Velmurugan *et al.*
Journal of Cloud Computing (2024) 13:86
https://doi.org/10.1186/s13677-024-00634-8

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Provably secure data selective sharing scheme with cloud-based decentralized trust management systems

S. Velmurugan[1], M. Prakash[2], S. Neelakandan[3] and Arun Radhakrishnan[4*]

Abstract

The smart collection and sharing of data is an important part of cloud-based systems, since huge amounts of data are being created all the time. This feature allows users to distribute data to particular recipients, while also allowing data proprietors to selectively grant access to their data to users. Ensuring data security and privacy is a formidable task when selective data is acquired and exchanged. One potential issue that emerges is the risk that data may be transmitted by cloud servers to unauthorized users or individuals who have no interest in the particular data or user interests. The prior research lacks comprehensive solutions for balancing security, privacy, and usability in secure data selective sharing schemes inside Cloud-Based decentralized trust management systems. Motivating factors for settling this gap contain growing concerns concerning data privacy, the necessity for scalable and interoperable frameworks, and the increasing dependency on cloud services for data storage and sharing, which necessitates robust and user-friendly mechanisms for secure data management. An effective and obviously secure data selective sharing and acquisition mechanism for cloud-based systems is proposed in this work. We specifically start by important a common problematic related to the selective collection and distribution of data in cloud-based systems. To address these issues, this study proposes a Cloud-based Decentralized Trust Management System (DTMS)-connected Efficient, Provably Secure Data Selection Sharing Scheme (EPSDSS). The EPSDSS approach employs attribute-based encryption (ABE) and proxy re-encryption (PRE) to provide fine-grained access control over shared data. A decentralized trust management system provides participant dependability and accountability while mitigating the dangers of centralized trust models. The EPSDSS-PRE paradigm would allow data owners to regulate granular access while allowing users to customize data collection without disclosing their preferences. In our strategy, the EPSDSS recognizes shared data and generates short fingerprints for information that can elude detection before cloud storage. DTMS also computes user trustworthiness and improves user behaviour administration. Our research demonstrates that it's able to deliver trustworthy and safe data sharing features in cloud-based environments, making it a viable option for enterprises seeking to protect sensitive data while maximizing collaboration and utilization of resources.

Keywords Data security, Efficient Provably Secure Data Selection Sharing Scheme (EPSDSS), Decentralised Trust Management Systems (DTMS), Attribute-Based Encryption (ABE), Proxy Re-Encryption (PRE)

*Correspondence:
Arun Radhakrishnan
arun.radhakriahnan@ju.edu.et
Full list of author information is available at the end of the article

Introduction

Artificial intelligence (AI) can be deployed more quickly due to the large amount of training data that 5G and Internet of Things (IoT) technology developments have made available. In the context of data governance and sharing, protecting privacy and guaranteeing data security have become crucial problems at the same time. Strong risks to individual privacy have been brought about by powerful data mining and analysis. In the past, the majority of users made the decision to share and distribute their data through cloud servers. The vast majority of data stored in cloud storage, especially that generated by IoT devices with direct connections to human life, is extremely sensitive. A person may face serious problems if their personal information is unlawfully obtained or leaked and is connected to their real identity. These data have certain features and may include personal information about their life, work, and health. For all contemporary businesses using big data and AI, then, integrating data and generating value while preserving data security and privacy has become an essential problem [1].

Cloud computing has become widely accepted in both personal and professional domains because to the rapid advances in computer science and the advent of concepts like big data and the Internet of Things (IoT). Our work methods and way of life have been profoundly impacted by this technology [2, 3]. The advantages of cloud storage, such as its vast storage capacity, simple usage, high degree of flexibility, and lack of platform limitations, have drawn the attention of academics and engineers [4]. One element of cloud computing is cloud storage. Because cloud storage systems offer advantages over local storage, more companies and individuals are moving their data to them [5, 6].

Data owners in cloud computing must rely on third-party cloud storage providers to manage their data, data integrity is crucial. To solve this issue, scientists are developing more and more innovative algorithms for explanations of cloud storage data integrity. These algorithms could preserve the accuracy of outsourced data while enhancing security [7]. Despite the availability of multiple data services, data owners are apprehensive to commit their critical data to cloud service providers (CSPs) for third-party cloud storage due to concerns about CSP integrity [8], as well as the shared nature of cloud storage environment. Cloud computing, which primarily includes both computational and data storage components, is very similar to infrastructure as a service (IaaS) and cloud storage. The exact location of externally managed data in cloud storage or the computers handling those operations are frequently unknown to cloud users utilizing Infrastructure as a Service (IaaS). Because of this, cloud storage data privacy presents a significant security risk, which is exacerbated by the existence of dishonest users and leads to issues with data integrity and confidentiality. As cloud computing depends on remote data storage, cloud storage security is important to its success. This reliance increases serious problems for cloud storage. In accordance with the ACID (Atomicity, Consistency, Isolation, Durability) principles that manage transactions, data in integrity is a essential component of database system management (DBMS). It includes completeness, correctness, and consistency. When CSPs are unable to safely ensure that the data they provide in response to client inquiries is accurate and comprehensive, a problem arises [9].

One of the most important technologies for safe resource sharing is access control. Identity authentication and authority distribution are two essential access control technologies that prevent unauthorized users from accessing resources while enabling only authorized legal users to perform it. However, access control technology must handle two critical challenges immediately: user privacy protection and frequent permission changes. To overcome these concerns, this research presents a privacy-preserving dynamic access control paradigm [10]. With the advancements made by researchers in remote data auditing, businesses no longer need to rely solely on local data backup to safeguard the accuracy of outsourced data. This innovation verifies the accuracy of the data being inspected and helps conserve bandwidth and communication resources.

The user interacts with CSP to obtain proof of the original data, which confirms the data's accuracy. Though, frequent user involvement with CSP, audit processes, and routine data accuracy checks can use many computing and network resources. Researchers devised a mechanism known as the third-party auditor to make the deployment of public auditing easier. Users can delegate the auditing work to a third-party auditor (TPA) using this strategy, provided they receive thorough reports on the auditing results [11]. Due to its evident benefits over private auditing in terms of pricing and practicality, public auditing has grown significantly in popularity within the auditing system.

On the other hand, the bulk of current public auditing methods assumes that TPA will be entirely dependable and carry out every audit honestly, significantly increasing security concerns. As an illustration, customers only receive the audit's findings from TPA, while users need to be made aware of the auditing procedure. The security of the user's data will be jeopardized if an unreliable third-party auditor only assures the user that the audit findings are valid by performing legitimate audit work. Additionally, being a centralized organization, TPA is vulnerable to internal and external flaws. If these repercussions

result in TPA system failure, it will impact the auditing procedure. It's feasible that TPA and CSP will cooperate against their interests to hide data corruption even if everything is in order [12].

Data owners implement access controls to limit access to their information to those who need it. Since cloud servers cannot be relied upon to evaluate access constraints and make access decisions reliably, it may be challenging to enforce these criteria. Sensitive information can be protected by encrypting it before sharing it with the right people. Inappropriately, as the number of users in the system grows exponentially, the amount of ciphertext copies generated for each data item also rises exponentially, making traditional public key encryption approaches unsuitable for this task. In a distributed environment, Attribute-Based Encryption (ABE) is considered to be the best technology for resolving issues with data security and privacy protection. As a result, researchers have recently employed ABE to accomplish fine-grained blockchain data access control. Data owners can create extremely detailed access control policies with ABE, defining exactly which qualities must be met in order to access encrypted data. Only those who are permitted and possess the required characteristics can decrypt and view the data due to this finely tuned control, which makes it possible to precisely manage data access [13].

In the decentralized storage network, data ownership, privacy, and accessibility would all be redefined. Using traffic surveillance cameras as an example, the information might be kept on a decentralized storage network. As a result, everyone can confirm that the data exist, but only authorized parties can access them. An encryption technique with access control is necessary for several entities (like insurance firms) to access data. Conventional symmetric or asymmetric cryptography cannot address this criterion since these techniques call for identifying the decryption key before the encryption key. The proxy re-encryption (PRE) method is suitable for exchanging data.

Traditional data exchange systems regularly fail to meet modern initiatives security, efficiency, and scalability necessities. Therefore cloud-based decentralized trust management systems have been showed to be viable way to address this challenge, it is recognizable that can technique for secure data selective sharing has been predictable. Organizations can improve security by integrating decentralized trust management systems into cloud-based data sharing frameworks, which distributes trust management responsibilities and ensures strong protection against illegal access and data breaches. Further, establishing an emphasis on efficiency paves the way for simpler data sharing methods, which in turn maximize resource use and reduce overhead costs. Furthermore, this method encourages confidence, transparency, and originality in cooperative ecosystems, while also serving the purposes of compliance and regulatory standards. Finally, the plan allows this research to fully benefit from cloud computing while safeguarding sensitive data and maintaining a competitive advantage in today's digital economy by tackling these major issues.

