scheme based on identity authentication and access control for 5G industrial Internet collaborative systems. Subsequently, security analysis is conducted in Section 4, and efficiency evaluation is presented in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Data storage is a vital component within 5G industrial Internet collaborative systems. Ensuring the security of data storage in the industrial Internet primarily encompasses three aspects, namely confidentiality, integrity, and availability[6]. However, there are still some problems and challenges that affect the security of data storage in these systems. Data leakage risk constitutes a significant issue. Owing to the distinctive characteristics of 5G networks, including low latency and high rates, the massive data generated by the industrial Internet is more vulnerable to attacks such as theft and tampering during transmission, which may seriously affect the confidentiality of core enterprise data and cause huge economic losses. Another challenge lies in the complexity of data access control. In 5G industrial Internet collaborative systems, data flows across multiple entities and links, so a flexible access control model is needed to meet its requirements. However, traditional access control models such as role-based access control (RBAC) and attribute-based access control (ABAC) are difficult to apply in this environment. In addition, the distributed storage of data in such systems poses higher requirements for the efficiency and security of access control mechanisms. Presently, numerous research efforts are dedicated to addressing data storage security issues in cloud storage systems. These papers can be mainly divided into three categories: data encryption, access control, and data integrity.

2.1 Data Encryption

Data encryption serves as the foundational technology for ensuring data storage security, effectively thwarting unauthorized access during transmission, processing, and storage. In 5G industrial Internet collaborative systems, data encryption technology needs to adapt to the characteristics of data, such as scale,

flow, bandwidth, and latency, and meet the encryption requirements of different levels. These papers discuss how to achieve efficient and secure data encryption technology in cloud storage systems. Based on revocable-storage identity-based encryption (RS-IBE) technology, a new method for access control of shared data in the cloud is proposed in Refs. [7] and [8]. This method can achieve forward security and backward security, and can resist attacks of private key leakage. Based on the revocation mechanism, the concept of tree structure is introduced in Ref. [9]. To enhance the resistance of decryption keys against leakage, this scheme[9] divides the attribute set into two disjoint sets, and each set is combined with the master key to generate a key. Integrating direct revocation, partially hidden policy, and outsourced decryption attributes of ciphertext-policy attribute-based encryption (CP-ABE) scheme, an innovative data access control method is proposed in Ref. [10]. The security sharing and dynamic access revocation mechanism of electronic healthcare data in public cloud is deeply studied in Ref. [11], guaranteeing forward and backward security. Using CP-ABE, Ref. [12] designs a linear key-sharing scheme to resist chosen plaintext attack (CPA), and demonstrates good performance in dealing with policy change and file update. In Ref. [13], a privacy protection scheme is also constructed based on CP-ABE, employing concealed access policy to facilitate efficient permission verification. However, this scheme[13] only supports the AND policy, so it belongs to weak security model. In 5G industrial Internet collaborative systems, industrial data privacy concerns should receive greater attention.

2.2 Access Control

To safeguard user data security, cloud storage systems need to implement access control techniques to maintain the rights and interests of legitimate users. However, in 5G industrial Internet collaborative systems, traditional access control faces new challenges. For example, a centralized access control server is easy to become a prime target for attackers; once breached, it may lead to data leakage or service interruption. Attackers may steal or tamper with data and resources, or cause other forms of damage. Moreover, cloud security administrators may abuse their privileges to illegally obtain or modify resources and access permissions, thereby reducing user trust in and dependence on the cloud[14]. The following papers discuss how to achieve efficient, secure and flexible access control technology in cloud data storage systems. In the realm of centralized access control, Ref. [15] addresses the challenge of encrypting multiple files with similar access levels in centralized cloud storage by designing an extended file hierarchy CP-ABE scheme (EFH-CP-ABE). While this scheme enhances security and flexibility for cloud storage users, it does have the drawback of extended encryption and decryption computational times. Aiming at the security problems existing in data management operations in centralized cloud storage, Ref. [16] designs an improved model for data access and sharing based on proxy key protocols. This model can effectively resist various attacks, such as user or cloud impersonation, man-in-the-middle attack, and data confidentiality. However, it has the problem of low searching efficiency. For distributed access control, Ref. [17] stores all nodes' public keys and access matrices in the blockchain and proposes an efficient access control method using access matrices as record access policies. This scheme has superiority in terms of computation and storage consumption. In 5G industrial Internet collaborative systems, the access control technique should reduce the dependence on the centralized server.

2.3 Data Integrity

Data integrity verification and data possession are key security issues in cloud data storage systems. Established techniques primarily encompass mechanisms such as data possession proof, recoverability proof, and storage compliance verification. In 5G industrial Internet collaborative systems, a large amount of data will be generated when a large number of industrial terminals access the network, and such data is characterized by complex types and high mobility, which brings new pressure and challenges to data integrity. The following papers discuss how to achieve efficient, secure, and innovative data integrity in cloud data storage systems. Based on proxy re-encryption technology, a scheme that uses data certificates to achieve duplicate data deletion is proposed in Ref. [18]. A data certificate is a signature mechanism based on ownership proof, which adopts encryption algorithms that allow ciphertext decryption via generated keys. Performance analysis of this scheme shows that it can effectively prevent dictionary attacks and improve data security. Based on a cloud file system and verifier, a user-friendly data auditing scheme is constructed in Ref. [19], which does not rely on the collaboration of third parties. This scheme adopts the data reliability verification method proposed in Ref. [20], which can ensure data security and reduce the resource consumption of cloud storage. It also introduces a low-entropy security mechanism to enhance resistance against malicious data attacks.

Unfortunately, existing solutions such as role-based access control cannot provide fine-grained control, while existing attribute-based solutions are costly since they predominantly rely on computationally expensive bilinear map techniques. To address these issues, this paper designs a secure data storage and access control scheme for the low-latency requirement in 5G industrial Internet collaborative systems, and constructs a data access control system framework. The newly designed scheme can provide fine-grained access control while maintaining high efficiency. Moreover, this paper designs a new authentication and authorization scheme based on identity-based encryption techniques, which has better performance than traditional techniques.

3 Proposed Scheme

3.1 System Model

This paper proposes a secure and efficient data storage and access control scheme based on identity authentication and an encryption algorithm for secure cloud data storage in 5G industrial Internet collaborative systems. To this end, the scheme involves the following entities: the data owner, the data user, the edge cloud, and the authentication server.

The system model, as shown in Fig. 2, comprises five phases as detailed below. The corresponding notations are listed in Table 1.

3.1.1 Initialization Phase

During this phase, the authentication server generates public and private cryptographic parameters for the secure cloud storage system. These cryptographic parameters are used to generate keys for the data user and the data owner in the subsequent authentication phase. The initialization algorithm is defined as $\{sk_v, sk_{\text{prod}}, \text{PUB}\} \leftarrow \text{Init}(\text{ID}_v, \text{ID}_{\text{prod}})$.

After the initialization, the authentication server sends the private key sk_{prod} and the corresponding public keying materials PUB to the data owner via a secure channel. Similarly, it sends the private key sk_v and PUB to the data user via a secure channel.

