International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
www.computersciencejournals.com/ijcit
IJCIT 2022; 3(2): 33-40
Received: 04-07-2022
Accepted: 20-07-2022

Bhargavi Konda
Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Vinay Kumar Kasula
Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Mounica Yenugula
Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Akhila Reddy Yadulla
Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Santosh Reddy Addula
Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Corresponding Author:
Bhargavi Konda
Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Homomorphic encryption and federated attribute-based multi-factor access control for secure cloud services in integrated space-ground information networks

Bhargavi Konda, Vinay Kumar Kasula, Mounica Yenugula, Akhila Reddy Yadulla and Santosh Reddy Addula

DOI: https://doi.org/10.33545/2707661X.2022.v3.i2a.103

Abstract
Cloud services are a crucial application form within the integrated space-ground information network, enabling users to access information and services quickly and conveniently. The confidentiality and integrity of cloud data are directly linked to the data security of the integrated space-ground information network, leading to the circulation of cloud data primarily in encrypted form. Research on cloud access control technologies must address encrypted data while accommodating multi-factor descriptions in complex environments. Against this backdrop, this paper proposes a Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control (HE-FABAC) scheme by integrating homomorphic encryption technology with federated attribute-based encryption (ABE) and multi-factor access control. First, the design objectives and underlying assumptions are defined. Next, the specific scheme is constructed, detailing the HE-FABAC system model and related algorithms. Finally, the security and characteristics of HE-FABAC are analyzed and compared. By combining homomorphic encryption with federated attribute-based multi-factor access control, HE-FABAC enables multi-factor authorization management for encrypted cloud data. It leverages the computational and storage capabilities of cloud servers, reducing the encryption/decryption workload and key management complexity for individual users.

Keywords: Homomorphic encryption, federated attribute-based encryption, multi-factor, access control, cloud computing, integrated space-ground information network

1. Introduction
The integrated space-ground information network combines space-based backbone networks, space-based access networks, terrestrial node networks, and ground networks like the internet and mobile communications, expanding the scope of human information transmission comprehensively. The construction of the integrated space-ground information network will enhance the data propagation space of existing information systems, transforming them from traditional terrestrial interconnectivity to multidimensional interactions across land, sea, air, and space. Cloud computing, as one of the key applications of current information systems, offers vast storage capacity and powerful computational capabilities. The development of the integrated space-ground information network will broaden and diversify the scope of cloud services, allowing users to engage in data interactions and personalized computing anytime and anywhere, truly enabling the flexible nature of cloud services. However, the heterogeneity, openness, and high degree of integration in the space-ground information network introduce more complex security challenges for cloud data protection. Firstly, user access to the network will become more multifaceted and randomized, increasing management and control complexity. Cloud data management must address increasingly complicated factors. Traditional access control mechanisms based solely on usernames and passwords can no longer meet the demand for feature descriptions of both the client and server in cloud environments. Factors such as roles, time, physical environment, network attributes, security levels of resources, and even multidimensional information from land, sea, air, and space must be considered. Secondly, as the complexity of the network grows, cloud data volume will surge, leading to an increase in confidential data and privacy information. The storage and transmission technologies must be optimized to protect the encrypted form of cloud data.

Cloud computing environments delegate the computation and management of users' data to cloud servers, significantly reducing the resource consumption and time loss for individual users. However, this also introduces the risk of privacy data leaks. As cloud data is typically stored in encrypted form, it is crucial to focus on how to protect and manage this encrypted data.

Access control plays a pivotal role in ensuring the security and reliability of cloud data and services by effectively monitoring user access activities. Given the new security challenges in the cloud environment, access control technologies are evolving to address multi-faceted and encrypted data issues. Specifically, in the context of the integrated space-ground information network, the design of cloud access control schemes should satisfy the following requirements:

1. Ability to describe the complex access control factors and constraints introduced by the integrated space-ground environment, including time, physical environment, network attributes, and other complex factors from different spaces and dimensions.
2. Capability to manage encrypted data. Access control mechanisms must integrate with cryptographic techniques, otherwise, access control policies may become disconnected from data encryption and decryption processes. For example, while users may define time-based access control, the data might be encrypted with identity information as the key.
3. Access control schemes should consider the computational and key management capabilities of individual users, leveraging the computational and storage power of cloud servers to optimize performance.