The main contribution of the proposed method is

- This work addresses these issues by providing a cloud-based decentralized trust management system (DTMS) connected to an Efficient, Provably Secure Data Selection Sharing Scheme (EPSDSS).
- The proposed EPSDSS system uses cutting-edge cryptographic techniques, including proxy re-encryption and attribute-based encryption, to allow fine-grained control over who has access to what in a shared database.
- A decentralized trust management system reduces the risks associated with centralized trust models by ensuring the reliability and accountability of participating entities.
- Users of the EPSDSS-PRE system can change data collection without disclosing their choices, and data owners are given exact control over access to their data.
- In our organization, the EPSDSS can identify shared data and provide brief fingerprints for data that can effectively evade detection before being stored in the cloud.
- The DTMS also evaluates user dependability and enhances user behaviour administration based on this calculation. Data merging reduces the overall quantity of data selection, the burden on users, and the cloud load.
- The proposed EPSDSS-PRE systems are robust enough to resist the following three types of forgery attacks: Impersonation of a cloud service provider, selective data forgery, and trust credential forgery.

The rest of this paper is organized as follows. Literature survey section covers related works. Literature survey section describes system model and goals. Proposed system section investigates some of the preliminary knowledge used in this paper and describes detailed implementation specifics. Result and discussion section deals with security and performance analysis. Conclusion section explores into the discussion of the conclusion and future approaches.

Literature survey

Data selective sharing scheme with access control

Zhu et al. [14], access control is a crucial security precaution for protecting private information and system assets. Blockchain is a foundational technology architecture that combines high value, low cost, and trust. Numerous scholars have attempted to address the shortcomings of conventional centralized access control by merging access control with blockchain technologies. Most current blockchain access control techniques rely on permission verification and on-chain storage.

Wang et al. [15] propose an attribute-based distributed access control system (ADAC) for IoT that employs blockchain technology. To provide more fine-grained access control in open and lightweight IoT devices, the suggested ADAC analyzes variables like the manufacturer and the object-specified attribute. We generate a smart contract framework that incorporates a subject contract (SC), an object contract (OC), an access control contract (ACC), and numerous policy contracts (PCs) to maintain and access IoT device attributes for distributed and trustworthy access control (DTAC). SC and OC are in charge of handling subject and object attribute information, correspondingly. PCs are used to administer access control policies. ACC creates permission decisions by retrieving qualities and strategies.

A secure control method for blockchain data access based on digital certificates is described by Liu et al. [16]. This explanation removes the certainty to authenticate third-party participants' encrypted identity signatures by relating blockchain integration and digital certificate technology to provide a safe authentication mechanism for private data within blockchains. Fair contract signing by numerous signers over the blockchain is made possible by the effective network forwarding mechanism selected in this study. Contract confidentiality and participant particularity can be protected by this protocol.

Yu et al. [17] analyze the Internet of Things data exchange paradigm using block chain ability. The significances of the test model show that the block-based chain proposed in this study provides enhanced security and privacy for IoT data sharing. Send write transactions at 100 TPS and query transactions at 250 TPS to optimize throughput. With a maximum write throughput of 60 transactions per second (TPS), the model clearly displays its implementation potential, outperforming Ethereum and Bitcoin on the public chain. Data sharing and storage are made possible by this design, which does not depend on a centralized third-party organization. Additionally, it creates participant confidence, which guarantees secure data sharing.

Conventional data sharing management

Xiang et al. [18] devised a method for verifying the accuracy of the information, and it relies heavily on the subjects chosen by the users. A relation authentication label was developed for this purpose. Notably, this strategy effectively conceals the keywords from the Cloud Service Provider (CSP) while adding little burden to the auditing procedures. These techniques use the public key (PK) for authentication after the private key (SK) is used to create homomorphic verifiable tags for the data. These systems do, however, run into issues with certificate administration. Managing certificates becomes more complicated when dealing with many users, significantly taxing the system.

Xiang et al. [18] devised a method for verifying the accuracy of the information, and it relies heavily on the subjects chosen by the users. A relation authentication label was developed for this purpose. Notably, this strategy effectively conceals the keywords from the Cloud Service Provider (CSP) while adding little burden to the auditing procedures. These techniques compute the homomorphic verifiable tags for the data using the private key SK and then use the public key PK to verify that they are accurate. These systems do, however, run into issues with certificate administration. Managing certificates becomes more complicated when dealing with many users, significantly taxing the system.

Fan et al. [19] Because of centralization, traditional CP-ABE systems lack credibility when storing and granting access policies through the internet. We address the aforementioned problem in this research by presenting a verifiable and secure one-to-many data exchange system. Cloud non-repudiation and user self-certification are made possible by the usage of blockchain to record the access rules. Our efficient certification scheme considers the computer capabilities of the vehicle user. In the meanwhile, we suggest a policy concealing strategy in light of the sensitive information found in the access policy. When a vehicle user wants to stop sharing data in vehicular social networks (VSNs), our approach also enables data revocation.

A blockchain-based access control architecture was put forth by Yang et al. [20]. AuthPrivacyChain designed unique identifiers using blockchain entity addresses for authentication and authorization processes. Because of the distributed nature of the blockchain, a new distributed and decentralized cloud access control architecture has been built further to increase the security and privacy of data applications. Wang et al. [21] suggested a trust management system based on a multi-criteria decision-making approach. Each vehicle in this model evaluates the messages' dependability and determines the sender's trust value. The closest Roadside Unit (RSU) receives the

trust values calculated by each car and processes them using a multi-criteria decision-making process. After a new block is generated, the RSU works to reach a consensus before adding the block to the blockchain. The RSU cannot act as miners in this system due to their high deployment costs, which is a limitation.

Trust management schemes in cloud computing environments

Tian et al. [22] provided a novel approach to decentralized trust management in identity-based multi-copy data-sharing audits. The group manager also assesses user credibility and improves user behaviour management based on trust values. Our system uses data merging to maximize efficiency, reducing the total number of data copies and lowering user and cloud overhead. Our solution, in particular, demonstrates strong security against forgery attempts from three different adversaries. To prove the effectiveness and viability of our strategy, we conduct thorough analyses of safety and performance and provide convincing evidence of its strong security measures and usefulness.

Gupta et al. [23] presented the SP-MAACS architecture to ensure complete security and privacy in multi-authority access control systems designed for exchanging healthcare data in the cloud. This technology promotes scalability and adaptability by enabling data owners to share their information with customers in both open and closed domains. Our implementation results show that this method improves decryption effectiveness while maintaining privacy, providing adaptive security within the conventional approach. By assigning decryption work to proxy servers in the future, decryption efficiency can be increased even more. One of the areas of healthcare data management and privacy protection that is now receiving the most attention is integrating the suggested control mechanism with blockchain technology. Achieving this connection could improve security, privacy, and auditability.

Li et al. [24] proposed cloud trust architectures typically have a centralized layout, which can result in high administrative costs, increased network traffic, and even single points of failure. The conclusions of trust ratings are also not widely accepted because of issues with openness and traceability. Therefore, blockchain technology is perfect for creating distributed and decentralized trust infrastructures. This essay looks into the potential for using blockchain-based trust mechanisms in existing public cloud environment. A multi-authority ABE that protects privacy while dynamically changing policies was suggested by Yan et al. [25]. Although this approach was developed for a situation involving several authorities, it proved ineffective in preventing dishonest individuals from disclosing their private keys. An attribute-level privacy and search system for encrypted data was proposed by Najafi et al. [26]. A safe and convenient method of storing and retrieving patient data was designed. This novel method successfully protected sensitive information by demonstrating its resistance to common keyword-guessing assaults.

Ruan et al. [27] introduced the Policy-Hiding and Multi-Authority Key Generation CP-ABE technique (PM-CPABE), which provided granular access control. This technique, in contrast to traditional techniques, can function in decentralized trust management systems without depending on a central authority with all the explanations. Policy masking more protects users' anonymity. The system supports a wide range of applications and allows for outsourced decryption as well. Thorough security assessments and performance associations support the usefulness and reliability of this method.

Li et al. [28], IntegrityChain is a blockchain-based decentralized storage system that provides verified data possession (PDP). We formalize a model for a system in which data owners can deposit funds with hosts in exchange for the safekeeping of their files. In contrast, hosts can earn money by providing secure storage and be punished by losing the deposit if data loss occurs. We investigate the practicality of the decentralized PDP and evaluate the security model's trade-off between a host and a data consumer. Combining multi-replica PDP with proof-of-retrievability, we assess the method's safety and provide a functional architecture. We develop a smart contract and deploy it on a test network to evaluate the gas cost for the functions. We conduct time-consuming local algorithm design to estimate off-chain consumption.

Limitations for existing system

- When dealing with a significant number of users and a massive volume of data, the system's scalability may be a concern. As the user base and data volume grow, the system's performance may remain the same, resulting in possible bottlenecks and slower response times.
- The shared data will be stored and managed by cloud service providers as part of the plan. However, ensuring CSP trustworthiness and reliability might be difficult. If the CSPs' security procedures are breached, there is a danger of unauthorized access or data breaches.
- Consequently, the scheme is meant to assimilate with an existing system, the underlying system's restrictions and constraints influence its effectiveness and functionality. If the present system has flaws or lacks

essential security process, it can impact the entire security and dependability of the selective sharing scheme.

