Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

Identity and Access Control Techniques for Enhanced Data Communication in Cloud

Rashmi Dixit*, **K. Ravindranath**

Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
E-mail: rashmirajivk@gmail.com

Received: 28 September 2023; **Revised:** 6 November 2023; **Accepted:** 10 November 2023

Abstract: The preservation of sensitive data and prevention of unauthorized access are critical objectives in cloud computing environments, necessitating the implementation of robust security measures. The present study delves into the amalgamation of the hierarchical role-based access control model (HR-ACM), Composikey, a composite key encryption algorithm, and interactive tree based zero-knowledge proof (ITZKP) protocol to construct a robust security framework for cloud security. This study presents an investigation into the viability and efficacy of integrating HR-ACM, Composikey, and ITZKP as a means of augmenting security measures and reducing potential hazards in cloud-based systems. The execution times comparison of the Composikey, identity-based proxy re-encryption (IBPRE), and k-times proxy re-encryption (KPRE) models were compared for different file sizes (1 KB to 2 GB) to evaluate the performance of the HR-ACM method. As roles increase, the role-based access control (RBAC) execution time increases significantly. The HR-ACM method takes 0.001 to 0.002 seconds to implement across all role counts. The results show that the HR-ACM method is more efficient and time-effective than the RBAC method. The response time measured in microseconds for the existing zero-knowledge proof (ZKP) and Libra systems for different file sizes, ranging from 1 KB to 1 GB. The results reveal that the verifier time for the ITZKP system aligns closely with the existing systems, demonstrating its ability to perform efficient verification processes. These findings collectively demonstrate the potential of the integrated security framework in enhancing cloud security.

***Keywords*:** cloud computing, HR-ACM, Composikey encryption algorithm, ITZKP, data security, access control

MSC: 68M10, 68M25

1. Introduction

Cloud computing has changed how individuals and organizations manage, retrieve, and manipulate data. Cloud-based systems have made data security a major concern. Uploading data to the cloud, places it outside an organization, requiring data security to consider confidentiality, integrity, and authenticity. Cloud computing security risks necessitate proper identity and access control. In a cloud-based setting, the techniques ensure that only authorized entities can retrieve data while preserving its confidentiality, integrity, and authenticity [1]. The role-based access control (RBAC) authentication model manages authorization. The system assigns roles based on job duties. User-side access control computation restricts data access to authorized parties. The sophisticated cryptographic algorithm, Composikey, provides high data confidentiality assurance. Proxy-based re-encryption protects data during transmission from the

DOI: https://doi.org/10.37256/cm.5120243721

cloud to the client. The cryptographic interactive tree based zero-knowledge proof (ITZKP) protocol secures auditing while protecting inputs and intermediate outcomes. It maintains data authenticity and confidentiality. Identity and access control are essential for cloud computing data security, confidentiality, integrity, and availability [2]. The present literature review aims to investigate the efficacy of diverse techniques utilized for augmenting identity and access control in cloud computing. A literature study focuses on the examination of various access control models, namely the hierarchical role-based access control model (HR-ACM), Composikey algorithm, ITZKP, identity and access management (IAM) framework, and attribute-based access control (ABAC) model [3].

The HR-ACM is a widely adopted method for cloud computing access control. It enables system administrators to designate user's roles based on their job duties and hierarchical positions. HR-ACM is effective at controlling access to cloud resources and preventing unauthorized access, according to studies. The study by Abo-alian et al. [4] proposes a hierarchical attribute-driven role-based cloud computing access control system. The authors emphasize the significance of access control and cryptography in cloud storage to assure data confidentiality. The authors argue that in large-scale cloud systems, manual assignment of roles and permissions can result in computational and online burdens for data proprietors. To address these issues, they propose a system that automatically assigns user roles based on policies applied to user and role attributes. a hierarchical attribute driven RBAC system is proposed, such that the user role assignments can be automatically constructed using policies applied on the attributes of users and roles. Wan et al. [5] propose hierarchical attribute-set-based encryption (HASBE) by extending ciphertext-policy attribute-set-based encryption (ASBE) with a hierarchical structure of users. Bhartiya et al. [6] posed an access control framework that addresses security concerns for the sharing of electronic health record (EHR) data. The proposed framework by the authors uses a hierarchy similarity analyzer (HSA) to identify conflicts and rank user/resource attribute similarity based on similarity factors. This results in justifiable subsets of matching rules and policies for sharing data between independent healthcare entities, while ensuring that only authorized users have access to controlled data.

To address cloud storage security issues, Zhou et al. [7] proposed a role-based encryption (RBE) scheme that incorporates cryptographic techniques with the RBAC model. The proposed RBE scheme enforces RBAC policies for encrypted data stored in public clouds and introduces a secure hybrid cloud storage architecture based on RBE. The Composikey algorithm is a composite key encryption technique that guarantees the security of critical information while maintaining access for authorized users. Waters [8] introduced a novel method for ciphertext-policy attribute-based encryption (CP-ABE) that accomplishes non-interactive and concrete cryptographic assumptions in the standard model. In proxy re-encryption (PRE) protocols, Ateniese et al. [9] introduced the concept of key-private re-encryption keys. They argued that previous schemes lacked this characteristic and defined a new security concept for PRE schemes that included the key-private characteristic. The authors presented the first implementation of a key-private PRE and demonstrated its security under the chosen-plaintext attack (CPA)-security extension of the Decisional Bilinear Diffie Hellman assumption and its key-privacy under the Decision Linear assumption in the standard model. Goyal et al. [10] suggests delegation of private keys which subsumes hierarchical identity-based encryption (HIBE).

The ITZKP protocol is a novel technique for secure auditing check that allows for calculations to be carried out without disclosing any of the inputs or intermediate results. Studies have shown that ITZKP is effective in enhancing cloud security. Zhang et al. [11] proposed a definition of data privacy with zero-knowledge proof for integrity tests in cloud storage. They emphasized that the cloud server must not reveal any valuable information about the stored data and proposed an algorithm covering data confidentiality, integrity, privacy, and soundness. This research is essential for enhancing data communication in the cloud by enhancing data integrity check. Yu et al. [12] proposed a secure zero-knowledge-based client-side deduplication scheme for cloud storage in smart cities over encrypted files. The scheme accomplishes a high probability of detecting client misbehavior and protects the confidentiality of data. A key distribution scheme based on PRE is also introduced to facilitate the delegation of decryption rights. Yang et al. [13] proposed a secure zero-knowledge-based client-side deduplication scheme for cloud storage in smart cities over encrypted files. The scheme accomplishes a high probability of detecting client misbehavior and protects the confidentiality of data. A key distribution scheme based on PRE is also introduced to facilitate delegation of decryption rights. Zero-knowledge proofs (ZKPs) are widely used in cryptography, cloud computing, and finance. Bick et al. [14] proposed distributed ZKPs in their recent study. The protocol is designed to safeguard the knowledge of the prover, including the graph. The authors design distributed ZKPs for the 3-coloring problem and the spanning-tree verification problem that are communication-efficient.

From the literature review, the combination and integration of the HR-ACM, Composikey (a composite key encryption algorithm), and ITZKP within the context of developing robust and secure architectures for cloud computing environments have not been extensively studied by researchers [15, 16]. While each of these systems has been individually explored, their combined utilization to enhance security measures in cloud environments remains relatively unexplored.