3.1.2 Data Uploading Phase

When the data owner intends to upload a shared file F, it establishes the data uploading process by generating a file encryption key k_F. Then, it encrypts F using k_F and generates a digital

Table 1. Key notations and definitions in this paper

Notation	Description
G, g, p	The cyclic group, its generator, and prime order
$\text{ID}_{\text{prod}}, \text{ID}_v$	Identities of the data owner and the data user, respectively
sk_{prod}, sk_v	Keying materials for ID_{prod} and ID_v, respectively
$H(), h()$	Hash functions
F	The file to be uploaded and downloaded
$\sigma_F, \sigma_M, \sigma_A, \sigma_{k_F}$	The digital signature of F, M, A and k_F, respectively
C_F, C_A, C_{k_F}	The ciphertext of F, A and k_F
$\text{Dec}_k()$	Decryption function
A	The attribute values of the data user
M	Random number generated by the data owner for authentication
r_1, r_2	Random numbers generated by the data owner and the data user for authentication, respectively
k_F	Key generated by the data owner for encrypting the file F
pk_a, sk_a, pk_b, sk_b	Two sets of public and private keys of the authentication server
R_1, T_1	Parameters used for mutual authentication between the data owner and the data user
$Enc_{k_F}(), Enc_k()$	The symmetric encryption function with keys k_F and k
k	The session key
$sk_{\text{req}}, pk_{\text{req}}$	Secret and public information of the authentication request message
pk_{resp}	Public information of the authentication response message
PUB	The set of public keying materials

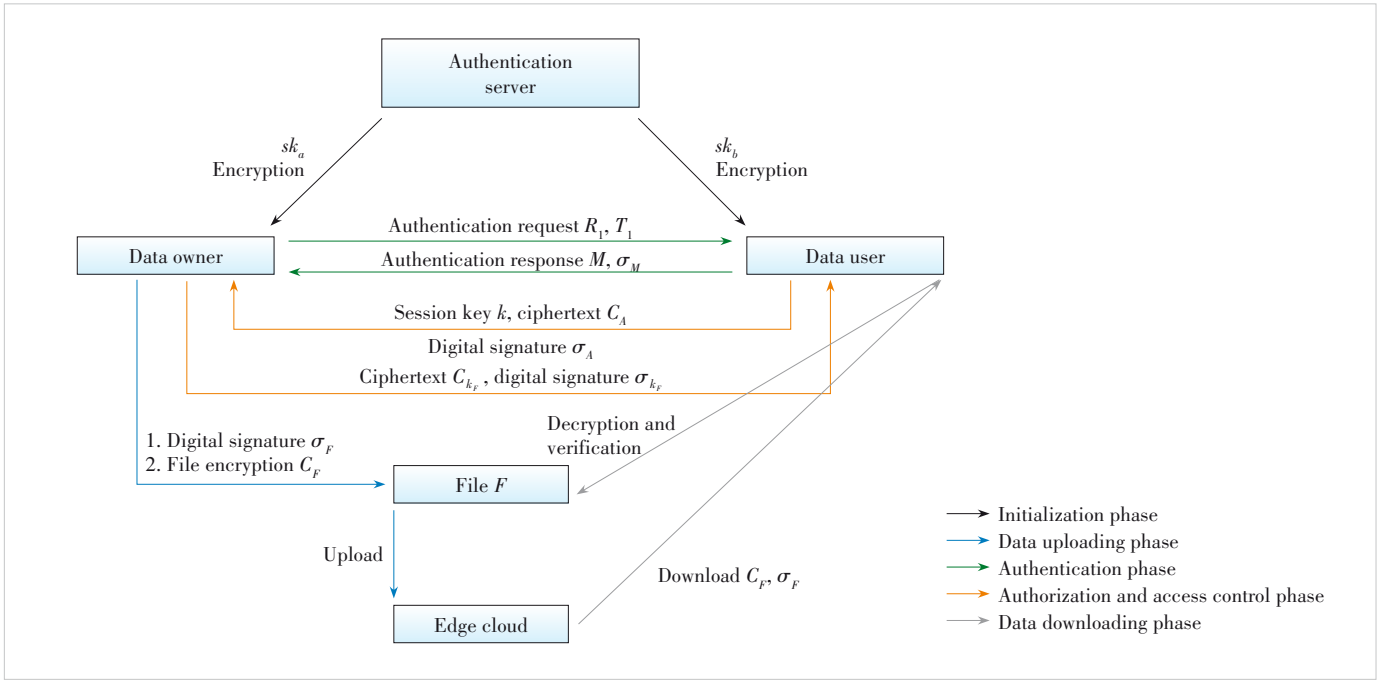

Figure 2. System model and workflow of the proposed scheme

signature σ_F for this file. Finally, the data owner uploads the encrypted file and the digital signature to the edge cloud. The data owner runs the data uploading algorithm to generate the file encryption key, ciphertext, and digital signature as $\{C_F, \sigma_F, k_F\} \leftarrow \mathrm{Dup}(F)$.

After the data uploading, the edge cloud stores the encrypted file C_F and the digital signature σ_F, which will be downloaded by the data user. The file encryption key k_F is held by the data owner, and will be distributed only to the authenticated and authorized data user.

3.1.3 Authentication Phase

Before the data user downloads the shared file F, the data owner initiates mutual authentication by sending an authentication request message pk_{req} to the data user. Upon receiving the authentication request, the data user verifies the request for authenticating the data owner, and then generates and returns an authentication response message (M, σ_M). Finally, the data owner verifies the response message for authenticating the data user. The mutual authentication process comprises the following three steps.

Step 1: The data owner establishes the authentication process by generating a set of secrets sk_{req} and a corresponding authentication request message pk_{req}. Then, the data owner sends pk_{req} to the data user. The authentication request generation algorithm is defined as $\{sk_{\mathrm{req}}, pk_{\mathrm{req}}\} \leftarrow \mathrm{Authreq}(\mathrm{ID}_v, \mathrm{PUB})$.

Step 2: Upon receiving the authentication request message pk_{req} from the data owner, the data user extracts the secret key sk_{req} from pk_{req} and generates the authentication response message pk_{resp} using the set of public keying materials PUB, its secret key sk_v, the data owner's identity $\mathrm{ID}_{\mathrm{prod}}$, and the received authentication request message pk_{req}. It then sends pk_{resp} back to the data owner for authentication. The response generation algorithm is defined as $\{pk_{\mathrm{resp}}\} \leftarrow \mathrm{Authres}(\mathrm{PUB}, sk_v, \mathrm{ID}_{\mathrm{prod}}, pk_{\mathrm{req}})$.

Step 3: Upon receiving the authentication response message pk_{resp} from the data user, the data owner verifies the integrity of pk_{resp} to authenticate the data user. The integrity verification algorithm is defined as $\{\mathrm{True}, \mathrm{False}\} \leftarrow \mathrm{Auth}(\mathrm{PUB}, sk_{\mathrm{req}}, pk_{\mathrm{resp}})$. It takes the set of public keying materials PUB, the set of secrets sk_{req}, and the received authentication response message pk_{resp} as input, and outputs True if the authentication is successful, otherwise False.

Following the authentication, the data owner and the data user mutually verify their identities, and both obtain the set of secrets sk_{req}.

3.1.4 Authorization and Access Control Phase

In this phase, the data user utilizes the newly acquired set of secrets sk_{req}, along with its own attribute values, to request reading permissions from the data owner for extracting the file F. The details of this phase can be described in three steps.

Step 1: The data user employs the set of secrets sk_{req} obtained during the authentication phase to encrypt and sign its attribute values A, generating the ciphertext C_A and digital signature σ_A.