- Encryption and cryptographic keys protect data in the secure data selective sharing system. Effective key management is essential for ensuring the security of shared data. However, securely distributing, preserving, and revoking keys can be difficult, especially when several users and data access levels are involved.
- While the plan seeks to allow selective data exchange, privacy concerns may still exist. Even within the system's specified sharing framework, users may need help accessing and using their data. It is critical to ensure adequate privacy measures and address user concerns to obtain user trust and acceptance.

Problem identification

- The current system needs a robust data security mechanism to protect the privacy, availability, and integrity of information. This is a significant concern, especially when handling private information.
- The existing system needs more selective sharing flexibility, making it easier for users to limit access to their data. There is a need for a more granular sharing system that enables users to establish and manage access privileges efficiently.
- Relying on a centralized trust management system raises issues about single points of failure and attack vulnerability. A more decentralized approach is required to improve the system's resilience and security.
- When dealing with many shared data and numerous users, the existing system may experience scalability challenges. The system's speed may decline as the user base and data size grow, affecting the overall user experience.
- Key management is critical for secure data sharing. However, ineffective essential management techniques in the current system may result in crucial compromise or loss. An enhanced key management strategy is required to protect the security and integrity of shared data.
- The system's usability may need to be improved, making it difficult for non-technical users to browse and use the selective sharing capabilities successfully. To improve the user interface and overall user experience, enhancements are required.

Proposed system

System model and threat model

As shown in Fig. 1, this study investigates the examination of four separate entities within a cloud-based system for selective data exchange and capture. Data owners, the cloud server, the users, and and an authority.

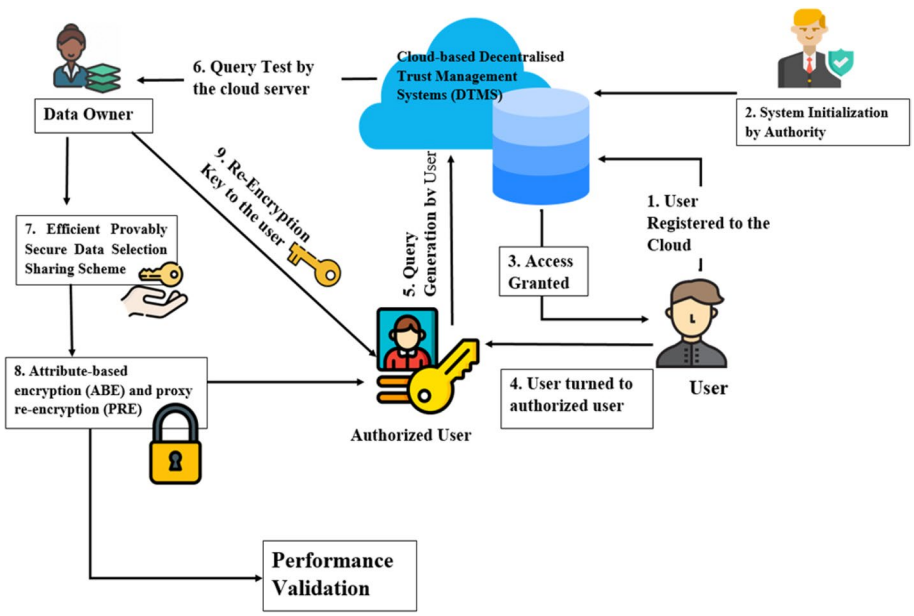

Fig. 1 Proposed method of EPSDSS-PRE

1) Data owners typically own and develop their own data. Data will be shared selectively, resulting in varying views for users with varied access. However, they lack trust in the cloud server's control over data sharing. To protect shared data, data owners create an access policy and encrypt it before transmitting it to the cloud. On the presumption that the data owners have complete trust for the system, they create indices for shared data.
2) Cloud Server: To establish whether the user's requirements can be satisfied, the cloud server assesses their qualities to check if they meet the access policy and the trapdoor. Once these conditions are met, the cloud server collaborates with the users to pre-decrypt the data using their newly updated secret keys and the provided trapdoor. The cloud server pledges to adhere to a data transmission protocol that accomplishes both goals by enabling pre-decryption. However, it also shows a bias in favour of the data providers' information.
3) Users: Each system member has a particular set of characteristics determining their place. The user builds a trapdoor that acts as a filter for the data and uses a modified secret key for pre-decryption to gain access to only the material that interests them. The cloud server receives the modified secret key and trapdoor in a single query. It is crucial to remember that users keep their private keys private, ensuring they are not disclosed to the cloud server or any prospective enemies. Users can work together on plans while having their personal information protected, thanks to this.
4) Authority: The authority represents the system's primary management. It is in charge of managing users' employment within the system by granting them various traits. According to the rights granted, it then gives each user a corresponding secret key. The public key has been made available, enabling index creation and data encryption. We take it for granted that every user has access to a secure channel for communication and that the system's authority can be trusted entirely. According to the study's premise, the power won't cooperate with the cloud server or another foe. By supervising and performing audits on specific agencies, the government or other public institutions can act as a trustworthy authority.

Design goals

In order to ensure efficiency and security in cloud-based systems, this study aims to create a reliable and efficient method for data collection and exchange. This will allow consumers to select and obtain pertinent data, and data owners to distribute their data selectively. The strategy seeks to satisfy the design requirements for efficiency, security, soundness, and correctness.

- Correctness: The user's characteristics must fit the access criteria in order for accurate data decryption to be possible, and the given trapdoor must match the defined index.
- Soundness: No non-interesting or undecryptable material should be sent to users.
- Security: The information must be kept secret to protect it from the cloud server and any unauthorized users. Users' interests should not be revealed through the index or the trapdoor.
- Efficiency: The approach should not impose excessive communication overhead or computation costs, particularly on users that access data using mobile devices with limited capabilities. Table 1 displays regularly used notations

Preliminaries

Efficient Provably Secure Data Selection Sharing Scheme (EPSDSS)

Definition of EPSDSS

The data selective sharing and gathering system is formally referred to as [29] to satisfy all the requirements above.

Table 1 Describes the main notation in the proposed method

Notation	Meaning
σ	Security parameter
A	Attribute universe
pk,*mk*	Public and master keys
sku	Secret key user
SK	Private key of DR
Ti	Block tag
DP, *TP*	Data proof and tag proof
AS	Access policy
mi	Data block
uid	Identity of user
usk, *psk*	Private and public key of user
TD	Trapdoor
msk	Master secret key
CT	Ciphertext
DO	The data owner
DU	The data user
ω	The attributes set of a specific DU

Definition 1 (DSs): These algorithms comprise a method for selectively sharing and acquiring data:

- **Setup**$(1^{\sigma}) \rightarrow (msk,\ pk)$. The safety features a component of the setup algorithm is σ.
- **SKGen** $(msk,\ pk,\ S_u) \rightarrow sk_u$. Each user's (u) set of characteristics S_u is fed into the algorithm alongside the master secret key msk, the public key and the public key itself pk. For each user (u), the procedure generates a secret key sk_u.
- **Encrypt** $(pk,\ (m,\omega),\ \mathrm{A}) \rightarrow (\mathrm{CT},\ \mathrm{I})$. Based on these features, the encryption mechanism is built to handle all inputs. There are two subroutines in the encryption method:
  - IndexGen $(pk,\ \omega) \rightarrow (I, M_I)$. Both the key data index M_I associated with the keyword w and a arbitrary index impression M_I are output by the function that generates the index. The keyword file list's entire contents will be encrypted with M_I.
  - DataEnc $(pk,\ m,\ \mathrm{A},\ M_I) \rightarrow \mathrm{CT}$. The information encryption algorithm generates the ciphertext CT given the random index stamp MM.
- **TDGen** $(Sk_u,\ S\hat{k}_u,\ pk,\ \omega) \rightarrow \mathrm{TD}$. Utilizing the trapdoor-generating technique with the decryption key $S\hat{k}_u$ as the input, it is possible to generate the related trapdoor TD.
- **Test** $(\mathrm{pk},\ (\mathrm{TD},\ \mathrm{I}),\ (tk_u,\ \mathrm{CT},\ \mathrm{A})) \rightarrow \hat{CT}$ or $\perp$. Inputs for the testing procedure include the data to be tested (ciphertext $\hat{CT}$), the public key pk, a set of trapdoors (TD and I), a distorted secret key tk_u, and the consistent access policy (A). It also has two test-related helper functions.
  - **HTest** $(pk,\ \mathrm{TD},\ \mathrm{I}) \rightarrow (\mathrm{D},\ \mathrm{P})$ or $\perp$. The test algorithm is stopped, and the program stops with the symbol if they don't match. If not, the function will obtain both the random element P, which contains the randomness of the index, and the random element D used for encryption.
  - **ATest** $(pk, tk_u, \mathrm{CT}, \mathrm{A}, \mathrm{D}, \mathrm{P}) \rightarrow \hat{CT}$ or $\perp$. The attribute test method uses random inputs (D, P). Its goal is to ascertain whether the initial access policy connected to the ciphertext is content by the qualities of the transmuted secret key tk_u. The test algorithm is terminated, and the program ends with the symbol if they don't match. If not, the previously decrypted ciphertext $\hat{CT}$ is generated.
- **Decrypt** $(S\hat{k}_u,\ \hat{CT}) \rightarrow \mathrm{m}$. The inputs of the decryption algorithm are the pre-decrypted ciphertext $\hat{CT}$ and the decryption key $S\hat{k}_u$. It outputs the data m.