This research paper explored the integration of the HR-ACM, Composikey, and ITZKP to develop an efficient security architecture for cloud security. The objective was to combine these systems and leverage their unique capabilities to enhance security measures in cloud computing environments. Through the adoption of a cryptographically secure algorithm for generating random component keys, the Composikey method achieved high entropy, surpassing the simplicity of a random function. The unique ITZKP was successfully employed for secure auditing checks through computation. The results of this research demonstrate the feasibility and effectiveness of combining HR-ACM, Composikey, and ITZKP to develop an efficient security architecture for cloud security. The utilization of these systems offers enhanced access control, strengthened data protection, and secure authentication, mitigating the risks associated with unauthorized access, data breaches, and identity theft in cloud environments.

2. Methodology

The proposed comprehensive security framework for cloud data protection is implemented and evaluated using the research methodology as a framework. In order to fully understand the state-of-the-art in cloud data security, attribute-based encryption (ABE), ITZKP, and related access control models, are conducted as part of the methodology. Figure 1 shows the flow chart of the research methodology.

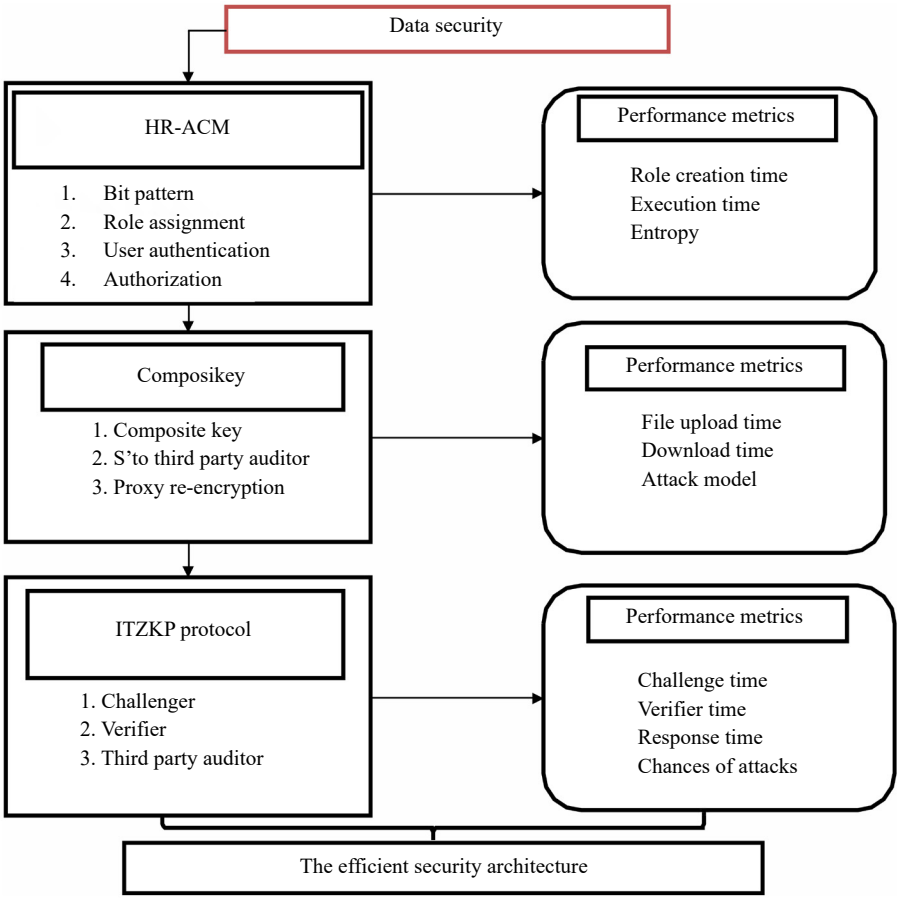


Figure 1. Research methodology

2.1 *Design and implementation phase*

In the design phase, we outline the blueprint for the security architecture by integrating HR-ACM, Composikey, and ITZKP. This involves defining the hierarchical access control structure using HR-ACM, establishing encryption policies with Composikey, and designing the authentication framework utilizing ITZKP.

2.1.1 *RBAC model (HR-ACM)*

The first step is to parse the OWL ontology, which involves reading and extracting the relevant information from the ontology file [17]. This can be done using an OWL parsing library or framework. The design phase of this research focuses on systematically developing and formalizing the methodology for integrating the OWL ontology, generating a bit pattern, assigning roles to users, authenticating users, and ensuring authentication of user actions. This phase aims to establish a comprehensive and robust design for the access control system within the cloud computing environment. Through parsing the ontology, relevant information is extracted and mapped to the bit pattern based on a defined encoding scheme. This encoding scheme assigns specific bit positions or sequences to represent classes, properties, or relationships associated with the access control system. Roles are mapped to users based on their specific privileges and responsibilities. This mapping ensures that users are granted appropriate access rights and permissions within the cloud computing environment, facilitating effective control and management of resources. Figure 2 shows the system architecture for RBAC model.

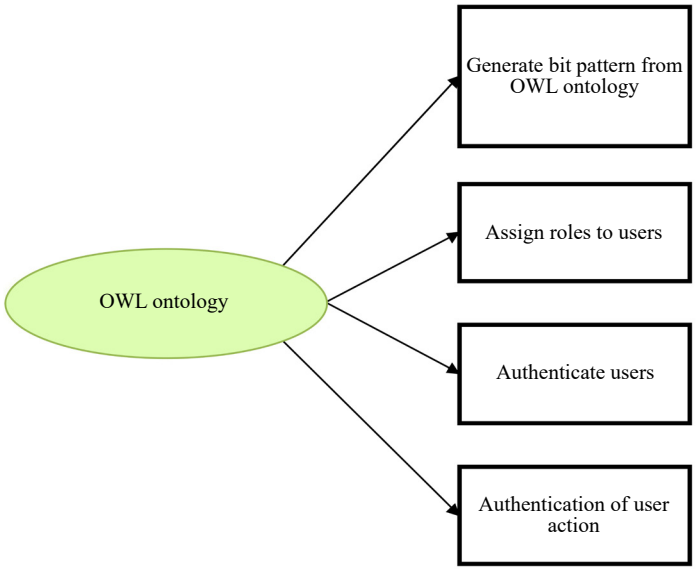


Figure 2. Design phase for RBAC Model

When uploading data to the cloud, the data owner creates a self-protected package with encrypted data, authorization rules, and re-encryption keys (Figure 3). Before uploading data to the cloud, the owner encrypts it with a random symmetric key. Encrypting the encryption key improves data security [18]. The symmetric method, advanced encryption standard (AES), recognized for its robustness, is utilized as it provides a suitable level of protection. Access control relies on authorization. This model controls data access permissions. These rules create re-encryption keys. The data owner, with a generator and element identities, is responsible for this process. An identity-based scheme is employed to generate the keys for re-encryption, effectively avoiding the inclusion of extraneous attributes and minimizing storage requirements. After generating the re-encryption keys, the final user can manipulate the data. It's important to note that the user can't do anything wrong to obtain the data. This data-centric method uses data to carry all necessary components within the same service provider or intercloud scenario. The decision point and PRE must be deployed for smooth cryptographic processes. The decision point controls the authorization model and starts the access procedure. It verifies a user's data

access rights. Permission is granted or denied after rule review. Permission grants re-encryption keys. This information is subsequently passed to a trustworthy third-party entity, known as the proxy, which undertakes the process of PRE. Subsequently, when the ultimate user intends to utilize the data, they perform decryption using their set of keys.
Importantly, the service provider cannot misbehave or access the data. The genuine user alone knows the re-encryption information, assuring data integrity and security.