The algorithm for generating the access request message is defined as $\{C_A, \sigma_A\} \leftarrow \mathrm{Accreq}(sk_{\mathrm{req}}, A)$ and run by the data user to generate an access control request. It takes sk_{req} and A as input, and outputs C_A and σ_A.

Step 2: Upon receiving C_A and σ_A, the data owner performs a verification process. If the verification is successful, the data owner will send back the ciphertext of the file encryption key k_F, along with its digital signature σ_{k_F}, to the data user.

The algorithm for generating the access response message is described as $\{C_{k_F}, \sigma_{k_F}\} \leftarrow \mathrm{Accres}(sk_{\mathrm{req}}, k_F)$ and executed by the data owner to generate the ciphertext and digital signature of k_F. It takes the set of secrets sk_{req} and the file encryption key k_F as input, and outputs the ciphertext C_{k_F} and the digital signature σ_{k_F} of k_F.

Step 3: Subsequently, the data user decrypts the received ciphertext C_{k_F} and verifies its signature σ_{k_F}. If the verification is successful, the data user confirms the correctness of the received encryption key and grants access to the file F. The extraction and verification algorithm is defined as $\{\mathrm{False}, k_F\} \leftarrow \mathrm{Acc}(sk_{\mathrm{req}}, C_{k_F}, \sigma_{k_F})$, which outputs the file encryption key k_F for a correct file encryption key, and False otherwise.

Following this phase, the data user gets the file encryption key k_F. Next, in the data downloading phase, the data user will use the newly acquired file encryption key k_F for extracting and verifying the file F.

3.1.5 Data Downloading Phase

In this phase, the data user downloads the ciphertext of the file C_F and its digital signature σ_F from the edge cloud. Then, the data user extracts the file F from C_F and checks the integrity of F using k_F. The extraction and verification algorithm is defined as $\{\mathrm{False}, F\} \leftarrow \mathrm{Ddl}\{k_F, C_F, \sigma_F\}$, which outputs the plaintext of the file F if F is not tampered by an adversary, and False otherwise.

3.2 Construction

The proposed efficient and secure data storage scheme for 5G industrial Internet collaborative systems is defined as a tuple (Init, Dup, Authreq, Authres, Auth, Accreq, Accres, Acc, Ddl) of nine probabilistic polynomial time algorithms. Each algorithm is detailed below.

1) $\{sk_v, sk_{\mathrm{prod}}, \mathrm{PUB}\} \leftarrow \mathrm{Init}(\mathrm{ID}_v, \mathrm{ID}_{\mathrm{prod}})$. The authentication server runs this algorithm to generate system parameters for the secure cloud storage system. This procedure is as follows. First, the authentication server generates a group G with a prime order p and a generator g. Second, the authentication server randomly generates its own private keys, denoted by $sk_a \in Z_p$ and

$sk_b \in Z_p$, respectively. Third, the authentication server generates the corresponding public keys $pk_a = g^{sk_a} \in G$ and $pk_b = g^{sk_b} \in G$. Thus, the public keying materials are defined as PUB={ pk_a, pk_b, G, g, p }. Fourth, Given the data owner's identifier $\mathrm{ID}_{\mathrm{prod}} \in \{0,1\}^n$, the authentication server calculates its secret key as $sk_{\mathrm{prod}} = H(\mathrm{ID}_{\mathrm{prod}})^{sk_a} \in G$, where $H: \{0,1\}^n \rightarrow G$ is a hash function. Finally, for the data user with the identifier $\mathrm{ID}_v \in \{0,1\}^n$, the authentication server computes its secret key as $sk_v = H(\mathrm{ID}_v)^{sk_b} \in G$, using the same hash function H.

2) $\{C_F, \sigma_F, k_F\} \leftarrow \mathrm{Dup}(F)$. The data owner runs this algorithm to generate the file encryption key, the ciphertext and the digital signature. First, for the file F to be uploaded, the data owner randomly generates a number $k_F \in Z_p$, and subsequently computes the digital signature $\sigma_F = h(F, k_F)$, where $h: Z_p \rightarrow Z_p$ is a hash function. Second, the data owner encrypts F as $C_F = \mathrm{Enc}_{k_F}(F)$, where Enc_{k_F} is a symmetric encryption algorithm such as Advanced Encryption Standard (AES), k_F is the file encryption key, and C_F is the ciphertext.

3) $\{sk_{\mathrm{req}}, pk_{\mathrm{req}}\} \leftarrow \mathrm{Authreq}(\mathrm{ID}_v, \mathrm{PUB})$. The data owner runs this algorithm to generate the authentication request message. First, the owner randomly generates $r_1 \in Z_p$ and $k \in Z_p$. Second, the owner computes $R_1 = g^{r_1} \in G$ and calculates $T_1 = h(e(H(\mathrm{ID}_v), pk_b)^{r_1}) \oplus k$, where $H: \{0,1\}^n \rightarrow G$ is a hash function, $e: \{G, G\} \rightarrow G_T$ is a bilinear mapping function, and $h: G \rightarrow Z_p$ is also a hash function. Finally, the owner gets $sk_{\mathrm{req}} = \{k, r_1\}$ and $pk_{\mathrm{req}} = \{R_1, T_1\}$.

4) $\{pk_{\mathrm{resp}}\} \leftarrow \mathrm{Authres}(\mathrm{PUB}, sk_v, pk_{\mathrm{req}})$. The data user runs this algorithm to generate the authentication response. First, the user calculates $k = h(e(sk_v, R_1)) \oplus T_1$, where $e: \{G, G\} \rightarrow G_T$ is a bilinear mapping function and $h: G \rightarrow Z_p$ is a hash function. Then, the user randomly generates $M \in Z_p$, and computes $\sigma_M = h(M, k)$, where $h: Z_p \rightarrow Z_p$ is a hash function. Finally, the user gets $pk_{\mathrm{resp}} = \{M, \sigma_M\}$.

5) $\{\mathrm{True}, \mathrm{False}\} \leftarrow \mathrm{Auth}(\mathrm{PUB}, sk_{\mathrm{req}}, pk_{\mathrm{resp}})$. The data owner runs this algorithm to verify the integrity of the authentication response message by checking whether $\sigma_M = h(M, k)$. If this equation holds, this algorithm outputs True, indicating successful authentication. Otherwise, the data owner outputs False.

6) $\{C_A, \sigma_A\} \leftarrow \mathrm{Accreq}(sk_{\mathrm{req}}, A)$. The data user runs this algorithm to generate an access control request. First, the data user encrypts its attribute value A using the session key $k \in sk_{\mathrm{req}}$ obtained during the authentication phase, producing $C_A = \mathrm{Enc}_k(A)$, where Enc_k is a symmetric encryption algorithm such as AES. Second, the data user generates a digital signature $\sigma_A = h(A, k)$, where $h: Z_p \rightarrow Z_p$ is a hash function.