In response to these needs, numerous models and schemes have been proposed. For example, the RBAC (role-based access control) model, with temporal attributes, has been extended to address time-sensitive access control in cloud computing environments. Other models, such as the CARBAC and SAT-RBAC models, incorporate role attributes and trust relationships into the access control design. Additionally, certain schemes protect cloud data based on location, as seen in works like Li *et al*.'s location-constrained model for cloud data storage. These models address the need for multi-factor access control but fail to effectively manage encrypted data. Traditional cloud access control for encrypted data often requires data owners to encrypt their data before using cloud services, relying on mechanisms like role encryption, IBE, and ABE. However, these models require users to perform encryption, key generation, and decryption operations, which consume significant resources and time, while also creating challenges in key management. At the same time, fine-grained encryption access control often results in an overwhelming number of keys for users to manage, which underutilizes the computational capabilities of cloud servers. Consequently, cloud services primarily function as data storage centers without fully leveraging their computational advantages.

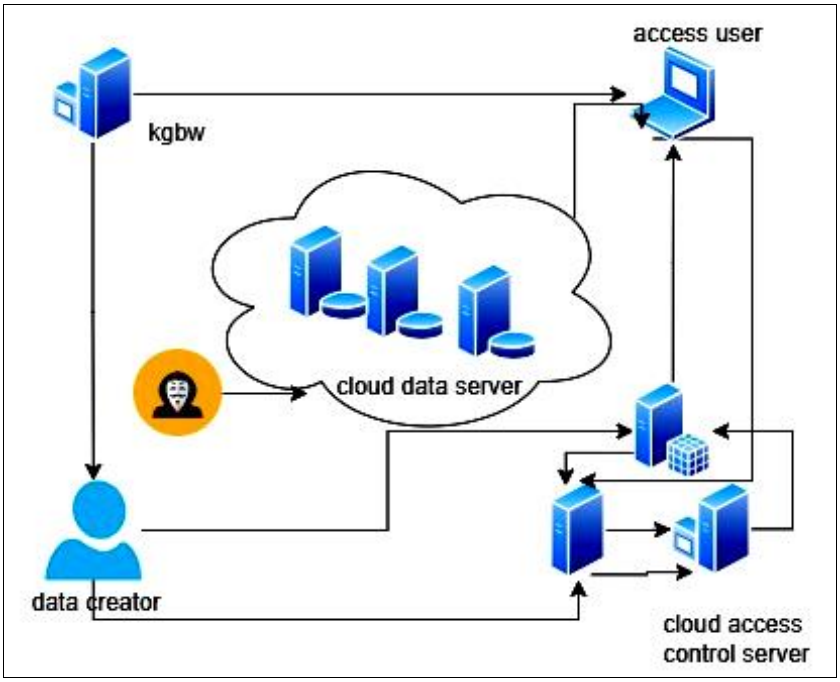

Fig 1: System Model of the Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control Scheme

In the complex, multidimensional environment of the integrated space-ground network, data management for users goes beyond traditional access control factors like roles, time, physical location, and network status. It must also consider factors like network boundaries and cross-domain issues. To address these challenges, this paper proposes a new cloud access control scheme using Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control (HE-FABAC). This approach integrates homomorphic encryption and multi-factor access control to manage encrypted cloud data in a multi-faceted way. By leveraging the computational power of cloud servers, the HE-FABAC scheme reduces the computational burden and key management complexity for individual users. First, the design objectives and assumptions for HE-FABAC are defined, and the system model and relevant

algorithms are described. Finally, the characteristics and advantages of HE-FABAC are analyzed. This approach provides fine-grained, multi-factor data protection for cloud computing environments and is expected to be a key foundation for ensuring data security in integrated space-ground information networks.