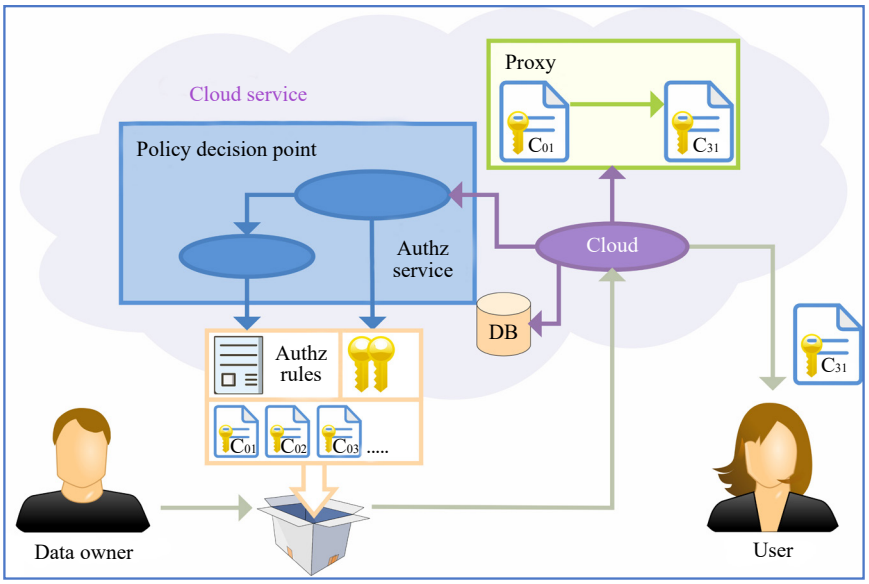


Figure 3. System architecture for RBAC model [18]

2.1.2 *ITZKP protocol*

The implementation of the ITZKP for cloud data storage systems. The protocol involves three parties: the data owner, who acts as the prover, the cloud storage server, and a verifier, who is a semi-trusted third party [19]. The goal is to ensure the integrity of the outsourced data without the need for periodic data integrity verification. In the ITZKP, the data owner relies on the cloud storage server to store and manage their data (Figure 4). As the data is no longer stored locally, the data owner utilizes ITZKP to check the integrity of the outsourced data. This protocol allows the data owner to verify that the cloud server is correctly storing and maintaining the data [20].

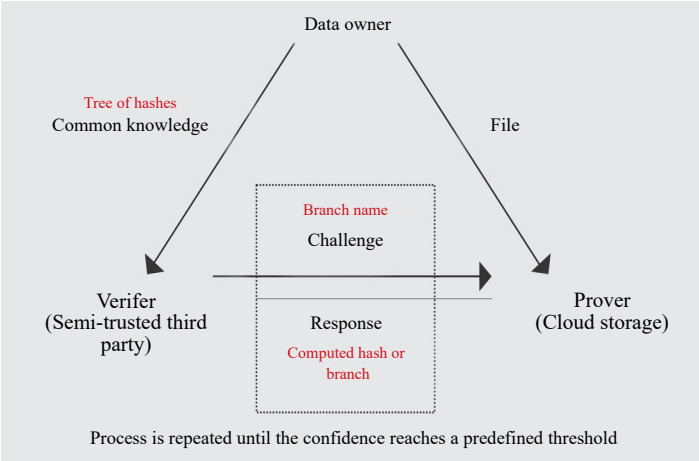
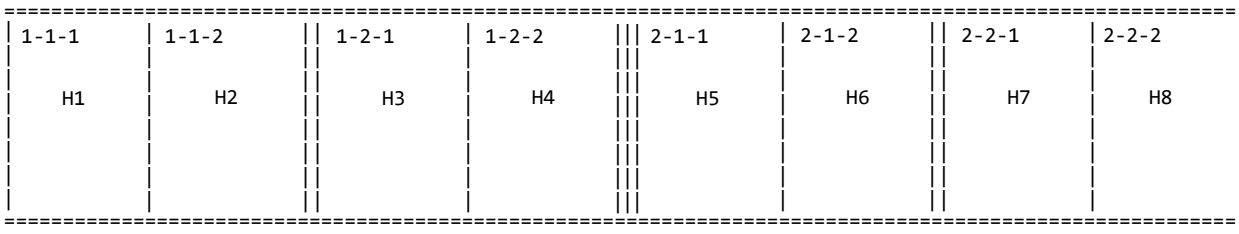


Figure 4. Design phase for ITZKP protocol

Figure 5 shows the hash calculation and hash tree used for implementation. The process begins with the data owner uploading the file to the storage cloud. The file is divided into smaller chunks, with a minimum block size of 4 kilobytes to optimize performance. In this scenario, the file is divided into eight chunks. To establish common knowledge between the verifier and the data owner, a complete tree of hashes is created for the file, represented as H15 + H13 + H14. This tree of hashes serves as the basis for verification.



(a)

Chunk	Hash
1	H13 + H9 + H10
2	H14 + H11 + H12
1-1	H9 + H1 + H2
1-2	H10 + H3 + H4
2-1	H11 + H5 + H6
2-2	H12 + H7 + H8
1-1-1	H1
1-1-2	H2
1-2-1	H3
1-2-2	H4
2-1-1	H5
2-1-2	H6
2-2-1	H7
2-2-2	H8

Complete file has Hash H15 + H13 + H14

(b)

Figure 5. (a) Hash tree; (b) Hash calculations

The verifier generates a challenge by selecting a random nonce (R) and a tree branch. The prover, in this case, the data owner, computes the hash of the selected tree branch (S). The prover then computes (R·S%N). The verifier performs further computations by evaluating (R·S%N)^2)%N and ((R^2)%N·(S^2)%N)%N. If there is a match, the verifier updates its confidence value.

This process is repeated until the confidence value reaches the desired set value. The ITZKP algorithm ensures that the data owner can verify the integrity of their outsourced data in a secure and efficient manner, eliminating the need for periodic data integrity verification [21]. Within the ITZKP algorithm, certain technical details are relevant. These include the usage of a large prime number (base) denoted as 'g' and another large prime number (modulus) denoted as 'N'. Additionally, the public key of the trusted third-party auditor (TPA) is referred to as '$KPUB_{TPA}$'.

Figure 6 illustrates the step-by-step execution of the ITZKP algorithm, which ensures data integrity in cloud storage systems. The algorithm involves the data owner, third party auditor, and prover (storage service) [22].

Data owner: The data owner initiates the protocol by uploading a file, denoted as 'data' to the storage service. The data is then subjected to a hash function, H(data), resulting in the hash value D. Next, the data owner computes a secret, S, using the formula g^D mod N, where 'g' represents a large prime number (base) and 'N' represents another large prime number (modulus). To protect the secret, it is encrypted with the public key of the TPA, $KPUB_{TPA}$, resulting in R0. The data owner sends the tuple (filename, R0) to the TPA [23, 24].

Third-party auditor: The TPA plays a crucial role in the protocol. They generate a key pair consisting of the public key $KPUB_{TPA}$ and the private key $KPRI_{TPA}$. The public key, $KPUB_{TPA}$, is published as common knowledge among all

parties. The TPA receives the tuple (filename, S) from the data owner and decrypts R0 using the private key $KPRI_{TPA}$, resulting in S. The TPA then stores the filename and S in its local database for future reference.

Figure 6 depicts the flow of the ITZKP algorithm, consisting of two main phases: challenge initialization phase and challenge phase.