Definition of correctness

The user can positively decrypt the data if their properties fit the related access policy and the provided trapdoor corresponds to the data index.

Definition 2 (Correctness): If AA and SS meet A, the data sharing and acquisition strategy DSS is accurate.

$$\Pr\left[Decrypt\left(s\hat{k}_u, Test(pk, (TD, I, (tk_u, CT, A)))\right) = m\right] = 1$$

Consideration is made of the probability factor when choosing.

$$(msk,\ pk) \leftarrow \mathrm{Setup}(1^{\sigma})$$
$$Sk_u \leftarrow \mathrm{SKGen}\ (msk,\ pk,\ S_u)$$
$$(\mathrm{CT},\ \mathrm{I}) \leftarrow \mathrm{Encrypt}\ (pk,\ (m,),\ \mathrm{A})$$
$$\left(\mathrm{TD},\ tk_u,\ S\hat{k}_u\right) \leftarrow \mathrm{Query}(sk_u,\ pk,\ \omega)$$

Definition of soundness

In the DSS, we assess keywords before policies. What would occur if a malicious cloud provided information directly to users who could decrypt it but didn't need it (keyword match) this is too much for security to handle. As a result, we define soundness using the three scenarios below.

- Case 1 (HTest Soundness): Failing the HTest becomes far more likely if the phrase in the trapdoor does not correspond to the term in the index.
- Case 2 (HTest Non-Bypassability): The cloud server shouldn't be able to pre-decrypt the user's data if possible, but the user has no intention of doing so.
- Case 3 (Decryption Dependability): Since the information already decrypted for each user is specific to them, it is doubtful that two users will ever be able to decode each other's data.

Definitions of security

In this paper, security is defined explicitly as a strategy involving selective data collection and sharing.

1. **Data Security:** In circumstances where the two defied plain texts share a single keyword that is unequal to both primary readers, we employ a more lenient variation of Index chosen plaintext attacks (IND-CPA) termed selective IND-CPA to ensure the integrity of the entire encryption method (*i.e.*, $(m_0, \omega), (m_1, \omega)$, *and* $\omega \neq m_0,\ \omega \neq m_1$).

Before "passive" eavesdropping may occur, data ciphertext must be pre-decrypted on the cloud server or transformed from its original form into a more easily decryptable form. Outsourcing ABE decryption is made

easier with the Index repayable chosen ciphertext attacks (IND-RCCA) method as well. To protect information encrypted with the same method, we detail the use of Selective IND-RCCA Security.

Definition 3 When challenger C interacts with an opponent whose life is a probabilistic polynomial in the security limitation, the resulting game is called the Selective-IND-RCCA-Game.

- **Init:** B grants D the experiment keyword ω^* and the experiment access policy B*.
- **Setup:** D runs System (1^σ) to make (msk, pk), and contributes pk to B.

Phase 1: D is initialized with an integer j = 0, a set T, and a set D. Any of the following questions can be adaptively asked by A.

Phase 2: If the outputs would be either m_0 or m_1, then D answers with a specific message test. Decrypt inquiries will be replied to in Phase 1 for Phase 1 queries. Phase 1 is repeated, but B can't ask a simple decryption question this time.

Guess: A generates a τ estimate.

2. Index Security: To defend the confidentiality of the indexed keywords, we necessitate that a challenger, absent disclosure of a relevant trapdoor, be unable to tell the difference between two indices constructed from keywords of identical length. We also want adaptive trapdoor searching, which necessitates semantically protected index chosen keyword attacks (IND-CKA). To define Selective IND-CKA security, we begin by choosing the challenge data in the first phase of the game.

   - **Init:** A hands over the m^* challenge information to D.
   - **Setup:** D runs the System (1^σ) procedure to create (msk, pk). It contributes pk to B
   - **Phase 1:** B may search all trapdoors for the ω_j keyword.
   - **Challenge:** Both ω_0 and ω_1 haven't been queried yet in Phase 1, so that's the sole limitation. To respond to B's index, I δ, D first flips an arbitrary coin δ and then executes the Encrypt method.
   - **Phase 2:** If the disputed terms are not challenged, everything in Phase 2 will remain the same.
   - **Guess:** B outputs a guess δ

3. **Trapdoor Security:** Unless a matching index is made public, an adversary should be unable to theoretically tell apart two trapdoors with identical-length keywords, as with index security.

An adversary can distinguish between the two challenge trapdoors with the help of an offline keyword-guessing attack. The public key is necessary for index building because our DSS uses encryption. According to the weaker security model for trapdoors presented in this article (it is difficult to discover the inner keyword when the trapdoor is given), the trapdoor's security needs only be one-way.

Definition 4 (Selective-IND-RCCA): For a DSS system to meet the security requirements of selective-IND-RCCA, adversaries' probabilistic polynomial-time attacks against the system must have a slight advantage.

Definition 5 Suppose a DSS strategy stops any opponent from winning the Selective-IND-CKA-Game with a probabilistic polynomial-time advantage by a margin more significant than a trivial one. In that case, it is said to be Selective-IND-CKA secure.

Definition 6 (DSS Security): It can be trusted if a DSS generates Selective-IND-CKA and Selective-IND-RCCA secure one-way trapdoors.

Remark 1: We presume that $\omega_0 \neq \omega_1$ and $m_0 \neq m_1$ while defining particular security. This security definition, according to us, already includes the scenario in which $\omega_0 = \omega_1$ or $m_0 = m_1$.

- (m_0, ω_1) IND (m_1, ω_1): Selective IND-CPA (where the attacker cannot obtain enough attributes) for data
- Then, we can say: (m_0, ω_0) IND (m_1, ω_1).

Construction of EPSDSS

System initialization by authority

The administrator starts the system's setup algorithm.

Using the data owner's properties, the specialist creates a secret key using the private essential generation technique.

SKGen $(msk, pk, S_u) \rightarrow sk_u$. It selects a random integer $r_{sub} \in Z_p^*$ and uses that as the secret key for each user u with the attribute set Su.

$$sk_u = \left((ac, bc),\ K = g^a g^{\beta u},\ K_a = g^{\frac{a}{a}} g^{\frac{\beta u}{a}},\ K_b = g^{\frac{a}{b}} g^{\frac{\beta u}{b}},\ K' = g^u,\ \forall x \in S_u :,\ K_x = H\left(x^u\right)\right) \quad (1)$$

Where u is arbitrarily selected from Z_p^*.

Data encryption by owners

Our architecture uses an LSSS structure called (M, p) to signify the access policy. The rows of M are qualities, and M is a $n \times 1$ contact conditions. The owner of the data then encrypts it using the subsequent procedure. We use the structure to convert the RCCA secure selected CPA secure CP-ABE.

Encrypt $(pk, (m, \omega), (M, p)) \rightarrow (CT, I)$. It is split into two different functions:

IndexGen $(pk, \omega) \rightarrow (I, R_I)$. An arbitrary text $R \leftarrow \{0,1\}^{\sigma}$ and two arbitrary amounts $s_t = s_1 + s_2$ are chosen and used to initialize the index creation subroutine's seed set $s_1, s_2 \in X_p^*$. The ciphertext for the input keyword w is then calculated as

$$c_{\omega} = e\left(H_0(\omega), g^{abc}\right)^{st}$$

The index is then displayed as

$$I = (I_1 = D \oplus H_1(c_{\omega}),\ I_2 = H_2(c_{\omega}))$$
$$L_1 = \left(g^b\right)^{s_1},\ L_2 = \left(g^a\right)^{s_2}$$
$$L_3 = g^{st}.H_0(\omega)^{st}$$

and the arbitrary index imprint $R_I = (R, s_t)$.

$$CT = \begin{pmatrix} C_0 = m \oplus H_4(k), C = k.e(g,g)^{as_a}, C' = g^{sa+st}.H_0(R), \\ \{C_i = g^{\beta\sigma_i}.H(p)(i)^{-r_i}, D_i = g^{ri}\} i \in [1,n] \end{pmatrix} \tag{2}$$

Where r1,,rn are arbitrarily selected in X_p^*. The information owner then updates the index and sends it to the $I||CT||(M,p)$ cloud server. Keep in mind that the (M,p) access policy is linked directly to the encrypted text.

Query generation by users

The users will use the algorithm below to produce a keyword query:

- SKTran $(sk_u) \rightarrow (tk_u, \hat{sk}_u)$. Transforming with the Key to the Secret The user's secret Key sk_u is turned into a transformed secret key via a subroutine picking a random number $z \in X_p^*$. pk as

$$TD = \begin{pmatrix} T_1 = g^{act1}H_0(\omega^*)^{ac+act1}, \\ \hat{T}_1 = (K_b)^z \left(gH_0(\omega^*)\right)^{act2} \\ T_2 = g^{bct1}H_0(\omega^*)^{bc+bct1} \\ \hat{T}_2 = (K_a)^z \left(gH_0(\omega^*)\right)^{bct2} \\ T_3 = \left(g^{abc}\right)^{t1}, \hat{T}_3 = \left(g^{abc}\right)^{t2} \end{pmatrix} \tag{3}$$

The query (TD, *tkj*) is stored along with the corresponding decryption key *tkj* and sent to a remote server.