2. Relevant Technologies

Homomorphic Encryption (HE) is a cryptographic technique that enables computations to be performed on ciphertexts without decrypting them. This ensures data confidentiality throughout the computation process. In the context of cloud environments, homomorphic encryption allows encrypted data to be manipulated directly by cloud servers, meeting the demands for secure yet efficient data handling in multi-factor access control systems [1-3].

The HE-based mechanism operates as follows

1. The data owner A encrypts their data D using their public key, generating a ciphertext C_A.
2. This encrypted data C_A is then uploaded to the cloud server.
3. When a user B with valid attributes or credentials needed to access D, the cloud server performs computations on C_A to transform it into $C_{A\to B}$, a form accessible to B.
4. B can then decrypt $C_{A\to B}$ using their private key and retrieve the plaintext D.

In this process, *A* only needs to perform the initial encryption, significantly reducing the computational burden on the data owner. The computational tasks are offloaded to the cloud server, which leverages its powerful processing capabilities to manage the encrypted data while maintaining its confidentiality [4-7]. HE-based systems address the challenges posed by traditional access control mechanisms, which rely on users managing multiple keys or performing repeated decryption operations. By supporting direct computation on encrypted data, HE eliminates the need for plaintext exposure during the access control process, ensuring enhanced data security. To further adapt to multi-factor access control, HE mechanisms integrate federated attributes into the encryption process. This involves encoding access conditions, such as roles, time, physical environment, or network attributes, directly into the encryption scheme [8]. By incorporating federated attributes, HE-based systems can dynamically adjust access permissions without requiring separate decryption operations for each attribute. For example, multi-factor HE models allow for the inclusion of attribute-based access conditions, such as location constraints, time-specific permissions, or role-based attributes [9-11]. These attributes are encoded into the ciphertext, and only users satisfying all conditions can decrypt the data. Homomorphic encryption, combined with multi-factor access control, supports fine-grained, attribute-based encrypted data management [12-14]. This approach reduces key management complexity and computational overhead for individual users while maximizing the utilization of cloud server resources. Such schemes are essential in complex environments like the integrated space-ground information network, where multi-dimensional factors must be addressed to ensure secure and efficient data management.

This paper builds upon existing homomorphic encryption models, extending their functionality to handle the diverse and complex requirements of multi-factor access control in cloud environments. By leveraging HE, we propose an advanced cloud data encryption and decryption algorithm that supports fast and effective data authorization management, ensuring security and efficiency in cloud-based systems.

3. Design Objectives and Scheme Assumptions

3.1 Design Objectives

The design objectives of the HE-FABAC scheme are as follows:

1. **Multi-Factor Attribute Description for Encrypted Data Access Control**

HE-FABAC integrates multiple access control attributes into the encryption and key construction processes. These attributes are based on existing multi-factor access control research and are designed to enhance the expressiveness of access control policies for encrypted data.

2. **Granular Encrypted Data Management and Simplified Key Handling**

HE-FABAC enables fine-grained encrypted data management while simplifying key management for users. The decryption process depends on a user's private key and their associated complex access control attributes. Users are only required to maintain a single private key, while the decryption key is dynamically derived from their private key and the access control conditions. This eliminates the need to store separate keys for different encrypted data granularity.

3. **Efficient Utilization of Cloud Server Resources**

In HE-FABAC, the data creator only needs to compute the initial ciphertext without generating multiple ciphertexts for different data-sharing requirements. The cloud server performs the homomorphic operations necessary to enable attribute-based access. This approach reduces the computational burden on individual users, allowing cloud resources to handle encryption transformations and attribute evaluations.

4. **Robust Security Against Attacks**

The HE-FABAC scheme is designed to resist traditional cryptanalysis and brute-force attacks, prevent collusion among unauthorized entities, and safeguard against data privacy breaches by semi-trusted cloud service providers.