Challenge initialization phase: The challenger, represented as a component in the figure, generates a large random number A. It computes g^A mod N, denoted as C0, and sends C0 to the prover (storage service). The prover, in turn, generates a large random number B, computes g^B mod N, denoted as C1, and sends C1 back to the challenger. Both the challenger and the prover also compute g^C0 mod N and g^C1 mod N, respectively, denoted as CE.

Challenge phase: The challenger generates another large random number, R, and encrypts (filename, g^R mod N) using CE, resulting in C. C is then sent to the prover. The prover receives C and decrypts it using CE, obtaining the filename and R. The prover retrieves the data associated with the filename, denoted as 'data', and computes its hash value, H(data), resulting in D. The prover computes the secret, S, using the formula g^D mod N. Finally, the prover computes the response R^S mod N, denoted as P, and sends the proof P to the TPA.

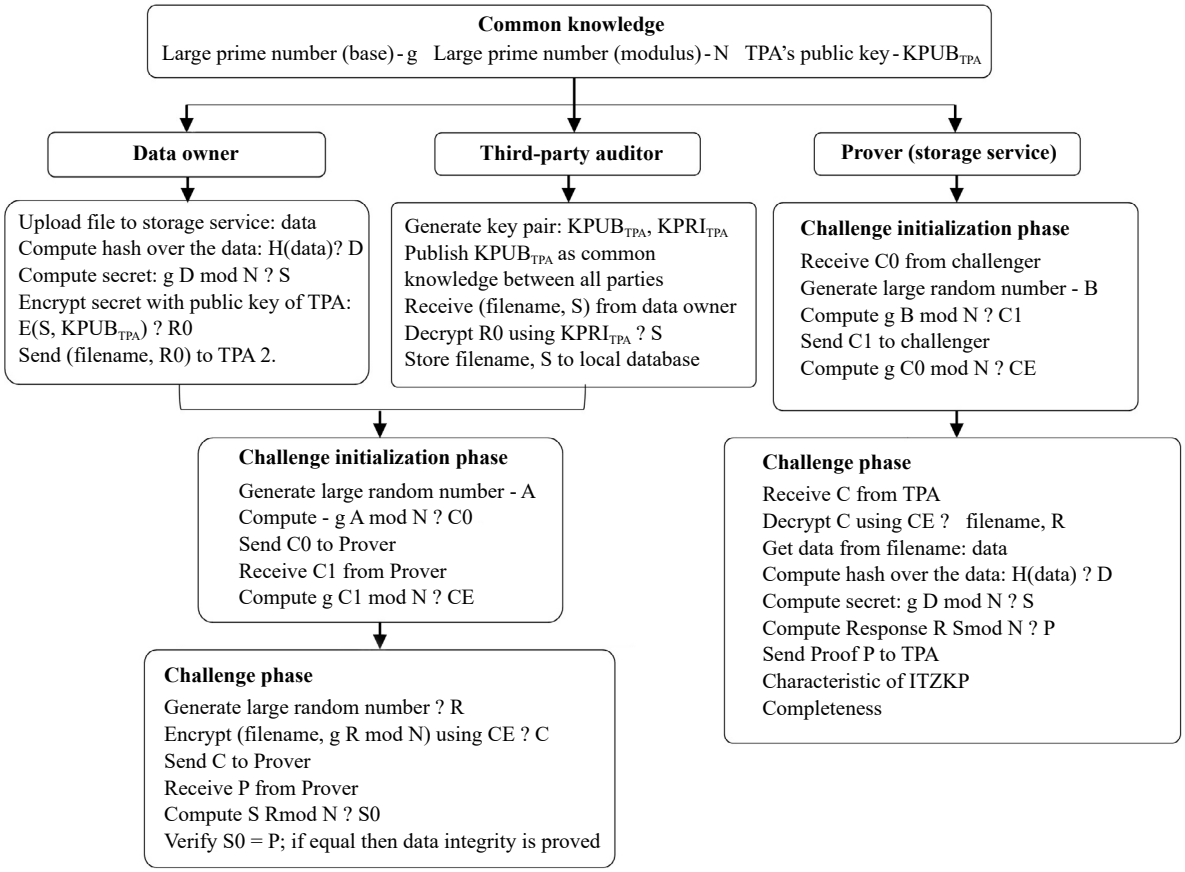


Figure 6. ITZKP algorithm

3. Results and analysis

The results of the experiments conducted to compare the HR-ACM method with the RBAC method reveal significant differences in execution time. The comparative analysis of the execution time of the HR-ACM method and RBAC method is depicted in Figure 7, considering varying role counts. The findings indicate that the HR-ACM approach exhibits superior performance in terms of execution time compared to the RBAC approach. This is evidenced by consistently lower time values observed for the HR-ACM approach across varying role counts. This suggests that the HR-ACM approach exhibits greater efficiency and expedites access control procedures. The observed variance in the duration of execution implies that

the HR-ACM approach is comparatively more optimized and efficacious in administering access control in the system. The enhanced effectiveness of the HR-ACM technique can be attributed to its utilization of a hierarchical framework and a role-based methodology. The HR-ACM approach streamlines access control procedures by arranging roles in a hierarchical structure, leading to expedited and more effective authorization processes. The RBAC approach demonstrates a significant increase in its execution time as the quantity of roles grows. The HR-ACM method demonstrates a consistent and relatively short implementation time across all role counts, ranging from 0.001 to 0.002 seconds. According to the results, it can be inferred that the HR-ACM method presents a higher level of efficiency and time-effectiveness in comparison to the RBAC method. The RBAC method experiences a noticeable decrease in performance and prolonged execution durations as the number of roles increases. In contrast, the HR-ACM method ensures a uniform and optimized duration of execution, regardless of the number of roles involved. This result highlights the benefits of employing the HR-ACM methodology in situations that entail a greater quantity of roles. The implementation of the HR-ACM methodology can lead to expedited and optimized execution of access control procedures within organizations, resulting in enhanced operational efficiency and heightened system responsiveness. The HR-ACM method's exceptional efficacy can be ascribed to its inventive configuration that integrates hierarchical role-based access control principles.