7) $\{C_{k_F}, \sigma_{k_F}\} \leftarrow \mathrm{Accres}(sk_{\mathrm{req}}, k_F)$. The data owner runs this algorithm to generate an access control response message. First, the owner decrypts C_A using the secret key $k \in sk_{\mathrm{req}}$ to obtain the attribute value $A = \mathrm{Dec}_k(C_A)$, where Dec_k is a symmetric decryption algorithm such as AES. Second, the owner verifies if $\sigma_A \stackrel{?}{=} h(A, k)$ where $h: Z_p \rightarrow Z_p$ is a hash function. If the equation holds, verification succeeds; otherwise, the process aborts. Third, the data owner examines the attribute value A of the data user. If the user satisfies the required access conditions, the owner calculates the ciphertext $C_{k_F} = \mathrm{Enc}_k(k_F)$ and digital signature $\sigma_{k_F} = h(k_F, k)$, where Enc_k is a symmetric encryption algorithm such as AES, and $h: Z_p \rightarrow Z_p$ is a hash function.

8) $\{\mathrm{False}, k_F\} \leftarrow \mathrm{Acc}(sk_{\mathrm{req}}, C_{k_F}, \sigma_{k_F})$. The data user runs this algorithm to extract and verify the file encryption key k_F. First, the data user computes $k_F = \mathrm{Dec}_k(C_{k_F})$, where Dec_k is a symmetric decryption algorithm such as AES. Then, the data user verifies $\sigma_{k_F} \stackrel{?}{=} h(k_F, k)$, where $h: Z_p \rightarrow Z_p$ is a hash function. If this equation holds, verification succeeds, and the data user obtains the file encryption key k_F. Otherwise, the algorithm returns False.

9) $\{\mathrm{False}, F\} \leftarrow \mathrm{Ddl}\{k_F, C_F, \sigma_F\}$. The data user runs this algorithm to retrieve the file F. First, the user decrypts the ciphertext C_F to obtain the plaintext file $F = \mathrm{Dec}_{k_F}(C_F)$, where Dec_{k_F} is a symmetric decryption algorithm such as AES. Then, the user verifies $\sigma_F \stackrel{?}{=} h(F, k_F)$, where $h: Z_p \rightarrow Z_p$ is a hash function. If this equation holds, the file F is not tampered by attackers and this algorithm returns F. Otherwise, it returns False.

In the above construction, the efficient and secure data storage only utilizes bilinear and modular exponentiation operations during the authentication phase, while lightweight cryptographic algorithms are employed for handling large-scale data (i.e., the file F). As a result, these algorithms exhibit high efficiency. We will further evaluate the proposed scheme in Section 5.

4 Security analysis

In this section, we analyze the correctness and security of our proposed solution based on the security requirements outlined in Section 1. These requirements include industrial data integrity and confidentiality, identity authentication, authorization and access control, and identity-based key management. Furthermore, based on these requirements, we extend our analysis to provide proofs for the correctness and security of the secret key $k \in sk_{\mathrm{req}}$, as detailed below.

4.1 Integrity and Confidentiality Requirements

This work employs the symmetric key k_F to encrypt industrial data and generate digital signatures. Therefore, the confidentiality and integrity of industrial data is guaranteed by two factors: 1) the security of the encryption and hashing functions and 2) the security of the key k_F.

Since standard symmetric encryption algorithms and hashing functions are used, their correctness and security are guaranteed by the respective standards. Thus, the primary focus of this paper is on ensuring the security of k_F, as this guarantees the integrity and confidentiality of industrial data. Furthermore, based on the authorization and access control process, k_F is encrypted and digitally signed using the key $k \in sk_{\mathrm{req}}$. Hence, the security of k_F is contingent on the security of $k \in sk_{\mathrm{req}}$, making the protection of $k \in sk_{req}$ the main emphasis of our security design.

4.2 Identity Authentication Requirement

In the authentication phase, the data owner and the data user perform mutual authentication using the secret key $k \in sk_{\text{req}}$ and hash functions. Therefore, similar to the analysis in Section 4.1, the correctness and security of the authentication process rely on the correctness and security of the key $k \in sk_{\text{req}}$.

4.3 Authorization and Access Control Requirement

In the authorization and access control phase, the data owner distributes the file encryption key k_F to the data user for access authorization, and the latter can access the file only upon obtaining k_F. Therefore, the correctness and security of this phase depend on those of k_F.

Furthermore, the authorization and access control phase shows that k_F is encrypted and digitally signed using the secret key $k \in sk_{\text{req}}$. Accordingly, the correctness of authorization and access control relies on the correctness of $k \in sk_{req}$. At the same time, the security of authorization and access control is ensured by the security of $k \in sk_{\text{req}}$.

4.4 Identity-Based Key Management Requirement

The initialization phase indicates that the keys of both the data owner and data user are generated from their respective identities. Therefore, the proposed scheme satisfies the requirement of identity-based key management.

4.5 Secret Key $k \in sk_{\text{req}}$

4.5.1 Correctness

As previously mentioned, we need to prove the correctness of $k \in sk_{\text{req}}$, i.e., to verify that the data owner and data user compute an identical $k \in sk_{\text{req}}$. The proof is presented in four steps as follows.

1) From the Authreq algorithm in Section 3.2, the secret key $k \in sk_{\text{req}}$ is generated by the data owner. Therefore, we only need to ensure that the data user obtains the correct $k \in sk_{\text{req}}$.

2) From the Authres algorithm in Section 3.2, we can see that the data user computes the secret key as $k = h(e(sk_v, R_1)) \oplus T_1$.

3) Based on $sk_v = H(\text{ID}_v)^{sk_b} \in G$ in the Init algorithm in Section 3.2, we derive the key formulation as $k = h(e(H(\text{ID}_v)^{sk_b}, R_1)) \oplus T_1$.

4) Combined with $R_1 = g^{r_1} \in G$ and $T_1 = h(e(H(\text{ID}_v), pk_b)^{r_1}) \oplus k$ in the Authreq algorithm in Section 3.2, we further deduce: $k = h(e(H(\text{ID}_v)^{sk_b}, g^{r_1})) \oplus T_1 = h(e(H(\text{ID}_v), g^{sk_b r_1})) \oplus T_1 = h(e(H(\text{ID}_v), pk_b{}^{r_1})) \oplus T_1 = h(e(H(\text{ID}_v), pk_b)^{r_1}) \oplus T_1 = h(e(H(\text{ID}_v), pk_b)^{r_1}) \oplus h(e(H(\text{ID}_v), pk_b)^{r_1}) \oplus k = k$.

From the above discussion, it is evident that the secret key $k \in sk_{\text{req}}$ obtained by the data owner is the same as that generated by the data user. Therefore, our scheme proposed in this paper is correct.

4.5.2 Security

This subsection primarily demonstrates the security of the secret key $k \in sk_{\text{req}}$. First, we define the Bilinear Computational Diffie-Hellman (BCDH) problem, a well-known mathematical problem hard to solve. Then, we introduce the Indistinguishability under Chosen-Plaintext Attack (IND-CPA) security model to formalize the adversary's attack model. Finally, we prove the security of $k \in sk_{\text{req}}$: If an adversary can obtain $k \in sk_{\text{req}}$ with non-negligible probability, we can solve the BCDH problem with non-negligible probability. Since the BCDH problem is computationally hard, the proposed scheme is provably secure.

Definition 1 (BCDH Problem):

Given $a, b, c \in Z_p$ and $g, g^a, g^b, g^c \in G$, it is hard to compute $e(g,g)^{abc}$ in polynomial time.