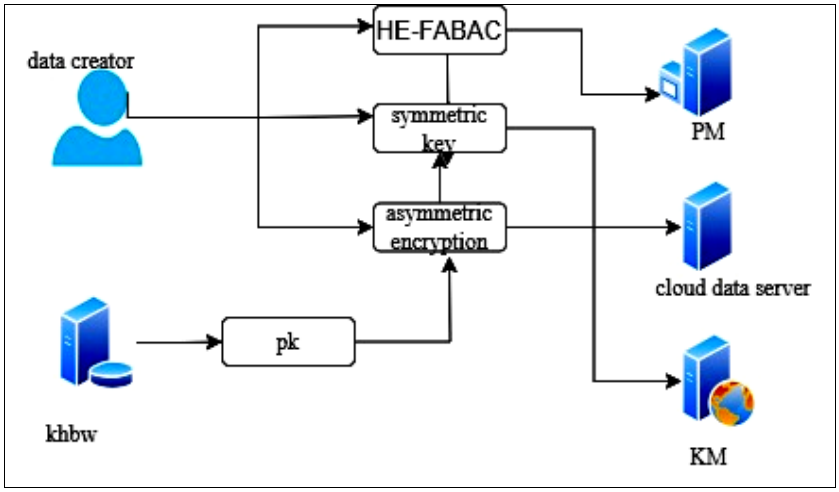

Fig 2: Data Creation Process in the Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control Scheme

3.2 Scheme Assumptions

To implement the HE-FABAC scheme, the following assumptions are made:

1. Network Connectivity

Data creators AAA and users UUU must be connected to the internet to interact with the Key Management Server (KMS), Policy Management Server (PMS), and Cloud Data Server (CDS). These interactions include key acquisition, policy publication, and data access.

2. Trust Model

- The Key Generation Center (KGC), data creators, and users are assumed to be trusted. The KGC is responsible for generating public parameters and private keys. Data creators produce the initial encrypted data, while users are expected to keep their private keys confidential.
- The KMS and PMS are semi-trusted entities. They handle attribute policy management and key distribution but may have limited knowledge about user data.
- Cloud servers are also semi-trusted. They perform homomorphic computations on encrypted data without accessing plaintext.

3. Untrusted Cloud Storage

The Cloud Data Server (CDS) is untrusted, providing only storage services while adhering to open and potentially vulnerable conditions. This requires the encryption scheme to ensure data security even if the storage provider is compromised.