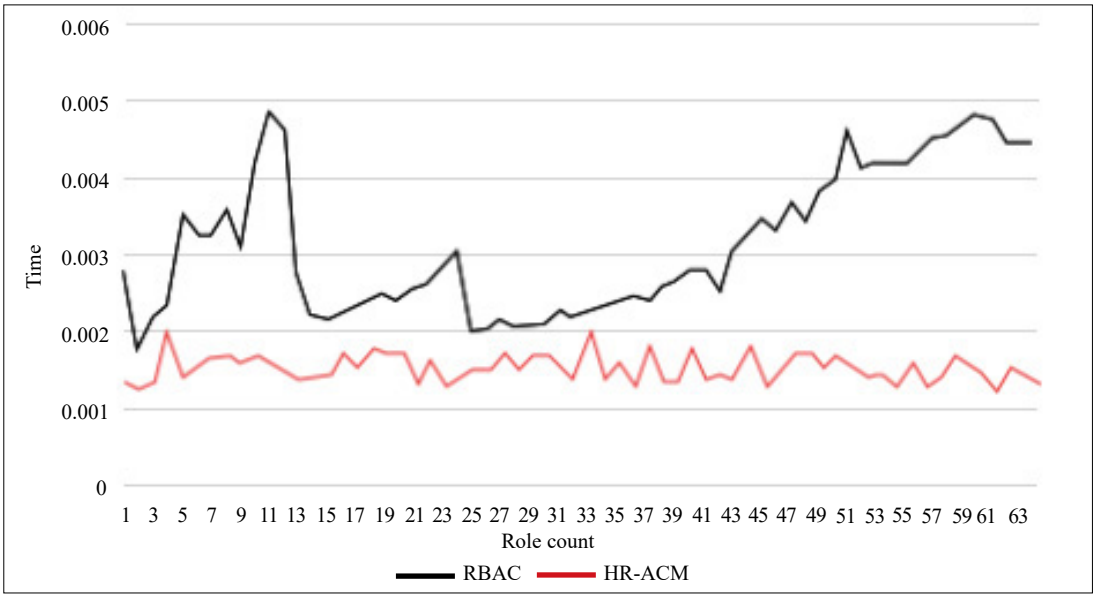


Figure 7. Execution time comparison between HR-ACM method and RBAC method

Figure 7 displays a graphical representation where the horizontal axis represents the quantity of roles, while the vertical axis indicates the duration of execution measured in seconds. The visual illustration portrays a dichotomous presentation, wherein one line signifies the HR-ACM approach while the other delineates the RBAC technique. Each data point depicted on the graph corresponds to a distinct count of roles and its corresponding duration of execution for each method. Upon examination of the graph, it becomes apparent that the HR-ACM method consistently demonstrates reduced execution times across all role counts in comparison to the RBAC method. The HR-ACM method exhibits a consistent trajectory, suggesting a uniform range of execution time within the interval of 0.001 to 0.002 seconds.

The observation suggests that the HR-ACM method demonstrates effectiveness and consistency regardless of the number of roles involved in the access control process.

On the other hand, the line representing the RBAC Method exhibits a distinct upward trend as the role count increases. The graph demonstrates that as the number of roles increases, the execution time for the RBAC method also increases significantly. The noted escalation in the deterioration of performance highlights the probable constraints of the RBAC approach in scenarios that entail a larger quantity of roles. The visual representation of the data reinforces the findings, emphasizing the superior swiftness of the HR-ACM method in terms of its execution. The HR-ACM method has

been shown to exhibit efficacy and flexibility in scenarios involving fluctuating role counts, as indicated by its uniform and diminished execution durations, as illustrated by the even trajectory of its graph. The graph presents a succinct and lucid depiction of the comparative analysis of the execution time between the HR-ACM method and RBAC method. The graphical depiction of the data improves comprehension of the performance disparities between the two techniques, empowering scholars, and professionals to make knowledgeable determinations about the choice of a suitable access control mechanism that aligns with their performance prerequisites.

To prove its efficacy and superiority, the execution time of the HR-ACM approach was compared to DAC (discretionary access control) and ABAC. The comparative analysis assessed access control techniques' execution duration efficiency and efficacy. The research involved 64 role tallies and their efficacy times. The results indicate that the three methodologies differ in implementation length. HR-ACM had a significantly lower execution time than DAC and ABAC.

To help explain the execution time differences between HR-ACM, DAC, and ABAC methods, Figure 8 was created. The graph's abscissa shows roles and the ordinate shows execution time in seconds. Figure 8 compares HR-ACM, DAC, and ABAC execution times. Three lines represent each technique.

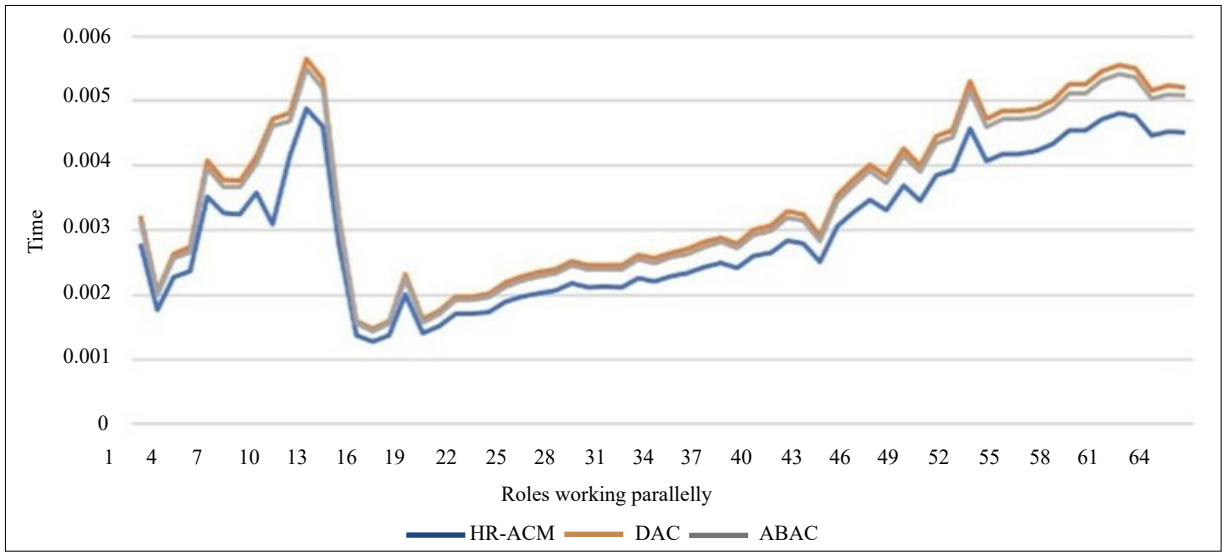


Figure 8. Execution time comparison between HR-ACM method and RBAC method

After scrutinizing the graph, it becomes evident that the HR-ACM method displays consistently reduced execution times across all role counts. The trajectory of the HR-ACM approach exhibits a relatively low slope, indicating a consistent and efficient pattern of execution durations. The findings underscore the HR-ACM method's performance superiority, as it consistently outperforms the DAC and ABAC methods in runtime.

In terms of comparison, the trajectories illustrated by the DAC and ABAC methodologies demonstrate a certain level of resemblance, indicating that their respective durations of execution are likely to be similar. Both methodologies exhibit slightly higher execution times compared to the HR-ACM method, indicating that they may require additional time to perform access control operations

To conduct a comprehensive assessment of the HR-ACM approach, a comparative analysis was performed versus the extant access control mechanisms, namely DAC and MAC (mandatory access control). Table 1 presents a comparison and summary of the attributes of these methodologies. The tabular data depicts a juxtaposition of the attributes inherent in the DAC, MAC, and HR-ACM methodologies. The DAC approach confers user privileges without regard to hierarchical structures, whereas the MAC methodology prioritizes security integrity. The HR-ACM approach, in contrast, allocates user privileges according to a hierarchical framework. In addition, it can be noted that both the MAC and the HR-ACM are designed to provide data integrity, whereas the DAC method does not offer this feature. In terms of authorization management, it is only the HR-ACM that includes this functionality. Regarding dynamics, it can be observed that both the DAC and MAC methods lack dynamic access control, whereas the HR-ACM method presents a degree of partial dynamic

control. Finally, with regards to flexibility, it can be observed that both the HR-ACM and MAC techniques demonstrate flexibility, while the DAC approach lacks this characteristic.