Definition 2 (IND-ID-CPA Security Model):

This security model is formally defined by a four-phase game between a Challenger and an Adversary $\mathcal{A}$, as described below:

1) Phase 1: $\mathcal{A}$ adaptively submits an identity $\text{ID}_i \in \{0,1\}^n$, where i=1, 2, ⋯, q_1. The Challenger runs $\{sk_v, sk_{\text{prod}}, \text{PUB}\} \leftarrow \text{Init}(ID_v, ID_{\text{prod}})$ and returns sk_{ID_i} to $\mathcal{A}$.

2) Challenge: $\mathcal{A}$ submits an identity $\text{ID}_v \in \{0,1\}^n$ with $\text{ID}_v \notin \{\text{ID}_1, \text{ID}_2, \cdots, \text{ID}_{q_1}\}$ and two messages k_0 and k_1 with $|k_0| = |k_1|$;

• The Challenger flips an unbiased coin with {0, 1}, and obtains a bit $b \in \{0,1\}$;

• The Challenger runs $\{sk_{\text{req}}, pk_{\text{req}}\} \leftarrow \text{Authreq}(\text{ID}_v, \text{PUB})$ and returns pk_{req} to $\mathcal{A}$.

3) Phase 2: $\mathcal{A}$ adaptively submits an identity $\text{ID}_j \in \{0,1\}^n$ with the limitation $ID_j \neq ID_v$, where j=1, 2, ⋯, q_2. The Challenger runs $\{sk_v, sk_{\text{prod}}, \text{PUB}\} \leftarrow \text{Init}(\text{ID}_v, \text{ID}_{\text{prod}})$ and returns sk_{ID_j} to the Adversary. Let $q_M = q_1 + q_2$.

4) Output: $\mathcal{A}$ outputs its guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Theorem 1: Suppose that H and h are random oracles. The proposed scheme is $(\varepsilon(\lambda), q_1, q_2, q_3, t)$-secure in the IND-CPA security model if the $(\varepsilon'(\lambda), t')$ BCDH assumption holds on the bilinear group (e, p, g, G, G_τ), where $\varepsilon'(\lambda) = \dfrac{\varepsilon(\lambda)}{eq_1q_2}$, and q_1, q_2 and q_3 are the numbers of queries made by the Adversary to H, h and the key generation, respectively.

Proof: Suppose there exists $\mathcal{A}$ that can $(\varepsilon(\lambda), q_1, q_2, q_3, t)$-break the IND-CPA security of the proposed scheme, we construct a Simulator $\mathcal{S}$ that uses $\mathcal{A}$ to break the BCDH assumption. Given $(g, g^{sk_a}, g^{sk_b}, g^{sk_c})$, the Simulator aims to output $e(g,g)^{sk_a sk_b sk_c}$.

1) Setup query: $\mathcal{S}$ sets $pk_a = g^{sk_a}$ and sends the system public parameters $(e, p, g, G, G_\tau, pk_a)$ to $\mathcal{A}$. $\mathcal{S}$ implicitly defines the master secret key is sk_a.

2) Random oracle query: It maintains two hash tables T_H and T_h.

• H-query: $\mathcal{S}$ selects a target index $i_v \in [1, q_1]$. $\mathcal{A}$ adaptively submits an identity $\mathrm{ID}_i \in \{0,1\}^n$, i=1, 2, ⋯, q_1. $\mathcal{S}$ first checks whether $H(\mathrm{ID}_i)$ is in T_H. If so, $\mathcal{S}$ returns $H(\mathrm{ID}_i)$ to $\mathcal{A}$; otherwise, $\mathcal{S}$ works as follows:

$$H(\mathrm{ID}_i) = \begin{cases} g^{z_i}(z_i \in Z_p), & i \neq i_v \\ g^{sk_b}, & i = i_v \end{cases} \tag{1}$$

The Simulator adds $(\mathrm{ID}_i, z_i, H(\mathrm{ID}_i))$ into T_H.

• h-query: $\mathcal{A}$ adaptively submits $W_j \in \{0,1\}^n$, j=1,2,⋯,q_2. $\mathcal{S}$ first checks whether $h(W_j)$ is in T_h. If so, $\mathcal{S}$ returns $h(W_j)$ to $\mathcal{A}$; otherwise, $\mathcal{S}$ randomly selects $w_j \in \{0,1\}^n$ sets $w_j = h(W_j)$, returns w_j to $\mathcal{A}$ and adds (W_j, w_i) to T_h.

3) Phase 1: $\mathcal{S}$ submits an identity $\mathrm{ID}_i \in \{0,1\}^n$. If $i = i_v$, $\mathcal{S}$ aborts; otherwise, $\mathcal{S}$ retrieves $(\mathrm{ID}_i, z_i, H(\mathrm{ID}_i))$ from T_H, computes $M_{\mathrm{ID}_i} = g^{sk_a z_i}$, and returns M_{ID_i} to $\mathcal{A}$. $\mathcal{A}$ can adaptively make this query up to q_{M_1} times.

4) Challenge: $\mathcal{A}$ submits two messages $k_0, k_1 \in \{0,1\}^n$ and identity ID_v. If $\mathrm{ID}_v \neq \mathrm{ID}_{i_v}$, $\mathcal{S}$ aborts; otherwise, $\mathcal{S}$ flips an unbiased coin with {0, 1} and obtains a bit $b \in \{0,1\}$. $\mathcal{S}$ randomly choose $\Gamma \in \{0,1\}^n$ and computes $R_1 = g^{sk_c}$ and $T_1 = \Gamma \oplus k$. $\mathcal{S}$ sends the challenged ciphertext pk_{req} to $\mathcal{A}$. When $h(e(g,g)^{sk_a sk_b sk_c}) = \Gamma$, pk_{req} is a correct ciphertext of the message k; otherwise, pk_{req} is a one-time pad of k_0 and k_1.

5) Phase 2: This phase is identical to Phase 1 with the limitation that $\mathrm{ID}_i \neq \mathrm{ID}_v$. $\mathcal{A}$ can adaptively make this query up to q_{M_2} times. Let $q_3 = q_{M_1} + q_{M_2}$.

6) Guess: $\mathcal{A}$ outputs its guess ω' on ω. If $b' \neq b$, $\mathcal{S}$ aborts; if $b' = b$, $\mathcal{S}$ randomly selects (W_j^*, w_j^*) from T_h and outputs W_j^*.

If pk_{req} is a correct ciphertext, $\Gamma = h\left(e(g,g)^{sk_a sk_b sk_c}\right)$ and $e(g,g)^{sk_a sk_b sk_c}$ are selected by $\mathcal{A}$ to query the h oracle. Hence, $\left(e(g,g)^{sk_a sk_b sk_c}, \Gamma\right)$ must exist in T_h. The simulation is computationally indistinguishable from the real scheme for $\mathcal{A}$.

To complete the proof, we calculate the advantage of $\mathcal{S}$ in solving the BCDH problem by defining the following events:

• E_1: $\mathcal{S}$ does not abort in Phases 1 and 2;
• E_2: $\mathcal{S}$ does not abort in the Challenge phase;
• E_3: $\mathcal{S}$ does not abort in the Guess phase;
• E_4: $\mathcal{S}$ outputs $e(g,g)^{sk_a sk_b sk_c}$.