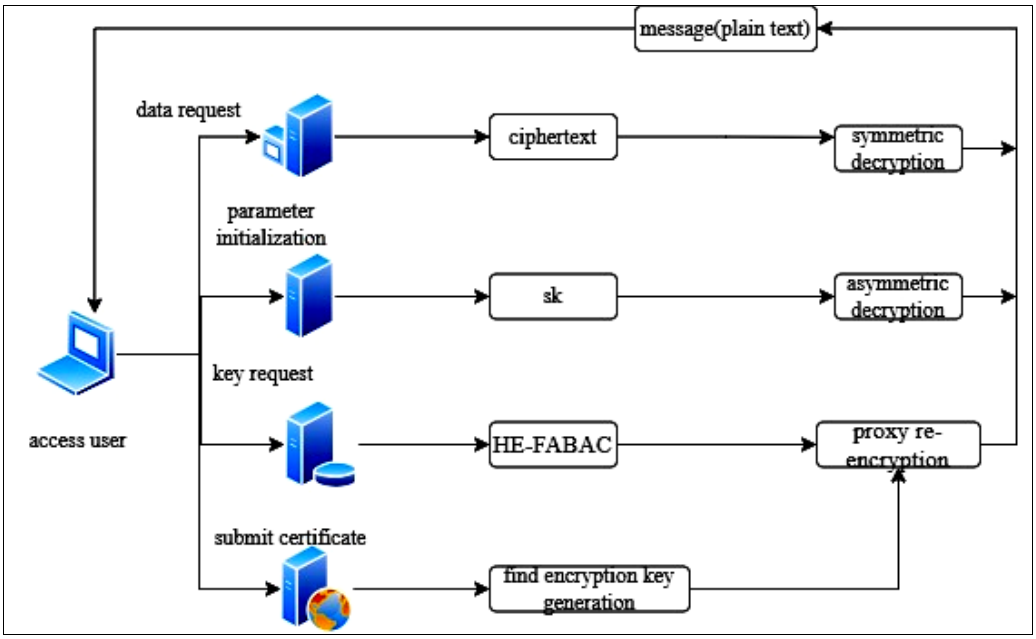

Fig 3: Data Access Process in the Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control Scheme

4. Construction of the Scheme

4.1 System Model

The Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control (HE-FABAC) scheme includes the entities and components described below, as illustrated in the system model (revised from the original PRE-MFAC design):

Entities

1. Data Creator (A)

Creates data *M*, applies security measures (e.g., encryption), and shares it via the cloud server.

2. Access User (U)

Requests access to the data M, retrieves it from the cloud, and decrypts it to utilize the data or service.

3. Cloud Data Server (CDS)

Stores ciphertext produced by symmetric encryption of data M.

4. Cloud Access Control Server (CACS)

Comprises three main components

- **Policy Management Server (PMS):** Manages and enforces access control policies.
- **Key Management Server (KMS):** Manages cryptographic keys and encrypted key storage.
- **Homomorphic Computation Server (HCS):** Performs homomorphic operations to evaluate access control attributes and derive decryption keys without revealing plaintext.

Security Assumptions: The scheme is designed to withstand brute-force attacks and cryptanalysis on both symmetric and homomorphic ciphertexts, while also addressing potential collusion between cloud service providers and attackers targeting access control mechanisms.

Table 1: Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control Scheme Parameters

Name	Description
pk_i	Public key of user i
sk_i	Private key of user i
$K(M)$	Ciphertext of message M encrypted with symmetric key K
$E(K)_i$	Ciphertext that can be decrypted by user i's private key
$Cert_i$	Public key certificate of user i
con_i	Access control conditions of user i
HE_FABAC	Access control policy
$rk_{i\to j}$	Proxy re-encryption key from user i to j (for Federated Access Control)

4.2. System Description of HE-FABAC

HE-FABAC includes two primary workflows: data creation and data access.

1. Data Creation Workflow

This workflow begins with the data creator A, who encrypts the data and uploads it to the cloud. A detailed step-by-step description follows:

Step 1: The data creator A encrypts plaintext data MMM using symmetric encryption, producing ciphertext $K(M)$, which is uploaded to the CDS.

Step 2: A encrypts the symmetric key K using their public key (pk_A), resulting in $E(K)_A$, which is uploaded to the KMS.

- A requests key generation from the Key Generation Center (KGC), providing parameters q.
- The KGC:

a. Initializes the system using $Setup(q)$, generating the security parameter table $param$.

b. Executes $KeyGen(param)$ to generate $A's$ public-private key pair (pk_A, sk_A), which is sent to A.

- A uses $Enc(K, pk_A)$ to encrypt K as $E(K)_A$, then uploads it to the KMS.

Step 3: The data creator A constructs a multi-factor access control policy P_{FABAC} , which is uploaded to the PMS.

2. Data Access Workflow

This workflow begins with access user *U*, who requests encrypted data from the cloud and decrypts it after meeting access control conditions.

Step 1: *U* requests the encrypted data $K(M)$ from the CDS.

Step 2: *U* requests the encrypted symmetric key $E(K)_U$ from the CACS.

1. *U* submits their credentials to the KMS for decryption key generation.
2. The PMS:

a. Evaluates $U's$ attributes and environment parameters against P_{FABAC}.

b. Extracts matching conditions con_U.