Table 1. Comparison with existing model

Characteristic	Access control		
	DAC	MAC	HR-ACM
Method	It gives rights to user	It provides security cleanness	Rights to user provided based on hierarchy
Integrity	No	Yes	Yes
Authorization management	No	No	Yes
Dynamic	No	No	Partial
Flexibility	No	No	Yes

The analysis comparing the execution times of HR-ACM, DAC, and MAC techniques indicates that the HR-ACM approach exhibits superior performance compared to the other two methods across multiple model characteristics. The HR-ACM approach exhibits greater dynamism and flexibility as compared to the DAC and MAC techniques, owing to its provision of additional features. This suggests that HR-ACM has the capacity to streamline and enhance access control procedures in a more efficient manner.

In addition to evaluating the performance of the HR-ACM method, the execution times of the Composikey, identity-based proxy re-encryption (IBPRE), and k-times proxy re-encryption (KPRE) models were compared for different file sizes. This comparison aimed to assess the efficiency of these encryption algorithms in terms of execution time while ensuring the security of critical information and maintaining access for authorized users. The execution times were measured for file sizes ranging from 1 KB to 2 GB. Table 2 presents the execution times for the IBPRE, KPRE, and Composikey models across various file sizes. To facilitate a visual comparison of the execution times for all three methods, a bar graph was generated (Figure 9).

Table 2. Execution time for different file sizes

File size	IBPRE	KPRE	Composikey
1 KB	3.41	3.503	2.23
100 KB	3.96	4.068	3.16
1 MB	4.4	4.52	3.12
5 MB	18.7	19.21	14.5
10 MB	35.2	36.16	25
100 MB	309.1	317.53	225
1 GB	2860	2938	2150
2 GB	5390	5537	3500

The graph in Figure 9 provides a clear visual representation of the execution time comparison among the Composikey, IBPRE, and KPRE models for various file sizes. The x-axis represents the file sizes, while the y-axis represents the

execution time in seconds. Analyzing the graph, it can be observed that the Composikey method consistently exhibits lower execution times compared to the IBPRE and KPRE models for all file sizes. This finding highlights the advantage of the Composikey method in terms of efficiency and execution speed, indicating its potential for delivering faster encryption operations. Furthermore, the graph demonstrates that the execution times for the IBPRE and KPRE models are nearly identical across all file sizes. This suggests that these two methods offer similar performance characteristics in terms of execution time.

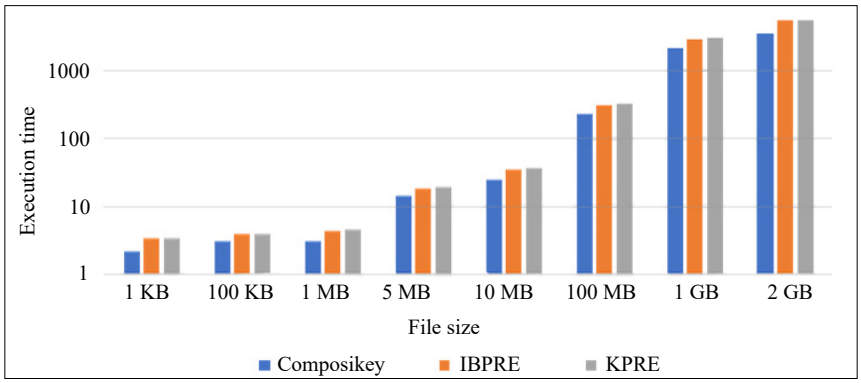


Figure 9. Execution time comparison among Composikey, IBPRE, and KPRE models for different file sizes

The results emphasize the superiority of the Composikey method in terms of execution time. Its consistently lower execution times across different file sizes make it an attractive choice for securing critical information while minimizing processing time. Overall, the comparison of execution times among the Composikey, IBPRE, and KPRE models reveals the efficiency and effectiveness of the Composikey method in achieving faster encryption operations. The graph in Figure 10 serves as a visual representation of the performance differences among the three methods, further supporting the findings.

In addition to the execution time comparison, the vulnerability of the algorithms to various types of attacks was evaluated. Table 3 presents a comparison of the DES (data encryption standard), IBPRE, and Composikey algorithms concerning their vulnerability to different types of attacks. It is observed that the IBPRE algorithm is resistant to known-plaintext attacks, while the DES algorithm and Composikey algorithm are not susceptible to such attacks. The DES algorithm is vulnerable to chosen-plaintext and ciphertext-only attacks, whereas the IBPRE algorithm is not susceptible to these attacks. Remarkably, the Composikey algorithm is not susceptible to any of the analyzed attacks.

Table 3. Types of attacks and comparison of algorithms

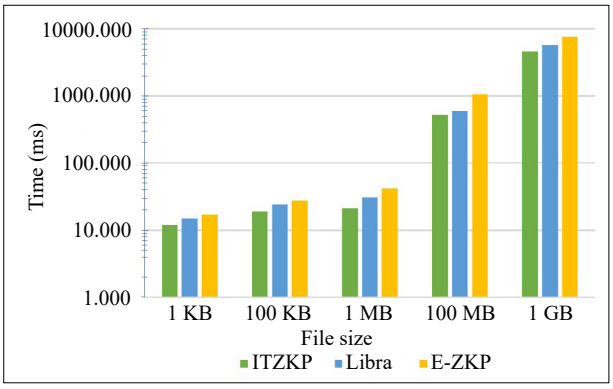
Characteristic	Algorithm		
	DES	**IBPRE**	**Composikey**
Known-plaintext attack	No	Yes	No
Chosen-plaintext attack	Yes	No	No
Ciphertext-only attack	Yes	Yes	No
Chosen-ciphertext attack	Yes	Yes	No

The ITZKP was successfully implemented for secure auditing checks by calculation, ensuring the confidentiality and integrity of the data without disclosing any inputs or intermediate results. The file was divided into 8 chunks, each with its own hash concatenated with the hashes of its immediate children, providing a secure and efficient data organization.

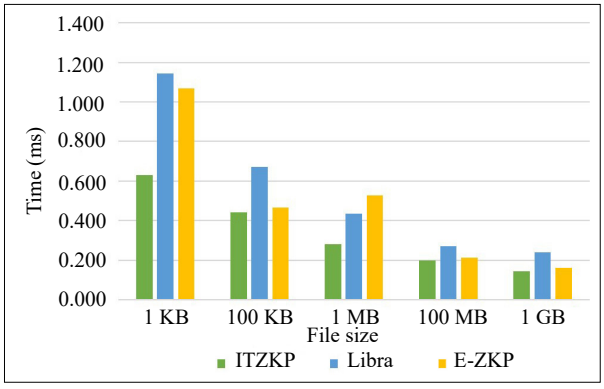
To evaluate the performance of the proposed ITZKP system, three attributes were considered for comparison: challenge time, response time, and verifier time. These attributes were compared with those of two existing systems, ZKP and Libra, under the same testing conditions for different file sizes ranging from 1 KB to 1 GB. The objective was to assess the efficiency and effectiveness of the proposed ITZKP system in comparison to the existing systems.

Figure 10(a) presents the results of the challenge time in milliseconds. The bar chart clearly demonstrates that the proposed ITZKP system requires less time compared to the existing ZKP (E-ZKP) and Libra systems. This indicates that the ITZKP system offers improved efficiency and faster processing time for handling challenges, ensuring quick and reliable verification processes. Figure 10(b) illustrates the response time measured in microseconds for the existing ZKP (E-ZKP) and Libra systems for different file sizes, ranging from 1 KB to 1 GB. The ITZKP system was used to verify the prover's response. The graph shows that the response time for the ITZKP system is comparable to the existing systems, indicating its effectiveness in generating timely responses without compromising security. The verifier time, as shown in Figure 10(c), is another critical aspect of the auditing process. The graph displays the time taken by the verifier to verify the authenticity of the data. The results reveal that the verifier time for the ITZKP system aligns closely with the existing systems, demonstrating its ability to perform efficient verification processes.