Cloud server's test query

The cloud server starts the testing procedure once it receives the data and query to determine if the properties of the updated secret key match the access policy linked to the data ciphertext. Additionally, it confirms that the term in the trapdoor and the index match.

Test $(pk, (TD, I), tk_u, CT, \mathrm{X}) \rightarrow C\hat{T}$ or $\perp$. Keyword and attribute testing procedures are also a part of the overall testing methodology.

- KTest $(pk, (TD, I)) \rightarrow$ (R, Q) or $\perp$. The index I keyword and the trapdoor TD are used to find a match in the keyword test. To begin, we get the keyword's ciphertext in the form of

$$C_{\omega^*} = \frac{e(T_1, L_1).e(T_2, L_2)}{e(T_3, L_3)} \tag{4}$$

Next, it determines whether $H_2(C_{\omega^*}) = I_2$. Symbolically ending the test algorithm if not. If not, then CW = CW applies. A second component, Q comprising the unpredictable nature of the index, will also be returned by the function together with the arbitrary component R used for encryption as $R = I_1 \oplus H_1(C_{\omega^*})$.

$$Q = \frac{e\left(\hat{T}_1, L_1\right).e\left(\hat{T}_2, L_2\right)}{e\left(\hat{T}_3, L_3\right)} \tag{5}$$

$$tk_u = \left(\hat{K} = (K)^z = g^{az}g^{\beta uz}, \hat{K} = (K\prime)^z = g^{uz}, \forall x \in s_u : \hat{K}x = (K_x)^z = H(x)^{uz}\right)$$

- TDGen $(sk_u, \hat{sk}_u, pk, \omega) \rightarrow$ TD. The trapdoor TD is generated using the trapdoor generation procedure, which also accepts the decryption key $t_1, t_2 \in X_p^*$ as an input.

Check $(pk, tk_u, CT, \mathrm{X}, R, P) \rightarrow C\hat{T}$ *or* $\perp$. A customary of constants DDD, where $I = \{i : \rho(i) \in S_u\}$ is defined as $\sigma_i = M_i.\overrightarrow{v}$, can be found using the procedure, if nothing

else. Recall that $\sum_i d_i\lambda_i = s_a$ is available. A calculation follows this.

$$P = \prod_{i\in I}(e(D_i\widehat{K}').e(E_i,\widehat{H}_{\rho(i)})) \tag{6}$$

Where $\widehat{D}' = D'/H_0(R)$ the subroutine then does calculations.

$$\widehat{D} = \frac{e(\widehat{D}',\widehat{K}')}{P.Q} = e(g,g)\alpha gs_a \tag{7}$$

It then gives you back the cleartext in the form of

$$\widehat{DT} = (C_0 = m \oplus H_4(k),\ C = k.e(z,z)^{\alpha s_a} \tag{8}$$

$$\widehat{D} = e(z,z)^{\alpha gs_a}) \tag{9}$$

Data decryption by users

By rerunning the decryption method, the user can read encrypted material that has already been validated.

$Decrypt(\widehat{sk}_u, \widehat{CT}) \rightarrow m$. The ciphertext $\widehat{CT} = (S_0, D, \widehat{D})$ and the secret decryption key $\widehat{sk}_u$ are sent into the decryption process. At first, it calculates

$$H = \frac{D}{\widehat{D}^{\frac{1}{\widehat{sk}_u}}} = \frac{h.e(z,z)^{\alpha s_a}}{(e(z,z)^{\alpha gs_a})^{\frac{1}{g}}} \tag{10}$$

The data is then recovered.

$$m = D_0 \oplus K_4(H) \tag{11}$$

Now, it recomputes the $s' = H3(K,m)$ and checks whether the subsequent two equations can hold $D = e(z,z)^{\alpha g}$ and $\widehat{D} = e(z,z)^{\alpha gs_a}$. $D0$ represents predecrypted ciphertext. The data 'm' is recognized as it happens. The user does not care about the number of properties in the ciphertext, as they merely conduct simple decryption computations. The approach requires minimal processing resources, making it suitable for various mobile devices.

Correctness and soundness proofs

Correctness proof

Correctness of KTest: $d_{\omega*}$ can be calculated as in Eq. 4.

$$d_{\omega*} = e(K_0(\omega*)z)^{abds_t} = e(K_0(\omega),z)^{abds_t} = d_\omega \tag{12}$$

So, we have

$$K_0(d_{\omega*}) = K(d_\omega) \tag{13}$$

Correctness of ATest: Eq. 5's solution to Eq. 6 can be found if w = w.

$$\begin{aligned} P &= \prod_{i\in I}(e(E_i,\widehat{H}).e(E_i,\widehat{K}_{\rho(i)}))^{d_i} \\ &= \prod_{i\in I}(e(z^{\beta\sigma_i}.K(\rho(i)^{-r_i},z^{uz}).e(z^{ri},K(\rho(i))^{uz}))d_i \\ &= \prod_{i\in I} e(z,z)^{\beta uzd_i\sigma_i} \\ &= e(z,z)^{\beta uzs_q} \end{aligned} \tag{14}$$

Equation 7 can therefore be confirmed.

$$\begin{aligned} \widehat{D} &= \frac{e(\widehat{D}',\widehat{H})}{P.Q} \\ &= \frac{e(z^{sa}.z^{st},z^{\alpha z=g}.z^{\beta uz=g})}{e(z,z)^{\beta ugs_a},e(z,z)^{\alpha z=gs_t}.e(z,z)^{\beta uz=gs_t}} \\ &= \frac{e(z,z)^{\alpha gs_a}e(z,z)^{agst}.e(z,z)^{\beta ugs_a}e(z,z)^{\beta ugs_t}}{e(z,z)^{\beta ugs_a},e(z,z)^{\alpha gs_t}.e(z,z)^{\beta ugs_t}} \\ &= e(z,z)^{ags_a} \end{aligned} \tag{15}$$

Soundness proof

Therefore, only when $K_0(\omega*) \neq K_0(\omega), \omega* \neq \omega$, Q can be recovered to $e(z,z)^{\beta ugs_t}$. Let's check out the results of soundness case 3 by computing the component Q with the modified secret key of user u2. Keep in mind that the variables u and z, which are unique to each user, are tightly linked to P and Q. To decrypt ciphertext that has already been encrypted, the cloud server must employ a trapdoor and a secret key that has been updated using information belonging to a third user, user u2.

Attribute-based encryption (ABE) and proxy re-encryption

Definition 7 An ABPRE scheme is defined in definition one as a pair of probabilistic polynomial time algorithms (SETUP, KGEN, RKGN, EN, RENC, DC).

$SETUP\left(1^l\right) \rightarrow (qq, nl)$ In response to a given security parameter 1^l, the SETUP procedure of the system will return the master key *nl* and the system public limitation.

$RKGN(uk,S) \rightarrow (D)$: The algorithm for producing new keys RKGEN takes as inputs a secret key *uk* and an authorization structure S, and outputs a new key, rk.

$EN(S,n) \rightarrow (D)$: The encryption algorithm ENC generates the ciphertext D from the inputs of the contact structure S and the message m.

- $RENC(rk,D) \rightarrow \left(D'\right)$: The re-encryption method REENC first determines whether the index set in rk contents the entrance arrangement of C. This is done using the inputs of a re-key rk and a ciphertext D. The output is then "rejected" if the check fails or "re-encrypted ciphertext D' if it does.
- $DC(uk,D) \rightarrow (n)$: The decryption algorithm DEC first determines whether the index set in *uk* contents the entree structure of D when given a secret key *uk* and a ciphertext D as inputs. Then, if the check is successful, a message n is output in the message space; then, "reject" is output.

Correctness. There are two conditions for the correctness property. The following two equations must *hold*[2] for any communication m in the communication space.

1. $DC(KGEN(T, nl), EN(S, n)) = n$
2. $DC\big(KGEN(T'', nl), RENC\big(RKGN\big(KGEN\big(T', NL\big), S''\big), D\big)\big) = n$

Where T fulfils S, T' fulfils S', T'' fulfils S'', nl is a legitimate master key, D is the ciphertext associated with message n, and S is the access organization.

Plaintext security using selective structure selection

In the EPSDSS-PRE game, we argue that an ABPRE scheme demonstrates security against selected plaintext attacks if there is no probabilistic polynomial time adversary $\boldsymbol{\mathcal{A}}$ with a non-zero advantage. Init, the challenger, obtains a contest access organization S from the adversary $\boldsymbol{\mathcal{A}}$. The challenger does $SETUP(1^l)$ and provides $\boldsymbol{\mathcal{A}}$ with the organization public limitation pp that is returned. It holds onto the appropriate master-key nl.

Phase 1 The opponent $\boldsymbol{\mathcal{A}}$ asks the oracles the following questions:

- Key generation oracle P_{kg}: On input of an index set I_r, output a secret key $uk = KGEN(I_r, nl)$; otherwise, output "reject" if I_r does not fulfil S^*.
- The rekey generation oracle P_{rkg}: Given an index set I_r and an access organization AS, it outputs a rekey $rk = RKGN(KGEN(I_r, nl), S)$; otherwise, it outputs "reject". If I_r does not fulfil S^*, it outputs "reject".
- Re-encryption oracle P_{re}: If index set II does not fulfil access structure S^* but index set II does satisfy the access arrangement of ciphertext D, then output "reject"; else, output a ciphertext $D' = RENC(RKGN(KGEN(I_r, nl), S), D)$.