3. The KMS:

a. Extracts certificates of A and $(Cert_A, Cert_U)$.

b. Uses these certificates and con_U to compute a decryption key $ReKey(sk_A, pk_U, con_U)$ with homomorphic operations.

c. Sends $E(K)_U$ to the HCS, which performs necessary homomorphic transformations on $E(K)_A$ to derive $E(K)_U$, then returns it to U.

Step 3: *U* decrypts the data

1. *U* uses their private key skUsk_UskU and attributes conUcon_UconU to decrypt $E(K)_U$, recovering K.
2. *U* decrypts $K(M)$ using K, obtaining plaintext M.

4.3 HE-FABAC Scheme Algorithm Description

In the implementation of the HE-FABAC scheme, the following seven main algorithm functions are involved:

1) Parameter Initialization

$Setup(q) \to param$

- Select a prime number ppp of length qqq.

- Define groups G_1, G_2 as multiplicative cyclic groups with g as the generator of G_1.
- Define hash functions H_1, H_2, H_3, H_4 as follows:

a. $H_1: G1 \rightarrow \{0,1\}^*$
b. $H_2: \{0,1\}* \rightarrow Z_p^*$
c. $H_3: \{0,1\}^l \rightarrow G_2$
d. $H_4: G_1 \rightarrow \{0,1\}^*$

- Public parameters: $param = \{p, G_1, G_2, g, H_1, H_2, H_3, H_4\}$.
- Define the bilinear map $e: G_1 \times G_2 \rightarrow G_2$.

2) Key Generation

$$KeyGen(param) \rightarrow (sk_i, pk_i)$$

- Select a private key $x_i \in Z_p^*$.
- Compute the public key $pk_i = g^{x_i}$.
- Output the key pair (sk_i, pk_i).

3) Encryption

$$Enc(K, pk_A) \rightarrow E(K)_A$$

1. The data creator A encrypts symmetric key K using their public key pk_A.
2. Select $r \in G_2$ and compute $H_k = H_2(K \,||\, pk_A)$.
3. Encrypted ciphertext:

$E(K)_A = \{c_1, c_2, c_3, c_4, c_5\}$, where:

- $c_1 = g^r$
- $c_2 = e(pk_A, g)^r \cdot H_1(pk_A)$
- $c_3 = K \oplus H_3(r)$
- $c_4 = H_4(pk_A)$
- $c_5 = H_1(c_1 \,||c_2\, ||c_3||c_4\,)$.

4) Conditional Key Generation

$$ReKeyGen(sk_A, pk_U, con_U) \rightarrow r_{A \rightarrow U}$$

- Compute the re-encryption key based on condition con_U:

$r_{A \rightarrow U} = g^{-r} \cdot e(pk_U, H_1(pk_A \,||\, con_U))$.

5) Homomorphic Re-Encryption

$$ReEnc(E(K)_A, r_{A \rightarrow U} \rightarrow E(K)_U$$

- The homomorphic computation server performs re-encryption on $E(K)_A$, generating $E(K)_U$:$E(K)_U = \{c_1', c_2', c_3', c_4', c_5'\}$ where:

- $c_1' = c_1$
- $c2' = e(pk_U, g)r \cdot H_1(pk_U \,||\, con_U)$
- $c_3' = c_3$
- $c_4' = H_4(pk_U)$
- $c_5' = H_1\,(c_1' \,||c_2' \,||c_3' \,||c_4')$.

6) Decryption by Data Creator

$$Dec(sk_A, E(K)_A) \rightarrow K$$

- If the integrity condition holds, decrypt the symmetric key K as:$K = c_3 \oplus H_3(H_2(K \,||\, pk_A))$.

7) Conditional Decryption by User

$$CondDec(sk_U, E(K)_U, con_U) \rightarrow K$$

- Access user U decrypts the ciphertext $E(K)_U$if the condition $con_U$is satisfied $K = c_3' \oplus H_3(H_2(K \,||pk_U \,||con_U))$.