We have:

$$\begin{aligned} &Pr[E_1] = \left(1 - \frac{1}{q_1}\right)^{q_3}, \quad Pr[E_2] = \frac{1}{q_1}, \\ &Pr[E_3] = \frac{1}{2} + \varepsilon(\lambda), \quad Pr[E_4] = \frac{1}{q_2} \end{aligned} \tag{2}$$

Therefore, the advantage of the simulator in breaking the BCDH assumption is:

$$\begin{aligned} &Pr[E_1] \times Pr[E_2] \times (Pr[E_3] - \frac{1}{2}) \times Pr[E_4] = \\ &(1 - \frac{1}{q_1})^{q_3} \times \frac{1}{q_1} \times (\frac{1}{2} + \varepsilon(\lambda) - \frac{1}{2}) \times \frac{1}{q_2} = \\ &(1 - \frac{1}{q_1})^{q_3} \times \frac{\varepsilon(\lambda)}{q_1 q_2} \approx \frac{\varepsilon(\lambda)}{e q_1 q_2} \end{aligned} \tag{3}$$

5 Efficiency Evaluation

Currently, there are a plethora of attribute-based access control systems (e.g., Refs. [15 – 17]). However, a significant proportion of these systems has not adequately accounted for the distinct requirements introduced by the 5G industrial Internet collaborative systems. Consequently, this section aims to compare an identity authentication and access control-based data storage scheme tailored specifically for the 5G industrial Internet collaborative systems context with the investigations presented in Refs. [21] and [22]. The latter two studies expound upon comprehensive encryption algorithms and protocols for attribute-based access control systems. For the 5G industrial Internet collaborative systems, communicational costs emerge as a paramount concern for access control systems, encompassing computational and communicational costs. To this end, Section 5.1 compares the computational costs of the proposed scheme with Refs. [21] and [22]. Subsequently, Section 5.2 analyzes the disparities and similarities in communicational costs between the proposed scheme and those in Refs. [21] and [22]. During the file uploading and downloading phases, we employ lightweight symmetric cryptography for digital signature and encryption, where the computational and communicational overheads are contingent upon the file size. This aspect is discussed in Section 5.3. Note that Sections 5.1 and 5.2 focus exclusively on the authentication, authorization, and access control phases, while Section 5.3 expounds upon the implementation of the proposed system, thereby substantiating its efficacy.

5.1 Comparison of Computational Costs

Given that computational costs are primarily influenced by cryptographic operations, our focus is directed towards the computational costs of fundamental cryptographic operations (such as modular multiplication, hash functions, bilinear pairings, and modular exponentiation). Subsequently, a comprehensive comparison is undertaken between the proposed data storage scheme and the works in Refs. [21] and [22].

To assess the computational costs of fundamental cryptographic operations, we conducted experiments on a computer with an Intel i5 processor and the Ubuntu 22.04 operating system, using OpenSSL[23] and PBC[24] as cryptographic libraries. The cryptographic group (denoted by G) was a 255-bit elliptic curve group[24], the SHA256 hash functions[24] and AES symmetric encryption algorithm were adopted for all experimental evaluations.

The computational costs of fundamental cryptographic opera-

tions are presented in Table 2, with all results averaged over 500 iterations of the basic cryptographic operations. On the basis of these findings, we draw the following conclusions based on the data in Table 2.

1) The computational costs of modular multiplication and hash function are around 10^{-2} – 10^{-3} to those of modular exponentiation and bilinear pairing, because $T_h/T_p = 1.8/689.3 \approx 2.6 \times 10^{-3}$, $T_h/T_{\text{me}} = 1.8/79.5 \approx 2.3 \times 10^{-2}$ and $T_h/T_{\text{mm}} = 1.8/0.5 = 3.6$.

2) The computational cost of modular exponentiation is around 10^{-1} to that of bilinear pairing, since $T_{\text{me}}/T_p = 79.5/689.3 \approx 1.2 \times 10^{-1}$.

3) The computational costs of AES encryption and decryption on a 128-bit block are around 10^{-2} – 10^{-3} to those of modular exponentiation and bilinear pairing, because $T_{\text{ENC}}/T_p = 3.8/689.3 \approx 5.5 \times 10^{-3}$, $T_{\text{ENC}}/T_{\text{me}} = 3.8/79.5 \approx 4.8 \times 10^{-2}$, and $T_{\text{ENC}}/T_{\text{DEC}} = 3.8/1.4 \approx 2.7$.

The foregoing three conclusions indicate that, in terms of computational costs, modular multiplication, hash function, and AES encryption and decryption have negligible time costs compared to modular exponentiation and bilinear pairing. Therefore, in the subsequent evaluation, our focus centers on modular exponentiation and bilinear pairing. Furthermore, the above finding also highlights that the computational cost of bilinear pairings exceeds those of modular exponentiation. Based on this observation, by avoiding bilinear pairings, the proposed data storage scheme achieves a substantial reduction in computational costs.

Furthermore, building upon the data in Table 2, we deduce the composite computational costs for our data storage scheme, as well as those for the methodologies detailed in Refs. [21] and [22]. Our comparative analysis predominantly centers on a cloud environment, assuming uniform data accessibility and cloud-based access policies. The results of this analysis are presented in Table 3, encompassing the cumulative computational costs during the uploading and downloading phases. From Table 3, we arrive at the following conclusions.

1) The computational costs on the authentication server in Refs. [21] and [22] are approximately 5 to 10 times higher than that of our data storage scheme, because $(1.38\eta_t + 1.70)/0.32 > (1.38 \times 1 + 1.70)/0.32 \approx 9.6$ and $(1.38\eta_t + 0.40)/0.32 > (1.38 \times 1 + 0.40)/0.32 \approx 5.6$.

2) The computational expenses pertaining to the data owner and data user in Refs. [21] and [22] are approximately 2 to 3 times higher than that associated with our data storage scheme, because $(0.08\eta_{nln} + 0.69\eta_v + 2.54)/1.54 > (0.08 \times 1 + 0.69 \times 1 + 2.54)/1.54 \approx 2.1$ and $(0.4\eta_\tau + 2.15\eta_v + 1.33)/1.54 > (0.4 \times 1 + 2.15 \times 1 + 1.33)/1.54 \approx 2.5$.

3) The total computational costs of Refs. [21] and [22] are approximately 3 to 4 times higher than that of our data storage scheme, because $(1.38\eta_t + 0.08\eta_{nln} + 0.69\eta_v + 4.24)/1.86 > (1.38 \times 1 + 0.08 \times 1 + 0.69 \times 1 + 4.24)/1.86 \approx 3.4$ and $(1.38\eta_t + 0.4\eta_\tau + 2.15\eta_v + 1.73)/1.86 > (1.38 \times 1 + 0.4 \times 1 + 2.15 \times 1 + 1.73)/1.86 \approx 3.0$.

The aforementioned three research outcomes indicate that the computational costs in Refs. [21] and [22] surpass those of the secure data storage scheme proposed in this paper. This observation leads to the conclusion that the proposed scheme exhibits superior efficiency.

Table 2. Computational costs of basic cryptographic operations (unit: μs)

T_{mm}	T_h	T_p	T_{me}	T_{ENC}	T_{DEC}
0.5	1.8	689.3	79.5	3.8	1.4

Notes: T_{mm} is the computational cost of modular multiplication, T_h is the computational cost of hash function, T_p is the computational cost of bilinear pairing, T_{me} is the computational cost of modular exponentiation, and T_{ENC} and T_{DEC} are the computational costs encrypting and decrypting a 128-bit block using the AES algorithm.