Challenge The adversary $\boldsymbol{\mathcal{A}}$ outputs two messages of equal length, n_o, n_1, from the message space after it determines that Phase 1 is complete. The challenger encrypts N with S^* using a random choice of $\mu \in \{0, 1\}$. The adversary $\boldsymbol{\mathcal{A}}$ is then given the ciphertext D.

Phase 2 similar to Phase 1.

Guess the game is won if the adversary $\boldsymbol{\mathcal{A}}$ s guess is correct, which is $\mu' \in \{0, 1\}$ and wins the game if $\mu' = \mu$

$$Adv^{P_{kg}, P_{rkg}, P_{re}}_{EPSDSS-PRE, \boldsymbol{\mathcal{A}}} = \left| \Pr e\left[\mu' - \mu\right] - \frac{1}{2} \right| \tag{16}$$

Satisfying on access structure

We only consider the "AND" gate-based access arrangement between this system's positive and negative features. The notation denotes the contact construction $\wedge + g_i\left(-g_i\right)^3 \mathbf{i} \in \boldsymbol{\mathcal{I}}$. The authority would give each user a secret key that corresponded to a set of attributes $S \subseteq \boldsymbol{\mathcal{I}}$. If and only if the following conditions are satisfied, the user successfully decrypts the ciphertext:

- If the access structure contains $+g_i$, then $i \in S$;
- If access structure contains $-g_i$, then $i \notin S$;

The algorithm generates these first public settings to set up the complete system. Users obtain the keys with any KGEN quality set S. The encryptor can design an access policy incorporating positive and negative characteristics through an AND gate. When the RKGN method is executed, a rk is related with the set T of qualities, generating a new access construction. The translation will only succeed if S matches the ciphertext's access structure. Users can also obtain a re-encrypted ciphertext through the RENC technique with the input of a valid ciphertext and a re-key. Repeatedly encrypted text can likewise be decrypted using the DC method.

- $SETUP\left(1^l\right)$: Create a bilinear collection C of prime instruction q with the bilinear map $e : C \times C \rightarrow C_U$ using the $SETUP\left(1^l\right)$ command. It then chooses elements x, $u_i (1 \leq i \leq 3m)$ in Z_q and two random producers c, h of C. For each $1 \leq i \leq 3m$, let $T_i := c^{u_i}$, and $T_{i'} := h^{u_i}$, and let $X := e(c, h)^x$. $\left\langle e, c, h, X, \{U_i, U_{i'}\}_{1 \leq i \leq 3m} \right\rangle$ are all part of the public parameter qq. The master key, abbreviated nl, is composed of the positive attribute, 3 + di, the negative attribute, $-g$, and the $\left\langle x, \{u_i\}_{1 \leq i \leq 3m} \right\rangle$.

KGEN$(\mathbf{T}, \mathbf{nl})$: Let T stand for an index set of characteristics. It sets $s = s_1 + s_2 + \ldots + s_m$ by randomly selecting $s_1, \ldots, s_m$ from Z_q. Calculate $\hat{G} = h^{x-s}$ and, for respectively $i \in \boldsymbol{\mathcal{N}} (\boldsymbol{\mathcal{N}} = \{1, 2, \ldots, m\})$, as follows: if $i \in T, G_{i,1} = h^{\frac{s_i}{u_i}}$, $G_{i,2} = h^{\frac{s_i}{u_{2m+i}}}$; otherwise, $G_{i,1} = h^{\frac{s_i}{u_{m+i}}}, G_{i,2} = h^{\frac{s_i}{u_{2m+i}}}$;. It produces the secret key of the user $uk = \left\langle T, (G_{i,1}, G_{i,2})_{i \in \boldsymbol{\mathcal{N}}}, \hat{G} \right\rangle$.

EN$(\mathbf{n}, \mathbf{S})$: Specify an access structure with S. It chooses random $t \in Z_q$ and computes $\tilde{D} = n.X^t$, $\hat{D} = c^t$, and $\breve{D} = h^t$. to encrypt a message m GT. For $i \in \boldsymbol{\mathcal{N}}$ if $+g_i$ appears S, then $D_i = U_i^t$; if $-g_i$ appears S, then $D_i = S_{m+i}^t$; otherwise, $D_i = U_{2m+i}^t$. It produces the value $D = \left\langle S, \tilde{D}, \hat{D}, \breve{D}, (D_i)_{i \in \boldsymbol{\mathcal{N}}} \right\rangle$.

RKGN$(\mathbf{uk}, \mathbf{S})$: Define uk as a legitimate secret key made up of $\left\langle T, (G_{i,1}, G_{i,2})_{i \in \boldsymbol{\mathcal{N}}}, \hat{G} \right\rangle$, and let S stand for an access structure. It chooses random $g \in Z_q$, sets $\mathfrak{D} = b^g$, and sets $\hat{G}' = \hat{G}$. For $i \in \boldsymbol{\mathcal{N}}$: if $i \in T, G'_{i,1} = G_{i,1}.(S'_i)^g, G'_{i,2} = G_{i,2}.(S'_{2m+i})^g$; otherwise, $G'_{i,1} = G_{i,1}.(S'_{m+i})^g, G'_{i,2} = G_{i,2}.(S'_{2m+i})^g$;. $\mathbb{C}$ is the ciphertext of $\mathfrak{D}$ under the access structure S.

$$\text{It produces the value } rk = \left\langle T, S\left(G'_{i,1}, G'_{i,2}\right)_{i\in\mathcal{N}}, \hat{G}', \mathbb{C}\right\rangle \quad (17)$$

RENC(rk,D): Let D signify a well-formed ciphertext $\left\langle S, \tilde{D}, \hat{D}, \breve{D}, (D_i)_{i\in\mathcal{N}}\right\rangle$, and let rk denote a valid re-key consisting of $\left\langle T, S', \left(G'_{i,1}, G'_{i,2}\right)_{i\in\mathcal{N}}, \hat{G}', \mathbb{C}\right\rangle$. It determines whether T meets S; if not, it outputs $\perp$; if not, for $i \in \mathcal{N}$:

$+g_i$ is present in S, $E_i = e\left(D_i, G'_{i,1}\right) = e\left(c^{u_i t}, h^{\frac{s_i+g}{u_i}}\right) = e(c,h)^{t(s_i+g)}$;

if $-g_i$ exists in S presents $E_i = e\left(D_i, G'_{i,1}\right) = e\left(c^{u_{m+i} t}, h^{\frac{s_i+g}{u_{m+i}}}\right) = e(c,h)^{t(s_i+g)}$;

otherwise, $E_i = e\left(D_i, G'_{i,2}\right) = e\left(c^{u_{2m+i} t}, h^{\frac{s_i+g}{u_{2m+i}}}\right) = e(c,h)^{t(s_i+g)}$;

Then, it calculates $\overline{D} = e\left(\hat{D}, \hat{G}'\right)\prod_{i\in\mathcal{N}} E_i = e\left(c^t, h^{x-\sum_{i=1}^{m} s_i}\right).$ $e(c,h)^{mgt+s\sum_{i=1}^{m} s_i} = e(c,h)^{xt+mgt}$. Output a re-encrypted ciphertext with $D' = \left\langle S', \tilde{D}, \overline{D}, \breve{D}, \mathbb{C}\right\rangle$.

With the number of re-encryptions, the ciphertext size also grows linearly.

DC(D, uk): Let uk stand for a legitimate secret key, such as $\left\langle T, (G_{i,1}, G_{i,2})_{i\in\mathcal{N}}, \widehat{G}\right\rangle$, It determines whether T meets S; if not, it outputs $\perp$; if not, it does

1. Assuming D is a valid ciphertext consisting of the characters: $\left\langle S', \tilde{D}, \widehat{D}, \breve{D}, (D_i)_{i\in\mathcal{N}}\right\rangle$, where $i \in \mathcal{N}$:
   - $+g_i$ appears in S, $E_i = e(D_i, G_{i,1}) = e\left(U_i^t, h^{\frac{s_i}{u_i}}\right) = e(c,h)^{ts_i}$;
   - if not, $-g_i E_i = e(D_i, G_{i,1}) = e(U_{m+i}^t, h^{\frac{s_i}{u_{m+i}}}) = e(c,h)^{ts_i}$;
   - it produces $E_i = e\left(D_i, G_{i,2}\right) = e\left(U_{2m+i}^t, h^{\frac{s_i}{u_{2m+i}}}\right) = e(c,h)^{ts_i}$;
2. Alternatively, if D is a re-encrypted well-formed ciphertext made up of $\left\langle S', \tilde{D}, \overline{D}, \breve{D}, \mathbb{C}\right\rangle$ it is decrypted using usk to get $\mathfrak{D} = b^g$. The result is thus $\frac{\overline{D}}{e\left(\widehat{D}, \widehat{G}\right)\cdot\prod_{i\in\mathcal{N}} E_i} = \frac{n.e(c,h)^{xt}}{e(c^t, h^{x-s}).e(c,h)^{ts}} = n$
3. Alternatively, if D is a well-formed ciphertext that has been encrypted several times. The above phases apply to the decryption.