4.4 HE-FABAC Policy Description and Homomorphic Re-Encryption Parameter Extraction

The access policy P_{FABAC}consists of the tuple $(IDO, P - con_U)$:

- IDO: *Object identifier for the resource.*
- $P - con_U$: Access conditions for the subject, including attributes such as username, role, location, and network properties.

Upon receiving P_{FABAC}, the Policy Management Server (PMS):

- Appends the policy to the management list.
- Analyzes the user's access request, extracts IDO, and retrieves the corresponding $P - con_U$.
- Matches user attributes with $-con_U$.
- If the attributes meet the policy requirements, generates a unified condition description con_U and submits it to the Key Management Server (KMS) for processing.

5. HE-FABAC Scheme

5.1 Security

This paper introduces Homomorphic Encryption and Federated Attribute-Based Multi-Factor Access Control (HE-FABAC) as an enhancement over traditional proxy re-encryption-based methods. The security of HE-FABAC is rooted in its reliance on homomorphic encryption and federated attribute-based access control mechanisms, ensuring robust data protection. The use of homomorphic encryption enables operations on encrypted data without decryption, thus maintaining data confidentiality even during computations. The federated model enhances access control security by distributing attribute-based verification across multiple trusted nodes, minimizing the risk of centralized breaches. Furthermore, HE-FABAC incorporates multi-factor access control policies to evaluate attributes such as user roles, temporal factors, physical location, and environmental parameters. This integration ensures that even in semi-trusted cloud environments, sensitive data remains protected against adversarial attacks or insider threats. The scheme's security aligns with CCA requirements, ensuring resilience against chosen-ciphertext attacks.

5.2 Performance

To evaluate the performance of HE-FABAC, let nnn represent the number of access control attributes, t1t_1t1 the

time complexity of homomorphic encryption operations, and t2t_2t2 the time complexity of federated attribute verification. The time complexity of the main functions in HE-FABAC is summarized in Table 2.

Table 2: Time Complexity of HE-FABAC Functions

Function	Complexity
Encrypt (Enc)	$2t_1 + t_2$
Federated Validation	t_2
Decrypt (Dec)	$t_1 + t_2$

The space complexity primarily pertains to the storage of encryption keys. HE-FABAC enables users to decrypt data using a single private key regardless of the number of access attributes or granularity levels, keeping the key storage complexity at $O(1)$.

5.3 Features

HE-FABAC integrates homomorphic encryption with federated attribute-based multi-factor access control to address the challenges in fine-grained, multi-factor, and ciphertext-based access control. Key features of the scheme are summarized below:

- **Support for Ciphertext-Based Access Control:** HE-FABAC applies attribute-based policies to control access directly over encrypted data, enabling fine-grained control while maintaining data confidentiality.
- **Support for Fine-Grained Management:** The scheme supports fine-grained access control by dynamically generating encryption and decryption keys tailored to specific attribute conditions and user requirements. This flexibility ensures robust data sharing while minimizing overhead.
- **Support for Multi-Factor Access Control Descriptions:** The HE-FABAC model enables the definition of complex access policies, including attributes such as roles, temporal conditions, and environmental factors, combined with logical relationships.
- **Reduced User Key Management Overhead:** Users in HE-FABAC require only a single private key for decryption, irrespective of the number of attributes or granularity levels, significantly reducing the key management burden compared to other schemes that necessitate multiple keys.
- **Distributed Execution of Access Policies:** Unlike centralized systems, HE-FABAC distributes the access policy verification process across federated nodes, improving scalability and reducing the risk of bottlenecks.
- **Objective Evaluation of Access Policies:** HE-FABAC objectively evaluates access policies by verifying conditions in a distributed manner, enhancing reliability and reducing reliance on user input.