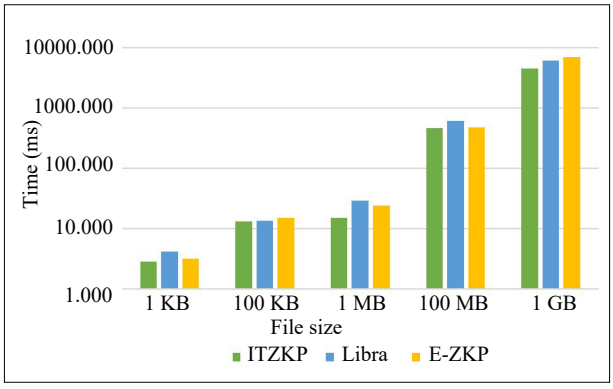
Additionally, the ITZKP system was subjected to testing to evaluate its resilience against potential attacks. Table 4 showcases the number of iterations carried out by the algorithm under different chances of attacks. The results indicate that the number of iterations increases as the chances of attack escalate, highlighting the system's ability to adapt and strengthen its security measures in response to potential threats.



(a) Comparison of challenge time



(b) Comparison of verifier time



(c) Comparison of response time

Figure 10. Comparison of challenge time, verifier time and response time

Table 4 shows the relationship between attack probability and ITZKP system iterations. To protect against attacks, the system iterates more. The analysis and comparison of the ITZKP system with existing systems shows its efficiency

and effectiveness in securing auditing checks. ITZKP outperforms ZKP and Libra in challenge, response, and verifier times. Its adaptability and resilience make it reliable in data integrity and confidentiality.

Table 4. Number of iterations for different chances of attack

Chances of attack	Number of iterations
10%	5
20%	7
40%	9
50%	11

3.1 *Performance analysis*

In this section, multiple users access the system in an Ethernet LAN with 100 Mbps network speed to evaluate its performance. Uploading is 3.61 Mbps and downloading is 2.87 Mbps. The evaluation device has a 32-bit operating system and a 2.27 GHz × 64-based Intel i3 processor. Performance analysis evaluates the system's responsiveness, efficiency, and ability to handle multiple user requests. To assess the system's performance, experiments are conducted in a controlled network environment. Response time, throughput, and resource utilization are assessed. The system's response time is measured. Throughput measures the system's data processing speed. Resource utilization assesses CPU and memory usage. The data in Table 5 shows that the system performs well in the network environment. User requests are processed quickly because the response time is acceptable. Throughput measurements show that the system can handle the specified downloading and uploading speeds, allowing users to transfer data smoothly. The system optimally uses hardware resources, ensuring performance without resource bottlenecks.

Table 5. Time analysis

File size	System	
	Existing system	Proposed system
1 Mbps	250	175
5 Mbps	700	500
10 Mbps	1000	800
15 Mbps	1500	1000

3.2 *Security module*

The security module enhances CSP (cloud service provider) and user security. Encryption, strong key sharing, and authentication safeguard data. Identity-based key exposure resilient auditing secures the proposed cloud storage system. System security is enhanced by robust auditing mechanisms that resist key exposure risks. Only authorized users can decrypt sensitive cloud storage data. Encryption algorithms protect data. Strong key sharing and authentication secure cloud storage provider-user communication. These processes verify user identities and exchange cryptographic keys to limit system access to authorized users. This auditing scheme protects data integrity and authenticity. Identity-based auditing prevents data tampering and access. Encryption, strong key sharing and authentication, and identity-based key exposure resilient auditing safeguard data in cloud storage.

4. Discussion

This research has yielded significant findings regarding the improvement of security measures in cloud computing settings by implementing innovative approaches such as HR-ACM, Composikey, and ITZKP. Nonetheless, there exist various potential areas for future research that can propel the domain of cloud security to greater heights. An area of potential future investigation pertains to the concept of scalability. The assessment of scalability of proposed methodologies is crucial considering the increasing size and complexity of cloud environments. Subsequent inquiry may encompass the evaluation of the efficacy and effectiveness of HR-ACM, Composikey, and ITZKP methodologies in the context of cloud infrastructures that are progressively more extensive. They would yield significant perspectives on their capacity to manage an increased quantity of data and user inquiries, guaranteeing the efficacy and efficiency of security protocols in cloud environments that are subject to change.

An additional area that warrants investigation in the future pertains to the resilience of the suggested methodologies in the face of sophisticated forms of attacks. The evaluation of the resilience of HR-ACM, Composikey, and ITZKP against advanced intrusion attempts or newly emerging cryptographic vulnerabilities is of utmost importance in light of the evolving nature of security threats. It is possible to perform thorough examinations and computer-based models to detect possible vulnerabilities and develop strategies to reduce the impact of these emerging hazards. This approach can improve the overall security stance of cloud computing systems. The validation of the practical applicability and effectiveness of the proposed techniques requires an important subsequent step, which is the deployment in real-world scenarios [24]. Undertaking comprehensive experimentation and performance assessments across a range of cloud computing environments would yield valuable insights into their practical feasibility and efficacy. Through the experimentation of methodologies across diverse scenarios and datasets, scholars can acquire a comprehensive comprehension of their efficacy and pinpoint any pragmatic obstacles that necessitate resolution.

5. Conclusions

This study evaluates various cloud computing security methods. HR-ACM, Composikey, and ITZKP have improved data security, access management, and authenticity. HR-ACM's strong security framework efficiently allocates user roles, verifies their actions, and restricts unauthorized cloud resource access. Composikey, a composite key encryption algorithm, has increased data security while maintaining access for authorized users. ITZKP's computation-based secure auditing checks have protected inputs and intermediate outcomes, ensuring data privacy throughout the process. Our experiments and performance evaluations proved the methodologies' efficacy and efficiency. The HR-ACM approach outperforms conventional RBAC frameworks in execution time and scalability. The Composikey encryption algorithm protects cloud data by outperforming previous methods. The ITZKP protocol also securely authenticates data integrity while protecting confidential data. The findings of this study contribute to the body of knowledge in cloud security and provide valuable insights for practitioners and researchers aiming to develop efficient and robust security architectures for cloud computing environments.

Our comparative analysis suggests the proposed methodologies improve security, performance, and privacy. HR-ACM provides precise access control and efficient user role management. The Composikey algorithm protects sensitive data, while the ITZKP protocol conducts audit checks securely and privately.

However, more research and improvement are possible. Next research may examine the scalability of the suggested methods in more complex and large-scale cloud environments. Their ability to withstand sophisticated attacks and implementation in real-world scenarios would reveal their practicality. Incorporating emerging technologies like machine learning and blockchain could improve security, while addressing privacy concerns would ensure responsible data management. To conclude, HR-ACM, Composikey, and ITZKP work well together to build resilient and secure cloud computing frameworks. Our study shows that technology can solve major security issues and provide a solid foundation for cloud-based solutions. These methods and further research can create a secure, private cloud computing environment. This can boost user trust and cloud technology adoption.

6. Future scope

There is potential for further investigation into the amalgamation of the suggested methodologies with nascent technologies such as machine learning, blockchain, or edge computing. The exploration of the potential synergies and advantages arising from the integration of these technologies can make a valuable contribution to the advancement of more resilient and effective security frameworks for cloud-based settings. Through the utilization of nascent technologies, scholars can effectively tackle distinct security hurdles and augment the comprehensive security and confidentiality facets of cloud computing systems.