ABPRE security justification

Theorem: The ABPRE method ensures selectivity in the chosen plaintext if the ADBDH assumption holds in $(C,\ C_U)$.

Proof. Let's suppose that SS-CPA security for ABPRE can be guaranteed under the enhanced decisional Bilinear Diffie-Hellman assumption.

In the SS-CPA-ABPRE game, suppose that your opponent $\mathcal{A}$ has a significant advantage and wins. In order to find $\mathfrak{D}_{adbdh}$ from $\mathfrak{D}_{rand}$ that isn't completely useless. A simulator S can be constructed using $\frac{\varepsilon}{2}$.

We initially allow the challenger to create the groups C and C_u using a generator g and an effective bilinear map e. Outside of S's line of sight, the contestant flips a fair binary coin w. The contestant groups $\langle c, B, A, D, A', Z\rangle \in \mathfrak{D}_{adbdh}$ if $w = 1$, and $\langle c, B, A, D, A', Z\rangle \in \mathfrak{D}_{rand}$ otherwise.

Init During this step, $\mathcal{S}$ is given a challenging access structure, and they make separate notes for the index set of positive and negative qualities, I_+^*, I_-^*. After that, S randomly chooses l, α_i, β_i, γ_i from Z_q for $i \in \mathcal{N}$ and produces the public key $X = e(B,A)^l$, $h = c^l$. The public parameters are output by $\mathcal{S}$ after that.

- $i \in I_+^*$, $U_i = c^{\alpha_i}, U_{m+i} = A^{\beta_i}, U_{2m+i} = A^{\gamma_i}, U'_i = c^{\frac{l}{\alpha_i}}$, $U'_{m+i} = A^{\frac{l}{\beta_i}}, U'_{2m+i} = A^{\frac{l}{\gamma_i}}$;
- $i \in I_-^*$, $U_i = A^{\alpha_i}, U_{m+i} = c^{\beta_i}, U_{2m+i} = A^{\gamma_i}, U'_i = A^{\frac{l}{\alpha_i}}$, $U'_{m+i} = c^{\frac{l}{\beta_i}}, U'_{2m+i} = A^{\frac{l}{\gamma_i}}$;
- *otherwise*, $U_i = A^{\alpha_i}, U_{m+i} = A^{\beta_i}, U_{2m+i} = c^{\gamma_i}, U'_i = A^{\frac{l}{\alpha_i}}$, $U'_{m+i} = A^{\frac{l}{\beta_i}}, U'_{2m+i} = c^{\frac{l}{\gamma_i}}$;

Phase 1: The key generation oracle P_{kg}, the re-key generation oracle P_{rkg}, and the encryption oracle P_{renc} are all queried by $\mathcal{A}$ in phase one.

- $\mathcal{A}$ queries Okg using the index set I_r. The security game states that if I_r satisfies AS, it produces. In the absence of this, $\mathcal{S}$ outputs uk after querying the oracle in Appendix B5 at $uk = \left\langle I_r, (G_{i,1}, G_{i,2})_{i\in\mathcal{N}}, \hat{G}\right\rangle$
- Using an access structure S^* and an index set I_r, $\mathcal{A}$ queries P_{rkg}. According to the security game, if condition I_r holds, then $\perp$ is generated. If not, S^* gives P_{kg} and receives a secret key $uk = \left\langle I_r, (G_{i,1}, G_{i,2})_{i\in\mathcal{N}}, \hat{G}\right\rangle$. $\mathcal{S}$ Performs the subsequent actions:
  - Set $\mathfrak{D} = c^g$ and $\hat{G}' = \hat{G}$; by selecting $g \in Z_q$ at arbitrary;
  - for $i \in \mathcal{N}$
    - $i \in I_r$, $G'_{i,1} = G_{i,1}.(U'_i)^g, G'_{i,2} = G_{i,2}.(U'_{2m+i})^g$;
    - *Otherwise* $G'_{i,1} = G_{i,1}.(U'_{n+i})^g, G'_{i,2} = G_{i,2}.(U'_{2m+i})^g$;

4. $rk = \left\langle I_r, S, \left(G'_{i,1}, G'_{i,2}\right)_{i\in\mathcal{N}}, \hat{G}', \mathbb{C} \right\rangle$, where $\mathbb{C}$ is the access structure AS-compliant ciphertext of $\mathfrak{D}$.
5. Including an access structure S', a ciphertext $D = \left\langle S, \tilde{D}, \widehat{D}, \breve{D}, (D_i)_{i\in\mathcal{N}} \right\rangle$, and an index set I_r. A request is sent from $\mathcal{A}$ to P_{renc}. If condition (II) meets condition (S) in the security game, then I_r produces. I_r outputs $\perp$ if S is not satisfied. S' then sends (I_r, S') to the re-key generation oracle, where he receives.

$$rk = \left\langle I_r, S', \left(G'_{i,1}, G'_{i,2}\right)_{i\in\mathcal{N}}, \hat{G}', \mathbb{C} \right\rangle$$

The ciphertexts D are re-encrypted by $\mathcal{S}$ using rk. For $i \in \mathcal{N}$:

$$-\ \text{If} + g_i \text{ performs in } S,\ E_i = e\left(D_i, G'_{i,1}\right) = e\left(c^{u_i t}, h^{\frac{s_i+g}{u_i}}\right) = e(c,h)^{t(s_i+g)};$$
$$-\ \text{If} - g_i \text{ appears in } S,\ E_i = e(D_i, G'_{i,1}) = e(c^{u_{m+i} t}, h^{\frac{s_i+g}{u_{m+i}}}) = e(c,h)^{t(s_i+g)};$$
$$\textit{otherwise}\ \ E_i = e(D_i, G'_{i,1}) = e(c^{u_{2m+i} t}, h^{\frac{s_i+g}{u_{2m+i}}}) = e(c,h)^{t(s_i+g)};$$

It then calculates $\overline{D} = e\left(\widehat{D}, \widehat{G}'\right) \prod_{i\in\mathcal{N}} E_i = e(c^t, h^{x-\sum_{i=1}^{m} s_i})$. $e(c,h)^{mgt+t\sum_{i=1}^{m} s_i} = e(c,h)^{xt+mgt}$ It then outputs the ciphertext that has been re-encrypted.

$$D_{recn} = \left\langle S', \tilde{D}, \overline{D}, \breve{D}, \mathbb{C} \right\rangle \tag{18}$$

Result and discussion

Experimental setup

The Pairing-Based Cryptography Library (PBC) is used to test the effectiveness of this strategy. A server with 32 GB of RAM and a 3.60 GHz Intel Core i9-9900KF CPU can be used to imitate cloud service providers' activities. The file used in this experiment is split up into 1000, 2000, 3000, 4000, and 5000 data blocks, with a 1 MB file size for each data block. According to the testing results, this scheme's tag production speed is comparable to that of the ID-MRPDP, MDSS, CP-ABE, and MA-ABE schemes. The more data blocks that need to be tagged, the longer it takes to generate tags. Compared to the previous two methods, this approach generates less hash information but returns tags faster. This system is safer and more effective than the ID-MRPDP, MDSS, CP-ABE, and MA-ABE schemes, according to thorough study and experimental data.

Performance validation

In this section, we evaluate our method in comparison to several multi-copy integrity auditing techniques. For example, there is CP-ABE, which is a certificateless multi-copy integrity auditing scheme that accounts for data dynamics; ID-MRPDP, which is an identity-based provable multi-copy data possession scheme designed for use with multiple clouds; and MDSS, which is another certificateless multi-copy integrity auditing scheme, as mentioned in reference [30]. First exhibited are functionality comparisons between our system and the precedent designs. Then, several entities' time and communication costs are illustrated. The time cost of distinct entities is then demonstrated through experiments. We also compare the overall scheme and our approach's capacity to recover data [31, 32].

Accuracy analysis

A comparison of the EPSDSS-PRE strategy's accuracy to various existing methods is shown in Fig. 2 and Table 2. The graph illustrates how the deep learning approach has an improved efficiency with maximum accuracy. In contrast to the accuracy values of 89.04%, 76.22%, 82.19%, and 85.14% for the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, the EPSDSS-PRE model has an accuracy of 96.53% for 1000 data blocks. However, the EPSDSS-PRE model has performed better with various data sizes. The EPSDSS-PRE model has an accuracy of 99.15% under 5000 data blocks, compared to the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, which have accuracy values of 92.38%, 79.87%, 85.14%, and 87.68%, respectively.

Encryption time analysis

In Table 3 and Fig. 3, the encryption time of the proposed EPSDSS-PRE methodology is compared to that of existing techniques. The data clearly shows that the EPSDSS-PRE technique outperformed all the other strategies. The suggested EPSDSS-PRE approach, for example, took only 231.04 ms to encrypt with 1000 data block, whereas other current methods such as ID-MRPDP, MDSS, CP-ABE, and MA-ABE have taken 476.75 ms, 789.31 ms, 954.32 ms, and 514.98 ms, respectively. Similarly, the suggested EPSDSS-PRE approach takes 311.84 ms to encrypt 5000 data blocks, while existing techniques like ID-MRPDP, MDSS, CP-ABE, and MA-ABE take 536.29 ms, 867.18 ms, 1289.74 ms, and 628.76 ms, respectively as their encryption time.