Table 3: Comparison of the Proposed Scheme with Existing Research

Attribute Model	Homomorphic Encryption	Federated Attribute-Based Multi-Factor Access Control	Fine-grained	Multi-factor	User Key Management	Encryption/Decryption Execution Side
TAAC	×	×	√	—	—	—
JBE	√	×	×	Small	User	—
CPRE	√	×	√	Large	User	—
Type-PRE	√	√	×	Large	User	—
ACC-PRE	√	×	×	Small	User & Cloud Server	—
PRE-MFAC	√	√	√	Small	User & Cloud Server	User & Cloud Server

Note: Since TAAC does not involve ciphertext access control, it does not involve key management

6. Conclusion

The rapid development of cloud computing and integrated information networks necessitates advanced access control mechanisms capable of safeguarding sensitive data. Traditional access control mechanisms often fail to meet the dual demands of multi-factor attribute evaluation and ciphertext-based access control. HE-FABAC addresses these limitations by combining homomorphic encryption with federated attribute-based access control. By enabling operations on encrypted data and leveraging federated evaluation of multi-factor attributes, HE-FABAC ensures secure and efficient access control in cloud environments. Users can define detailed access policies while maintaining a minimal key management overhead. The federated model further enhances system security by reducing the risks associated with centralized policy enforcement.

HE-FABAC represents a significant step forward in the development of secure, efficient, and scalable access control mechanisms for cloud environments and integrated information networks. It provides a robust foundation for future research in secure data storage, sharing, and computation in distributed systems.

7. References

1. Rizvi A, Javaid N, Yousaf S, *et al*. Server-aided fine-grained access control mechanism with robust revocation in cloud computing. IEEE Access. 2020;8:29782-29792.
2. Sun J, Zhou Y, Wei J, Du X. Attribute-based access control mechanisms for data security in cloud environment. In: Proceedings of the IEEE International Conference on Big Data Security on Cloud; 2021. p. 321-328.
3. Liu T, Zhang X, Ren Y, *et al*. AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud. IEEE Trans Netw Serv Manag. 2021;18(3):1342-1355.
4. Tang Q. Type-based proxy re-encryption and its construction. In: Proceedings of the International Conference on Cryptology in India: Progress in Cryptology; c2008. p. 130-144.
5. Yang K, Liu Z, Jia X, *et al*. Time-domain attribute-based access control for cloud-based video content sharing: A cryptographic approach. IEEE Trans Multimedia. 2016;18(5):940-950.

6. Wang J, Chen X, Li P. REKS: Role-based encrypted keyword search with enhanced access control for outsourced cloud data. IEEE Trans Inf Forensics Secur. 2021;18:2213-2225.
7. Kasula VK, Konda B, *et al*. Hybrid short comparable encryption with sliding window techniques for enhanced efficiency and security. Int J Sci Res Arch. 2022;5(1):151-161.
8. Liu Q, Wu J, Wang G. Secure role-based access control using hierarchical proxy re-encryption for cloud data. IEEE Trans Cloud Comput. 2021;11(2):338-349.
9. Su M, Shi GZ, Xie RN, *et al*. Multi-element proxy re-encryption scheme for mobile cloud computing. J Commun. 2015;36(11):73-79.
10. Li J, Zhao X, Zhang Y, *et al*. Provably secure certificate-based conditional proxy re-encryption. J Inf Sci Eng. 2016;32(4):813-830.
11. Liu Q, Wang G, Wu J. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. Inf Sci. 2014;258:355-370.
12. Yang Y, Lu H, Weng J, *et al*. Fine-grained conditional proxy re-encryption and application. In: Proceedings of the International Conference on Provable Security; c2014. p. 206-222.
13. Azmeera R, Tumma C, *et al*. Enhancing blockchain communication with named data networking: A novel node model and information transmission mechanism. J Recent Trends Comput Sci Eng. 2022;10(1):35-53.
14. Su M, Li F, Shi G, *et al*. A user-centric data secure creation scheme in cloud computing. Chin J Electron. 2016;25(4):753-760.