5.2 Comparison of Communicational Costs

Considering that communicational overhead is primarily influenced by message length, we juxtapose the message lengths of the secure data storage scheme proposed in this paper with those of Refs. [21] and [22] in Table 4. For our proposed scheme, owing to variable uploaded and downloaded file sizes, particular emphasis is placed on comparing the communicational overhead during the identity authentication and access control phases while temporarily disregarding the impact of variable-length ciphertexts on communicational costs.

As discerned from Table 4, the message lengths of our data

Table 3. Comparison of computational costs (unit: ms)

Metric	Proposed Data Storage Scheme	Ref. [21]	Ref. [22]
T_S	$4T_{\text{me}} = 0.32$	$(2\eta_t + 2)T_p + 4T_{\text{me}} = 1.38\eta_t + 1.70$	$2\eta_t T_p + 5T_{\text{me}} = 1.38\eta_t + 0.40$
T_U	$2T_p + 2T_{\text{me}} = 1.54$	$(\eta_{nln} + 6)T_{me} + (\eta_v + 3)T_p = 0.08\eta_{nln} + 0.69\eta_v + 2.54$	$(5\eta_\tau + \eta_v + 8)T_{\text{me}} + (3\eta_v + 1)T_p = 0.4\eta_\tau + 2.15\eta_v + 1.33$
T_a	$2T_p + 6T_{\text{me}} = 1.86$	$(\eta_{nln} + 10)T_{\text{me}} + (2\eta_t + \eta_v + 5)T_p = 1.38\eta_t + 0.08\eta_{nln} + 0.69\eta_v + 4.24$	$(5\eta_\tau + \eta_v + 13)T_{\text{me}} + (3\eta_v + 2\eta_t + 1)T_p = 1.38\eta_t + 0.4\eta_\tau + 2.15\eta_v + 1.73$

Notes: T_S is the computational cost of the authentication server; T_U is the computational cost of the data owner and data user; T_a is the total computational cost on the authentication server, data owner and data user; n is the number of attributes in Refs. [21] and [22]; η_t is the number of entries available in the tag-label-policy list in Refs. [21] and [22]; η_{nln} is the number of total non-leaf nodes in the tree access structure in Ref. [21] and [22]; η_v is the total number of user attributes in Refs. [21] and [22]; η_τ is the total number of attributes in the ciphertext in Refs. [21] and [22].

Table 4. Comparison of message lengths (unit: bit)

Metric	Proposed Data Storage Scheme	Ref. [21]	Ref. [22]
L_A	$4G_1 + 172\,4$	$4G_1 + 2\|\tau\|$	$(3\eta_\tau + 9)G_1 + \|\tau\|$
L_U	$512 + 2C_F$	$(\eta_\tau + 3)G_1 + \|\tau\|$	$(6\eta_\tau + 7)G_1 + 2\|\tau\|$
L_{en}	$4G_1 + 233\,6 + 2C_F$	$(\eta_\tau + 7)G_1 + 3\|\tau\|$	$(9\eta_\tau + 16)G_1 + 3\|\tau\|$

Notes: L_A is the communicational overhead during the authentication and authorization access control phase; L_U represents the communicational overhead during the file uploading and downloading phase; L_{en} is the overall communicational overhead incurred between the phases of authentication and authorization access control and the phases of file uploading and downloading; C_F is the ciphertext length; $|\tau|$ is the size of the access policy; G_1 is the multiplicative group of integers modulo a prime number p; η_τ is the total number of attributes in the ciphertext in Refs. [21] and [22].

storage scheme are shorter compared to the message lengths presented in Refs. [21] and [22], because $(9\eta_\tau + 16)G_1 + 3|\tau| > (\eta_\tau + 7)G_1 + 3|\tau| > (1 + 7)G_1 = 8G_1 = 8\,192\,\text{bit} > 4G_1 + 2236 + 2C_F > 6\,432\,\text{bit}$. In order to ensure a robust level of security, we predicate upon the assumption of G_1 possessing a bit-length of 1 024 bit. The results show that the communicational costs of the data security storage scheme proposed in this paper are lower than those of Refs. [21] and [22].

5.3 Application of Proposed Scheme

We conducted further tests on our solution to assess its computational overhead in the uploading and downloading phases under varying file sizes, as depicted in Fig. 3. We conducted separate tests on binary files of sizes 1 MB, 10 MB, 100 MB, 256 MB, 512 MB, 800 MB, and 1 GB (1 024 MB), respectively. Each computation was derived from the average of ten independent runs, and the results were graphically presented. As illustrated in Fig. 3, the computational overhead for both uploading and downloading phases exhibits a linear increase with the file size. The computational costs on the data owner and data user during the uploading and downloading phases are also presented, with these costs primarily stemming from data encryption and decryption. From the figure, it is evident that the data owner incurs significantly greater computational overhead than the data user. This disparity arises from our utilization of AES in the CBC mode for encryption, a process necessitating sequential block-wise encryption, while the decryption process can be parallelized on multi-core CPUs.

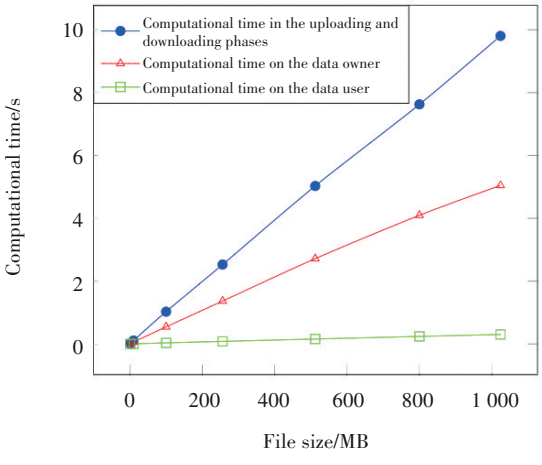

Figure 3. Computational time over different file sizes

To substantiate the efficacy of the proposed data storage scheme, we implemented and tested it in an experimental environment consistent with that described in Section 5.1. Notably, minor adjustments were made to the experimental environment to better align with our research objectives. We also constructed a composite system comprising four distinct computational units, which act as an authentication server, data user, data owner, and edge cloud, respectively. Intercommunication and data interchange amongst these computing entities were facilitated via a 50 Mbit/s Ethernet connection. After a sequence of iterative experiments, we found that the comprehensive execution time of the proposed data storage framework approached 60.71 ms, which is highly consistent with the data in Table 3. Further analysis of the experimental results revealed that the time overhead is mainly consumed by cryptographic operations, thus verifying the practical feasibility of the proposed scheme in real-world scenarios.

6 Conclusions

In this paper, we propose a novel data storage scheme based on identity authentication and access control for 5G industrial Internet collaborative systems. The scheme involves four entities: the data owner, the data user, the edge cloud, and the authentication server. Its distinctive feature is its identity authentication with access control. The data storage scheme includes five phases: system initialization, data uploading, mutual authentication, authorization and access control, and data downloading.

Compared to existing technologies, this approach offers several advantages for 5G industrial Internet collaborative systems. First, it can provide fine-grained access control that traditional schemes such as role-based access control cannot support. Second, it achieves higher efficiency than existing attribute-based access control schemes. Therefore, the proposed scheme provides both high efficiency and fine-grained access control that existing schemes lack. Moreover, this paper presents identity-based authentication and authorization techniques for checking the legitimacy of access requests. Experimental results demonstrate the feasibility of this data storage scheme in practical applications.