Finally, the preservation of privacy in the context of cloud computing is a topic that requires additional research. The research paper primarily cantered on the topics of data protection and access control. However, there is potential for future work to explore additional techniques that prioritize the preservation of user privacy, facilitate secure data sharing, and adhere to the ever-changing landscape of privacy regulations. The implementation of robust security measures and prioritization of user privacy in cloud architectures would foster trust among users and facilitate responsible data handling practices.

Conf1ict of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Prakash V, Williams A, Garg L, Savaglio C, Bawa S. Cloud and edge computing-based computer forensics: Challenges and open problems. *Electronics*. 2021; 10(11): 1229. Available from: https://doi.org/10.3390/electronics10111229.

[2] Sabitha S, Rajasree M. Access control schemes in cloud: Taxonomy and performance. *International Journal of Applied Engineering Research*. 2019; 14(23): 4209-4220.

[3] Solanki N, Huang Y, Yen IL, Bastani F, Zhang Y. Resource and role hierarchy based access control for resourceful systems. In: *2018 IEEE 42nd Annual Computer Software and Applications Conference* (*COMPSAC*). Tokyo, Japan: IEEE; 2018. p.480-486. Available from: https://doi.org/10.1109/COMPSAC.2018.10280.

[4] Abo-alian A, Badr NL, Tolba MF. Hierarchical attribute-role based access control for cloud computing. In: Gaber T, Hassanien A, El-Bendary N, Dey N. (eds.) T*he 1st International Conference on Advanced Intelligent System and Informatics* (*AISI2015*). Advances in Intelligent Systems and Computing, vol 407. Cham: Springer; 2016. p.381-389.Available from: https://doi.org/10.1007/978-3-319-26690-9_34.

[5] Wan Z, Liu J, Deng RH. HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Transactions on Information Forensics and Security*. 2012; 7(2): 743-754. Available from: https://doi.org/10.1109/TIFS.2011.2172209.

[6] Bhartiya S, Mehrotra D, Girdhar A. Proposing hierarchy-similarity based access control framework: A multilevel Electronic Health Record data sharing approach for interoperable environment. *Journal of King Saud University-Computer and Information Sciences*. 2017; 29(4): 505-519. Available from: https://doi.org/10.1016/j.jksuci.2015.08.005.

[7] Zhou L, Varadharajan V, Hitchens M. Achieving secure role-based access control on encrypted data in cloud storage. *IEEE Transactions on Information Forensics and Security*. 2013; 8(12): 1947-1960. Available from: https://doi.org/10.1109/TIFS.2013.2286456.

[8] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano D, Fazio N, Gennaro R, Nicolosi A. (eds.) *Public Key Cryptography – PKC 2011*. Lecture Notes in Computer Science, vol 6571. Berlin: Springer; 2011. p.53-70. Available from: https://doi.org/10.1007/978-3-642-19379-8_4.

[9] Ateniese G, Benson K, Hohenberger S. Key-private proxy re-encryption. In: Fischlin M. (ed.) *Topics in Cryptology – CT-RSA 2009*. Lecture Notes in Computer Science, vol 5473. Berlin: Springer; 2009. p.279-294. Available from: https://doi.org/10.1007/978-3-642-00862-7_19.

[10] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data.

In: *Proceedings of the 13th ACM conference on Computer and communications security*. New York: ACM; 2006. p.89-98. Available from: https://doi.org/10.1145/1180405.1180418.

[11] Zhang F, Fan X, Lei X, Wu J, Song J, Huang J, et al. Zero knowledge proofs for cloud storage integrity checking. In: *2020 39th Chinese Control Conference* (*CCC*). Shenyang, China: IEEE; 2000. p.7661-7668. Available from: https://doi.org/10.23919/CCC50068.2020.9189231.

[12] Yu Y, Li Y, Au MH, Susilo W, Choo KK, Zhang X. Public cloud data auditing with practical key update and zero knowledge privacy. In: Liu J, Steinfeld R. (eds.) *Information Security and Privacy. ACISP* 2016. Lecture Notes in Computer Science, vol 9722. Cham: Springer; 2016. p.389-405. Available from: https://doi.org/10.1007/978-3-319-40253-6_24.

[13] Yang C, Zhang M, Jiang Q, Zhang J, Li D, Ma J, et al. Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities. *Pervasive and Mobile Computing*. 2017; 41: 243-258. Available from: https://doi.org/10.1016/J.PMCJ.2017.03.014.

[14] Bick A, Kol G, Oshman R. Distributed zero-knowledge proofs over networks. In: Naor J, Buchbinder N. (eds.) *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms* (*SODA*). Society for Industrial and Applied Mathematics; 2022. p.2426-2458. Available from: https://doi.org/10.1137/1.9781611977073.97.

[15] Cheng X, Wang T. Trust value of the role access control model based on trust. In: Wong W. (ed.) *Proceedings of the 4th International Conference on Computer Engineering and Networks*. Lecture Notes in Electrical Engineering, vol 355. Cham: Springer; 2015. Available from: https://doi.org/10.1007/978-3-319-11104-9_21.

[16] Duan J, Gao D, Foh CH, Zhang H. TC-BAC: A trust and centrality degree based access control model in wireless sensor networks. *Ad Hoc Networks*. 2013; 11(8): 2675-2692. Available from: https://doi.org/10.1016/J.ADHOC.2013.05.005.

[17] Namasudra S, Roy P, Vijayakumar P, Audithan S, Balusamy B. Time efficient secure DNA based access control model for cloud computing environment. *Future Generation Computer Systems*. 2017; 73: 90-105. Available from: https://doi.org/10.1016/J.FUTURE.2017.01.017.

[18] Sookhak M, Yu FR, Khan MK, Xiang Y, Buyya R. Attribute-based data access control in mobile cloud computing: Taxonomy and open issues. *Future Generation Computer Systems*. 2017; 72: 273-287. Available from: https://doi.org/10.1016/J.FUTURE.2016.08.018.

[19] Perez JM, Perez GM, Gomez AF. SecRBAC: Secure data in the Clouds. *IEEE Transactions on Services Computing*. 2016; 10: 726-740. Available from: https://doi.org/10.1109/TSC.2016.2553668.

[20] Liu JK, Steinfeld R. (eds.) *Information security and privacy*. Cham: Springer; 2016. Available from: https://doi.org/10.1007/978-3-319-40253-6.

[21] Xia Y, Xia F, Liu X, Sun X, Liu Y, Ge Y. An improved privacy preserving construction for data integrity verification in cloud storage. *KSII Transactions on Internet & Information Systems*. 2014; 8(10): 3607-3623. Available from: https://doi.org/10.3837/TIIS.2014.10.019.

[22] Sun X, Chao HC, You X, Bertino E. (eds.) *Cloud computing and security*. Cham: Springer; 2017. Available from: https://doi.org/10.1007/978-3-319-68542-7.

[23] Sumathi D, Kathik S. Proof of retrievability using elliptic curve digital signature in cloud computing. International *Review on Computers and Software*. 2014; 9: 1526-1532. Available from: https://doi.org/10.15866/IRECOS.V9I9.2200.

[24] Satyanarayan R. A novel approach for energy-efficient container migration by using gradient descent Namib beetle optimization. *Contemporary Mathematics*. 2023; 5(1). Available from: https://doi.org/10.37256/cm.5120243085.