In the proposed scheme, the attributes of the data owner and data user are transparent to each other. However, in certain application scenarios, attribute privacy needs to be preserved. Therefore, in future work, we plan to design privacy-preserving

protocols and algorithms that allow the data owner and data user to compare their attributes without disclosing sensitive attribute information.

References

[1] Gao Y, Chen J J, and Li D P. Intelligence driven wireless networks in B5G and 6G era: a survey [J]. ZTE communications, 2024, 22(3): 99 – 105. doi: 10.12142/ZTECOM.202403012.
[2] Wang B Y, Li B C, Li H. Oruta: privacy-preserving public auditing for shared data in the cloud [J]. IEEE transactions on cloud computing, 2014, 2(1): 43 – 56. DOI: 10.1109/TCC.2014.2299807
[3] Shen J, Shen J, Chen X F, et al. An efficient public auditing protocol with novel dynamic structure for cloud data [J]. IEEE transactions on information forensics and security, 2017, 12(10): 2402 – 2415. DOI: 10.1109/TIFS.2017.2705620
[4] Jin H, Jiang H, Zhou K. Dynamic and public auditing with fair arbitration for cloud data [J]. IEEE transactions on cloud computing, 2018, 6(3): 680 – 693. DOI: 10.1109/TCC.2016.2525998
[5] Wang C, Wang Q, Ren K, et al. Privacy-preserving public auditing for data storage security in cloud computing [C]//Proc. IEEE INFOCOM. IEEE, 2010: 1 – 9. DOI: 10.1109/INFCOM.2010.5462173
[6] Yang P, Xiong N X, Ren J L. Data security and privacy protection for cloud storage: a survey [J]. IEEE access, 2020, 8: 131723 – 131740. DOI: 10.1109/ACCESS.2020.3009876
[7] Xu H, Sun B, Ding J W, et al. Analysis of feasible solutions for railway 5G network security assessment [J]. ZTE communications, 2025, 23(3): 59 – 70. doi: 10.12142/ZTECOM.202503007.
[8] Lee K. Comments on "secure data sharing in cloud computing using revocable-storage identity-based encryption" [J]. IEEE transactions on cloud computing, 2020, 8(4): 1299-1300. DOI: 10.1109/TCC.2020.2973623
[9] Xu S M, Yang G M, Mu Y. Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation [J]. Information sciences, 2019, 479: 116 – 134. DOI: 10.1016/j.ins.2018.11.031
[10] Xiong H, Zhao Y N, Peng L, et al. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing [J]. Future generation computer systems, 2019, 97: 453 – 461. DOI: 10.1016/j.future.2019.03.008
[11] Wei J H, Chen X F, Huang X Y, et al. RS-HABE: revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud [J]. IEEE transactions on dependable and secure computing, 2021, 18(5): 2301 – 2315. DOI: 10.1109/TDSC.2019.2947920
[12] Li J Q, Wang S L, Li Y, et al. An efficient attribute-based encryption scheme with policy update and file update in cloud computing [J]. IEEE transactions on industrial informatics, 2019, 15(12): 6500 – 6509. DOI: 10.1109/TII.2019.2931156
[13] Zhang L Y, Cui Y L, Mu Y. Improving security and privacy attribute based data sharing in cloud computing [J]. IEEE systems journal, 2020, 14(1): 387 – 397. DOI: 10.1109/JSYST.2019.2911391
[14] Almutairi S, Alghanmi N, Mostafa M. Survey of centralized and decentralized access control models in cloud computing [J]. International journal of advanced computer science and applications, 2021, 12(2): 1 – 8. DOI: 10.14569/IJACSA.2021.0120243
[15] Li J G, Chen N Y, Zhang Y C. Extended file hierarchy access control scheme with attribute-based encryption in cloud computing [J]. IEEE transactions on emerging topics in computing, 2021, 9(2): 983 – 993. DOI: 10.1109/TETC.2019.2904637
[16] Ghaffar Z, Ahmed S, Mahmood K, et al. An improved authentication scheme for remote data access and sharing over cloud storage in cyber-physical-social-systems [J]. IEEE access, 2020, 8: 47144 – 47160. DOI: 10.1109/ACCESS.2020.2977264
[17] Liu T L, Wu J G, Li J X, et al. Efficient decentralized access control for secure data sharing in cloud computing [J]. Concurrency and computation: practice and experience, 2023, 35(17): e6383. DOI: 10.1002/cpe.6383
[18] Begum B R, Chitra P. SEEDDUP: a three-tier secure data deduplication architecture-based storage and retrieval for cross-domains over cloud [J]. IETE journal of research, 2023, 69(4): 2224 – 2241. DOI: 10.1080/03772063.2021.1886882
[19] Gao X, Yu J, Shen W T, et al. Achieving low-entropy secure cloud data auditing with file and authenticator deduplication [J]. Information sciences, 2021, 546: 177 – 191. DOI: 10.1016/j.ins.2020.08.021
[20] Shen W T, Su Y, Hao R. Lightweight cloud storage auditing with deduplication supporting strong privacy protection [J]. IEEE access, 2020, 8: 44359 – 44372. DOI: 10.1109/ACCESS.2020.2977721
[21] Premkamal P K, Pasupuleti S K, Singh A K, et al. Enhanced attribute based access control with secure deduplication for big data storage in cloud [J]. Peer-to-peer networking and applications, 2021, 14(1): 102 – 120. DOI: 10.1007/s12083-020-00940-3
[22] Cui H, Deng R H, Li Y J, et al. Attribute-based storage supporting secure deduplication of encrypted data in cloud [J]. IEEE transactions on big data, 2019, 5(3): 330 – 342. DOI: 10.1109/TBDATA.2017.2656120
[23] OpenSSL. org. OpenSSL–1.0.1e. tar. gz [EB/OL]. [2023-10-20]. http:// www.openssl.org/source
[24] Lynn B. PBC library manual 0.5.11.2006 [EB/OL]. [2023-10-20]. http://crypto.stanford.edu/pbc/manual

Biographies

Wang Jigang (wang.jigang@zte.com.cn) is General Manager of the Cybersecurity Product Line at ZTE Corporation. His research interests include operating systems, cybersecurity, and cloud computing. Dr. Wang has participated in and supported a number of national key science and technology projects and national science and technology support programs, and has published multiple academic papers.

Liu Dong is Deputy Director of the Central Research Institute at ZTE Corporation. His research interests include operating systems, cybersecurity, and cloud computing. He has participated in and supported a number of national key science and technology projects and national science and technology support programs, and has published multiple academic papers.

Wan Changsheng received his BS degree in applied physics and PhD degree in physical electronics from University of Science and Technology of China in 1999 and 2004, respectively. Since June 2009, he has been with Southeast University, where he is currently a professor with the School of Cyber Science and Engineering. His research interests include network security, wireless communication, and data mining.

Lu Ping is the Vice President and Director of the R&D Project in the Technology Planning Department at ZTE Corporation. He also serves as the Executive Deputy Director of the National Key Laboratory of Mobile Network and Mobile Multimedia Technology. His research directions include cloud computing, big data, augmented reality, and multimedia service-based technologies.