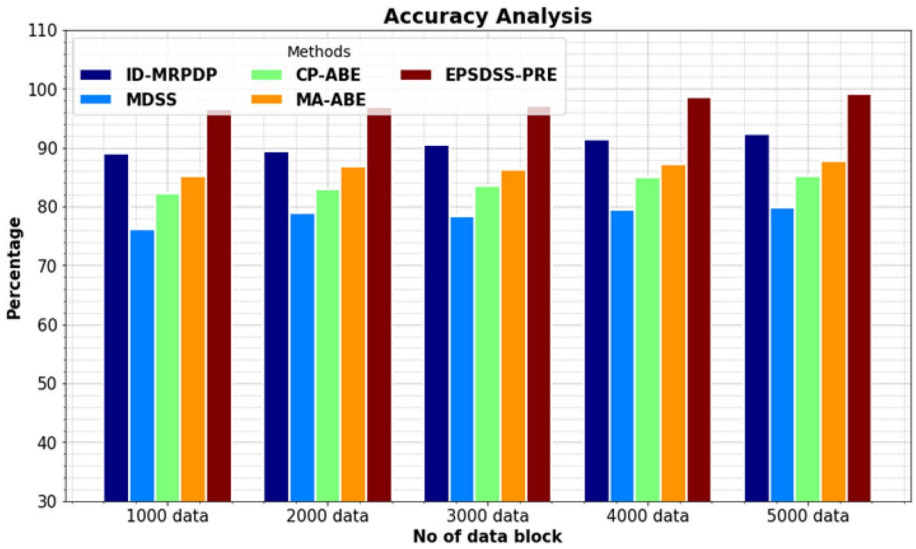

Fig. 2 Accuracy analysis for EPSDSS –PRE method with existing systems

Table 2 Accuracy analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	89.04	76.22	82.19	85.14	96.53
2000	89.32	78.94	82.95	86.73	96.89
3000	90.51	78.41	83.42	86.29	97.14
4000	91.43	79.43	84.95	87.13	98.62
5000	92.38	79.87	85.14	87.68	99.15

Table 3 Encryption time analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	476.75	789.31	954.32	514.98	231.04
2000	498.51	801.88	1054.21	538.14	245.19
3000	521.05	832.14	1099.32	567.93	278.19
4000	482.06	841.05	1210.84	598.12	295.43
5000	536.29	867.18	1289.74	628.76	311.84

Decryption time analysis

In Table 4 and Fig. 4, the Decryption time of the proposed EPSDSS-PRE methodology is compared to that of existing techniques. The data clearly shows that the EPSDSS-PRE technique outperformed all the other strategies. The suggested EPSDSS-PRE approach, for example, took only 209.14 ms to decrypt with 1000 data blocks, whereas other current methods such as ID-MRPDP, MDSS, CP-ABE, and MA-ABE have taken 812.78 ms, 1067.21 ms, 946.17 ms, and 549.78 ms, respectively. Similarly, the suggested EPSDSS-PRE approach takes 376.14 ms to decrypt 5000 data blocks, while existing techniques like ID-MRPDP, MDSS, CP-ABE, and MA-ABE take 915.19 ms, 1183.92 ms, 1074.31 ms, and 685.02 ms, respectively as their decryption time.

Key generation time analysis

In Table 5 and Fig. 5, the key generation time of the proposed EPSDSS-PRE methodology is compared to that of existing techniques. The data clearly shows that the EPSDSS-PRE technique outperformed all the other strategies. The suggested EPSDSS-PRE approach, for example, took only 104.98 ms to generate key for 1000 data blocks, whereas other current methods such as ID-MRPDP, MDSS, CP-ABE, and MA-ABE have taken 239.18 ms, 321.04 ms, 413.32 ms, and 564.81 ms, respectively. Similarly, the suggested EPSDSS-PRE approach takes 210.34 ms to generate key for 5000 data blocks, while existing techniques like ID-MRPDP, MDSS, CP-ABE, and MA-ABE take 283.56 ms, 372.85 ms, 491.58 ms, and 593.76 ms, respectively to generate keys.

Memory usage

A comparison of the EPSDSS-PRE strategy's memory usage to various existing methods is shown in Fig. 6 and Table 6. The graph illustrates how the deep learning approach has an improved efficiency with low

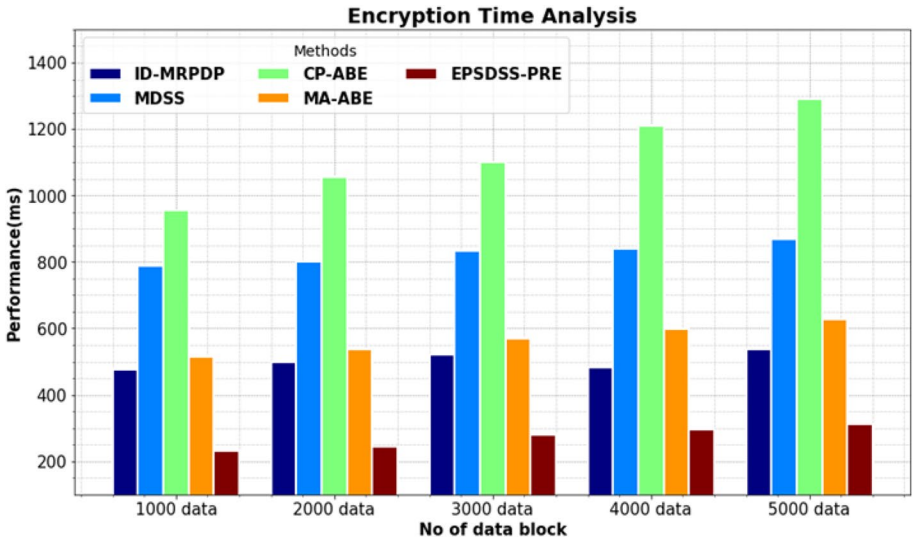

Fig. 3 Encryption time analysis for EPSDSS –PRE method with existing systems

Table 4 Decryption time analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	812.78	1067.21	946.17	549.78	209.14
2000	843.92	1103.84	963.32	593.14	243.85
3000	876.81	1094.78	994.93	615.05	285.43
4000	891.04	1123.83	1027.63	643.84	301.43
5000	915.19	1183.92	1074.31	685.02	376.14

memory usage. In contrast to the memory usage values of 73.198%, 48.127%, 61.293%, and 58.310% for the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, the EPSDSS-PRE model has a memory usage of 32.190% for 1000 data blocks. However, the EPSDSS-PRE model performed better with various data sizes. The EPSDSS-PRE model has a memory usage of 36.531% under 5000 data blocks, compared to the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, which have memory usage values of 78.215%, 54.927%, 66.318%, and 65.175%, respectively.

Data transfer rate analysis

A comparison of the EPSDSS-PRE strategy's data transfer rate to various existing methods is shown in Fig. 7

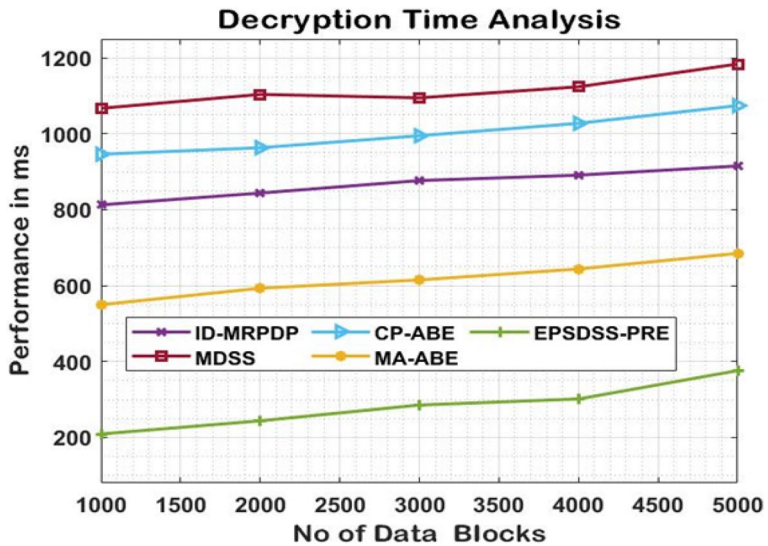

Fig. 4 Decryption time analysis for EPSDSS –PRE method with existing systems

Table 5 Key generation time analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	239.18	321.04	413.32	564.81	104.98
2000	263.47	374.28	473.85	573.23	121.31
3000	254.59	351.73	463.92	543.19	154.82
4000	271.06	369.31	483.17	563.93	193.52
5000	283.56	372.85	491.58	593.76	210.34

and Table 7. The graph illustrates how the deep learning approach has an improved efficiency with high data transfer rate. In contrast to the data transfer rate values of 59.42%, 73.95%, 65.74%, and 80.21% for the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, the EPSDSS-PRE model has a data transfer rate of 93.12% for 1000 data blocks. However, the EPSDSS-PRE model performed better with various data sizes. The EPSDSS-PRE model has a data transfer rate of 97.32% under 5000 data blocks, compared to the ID-MRPDP, MDSS, CP-ABE, and MA-ABE

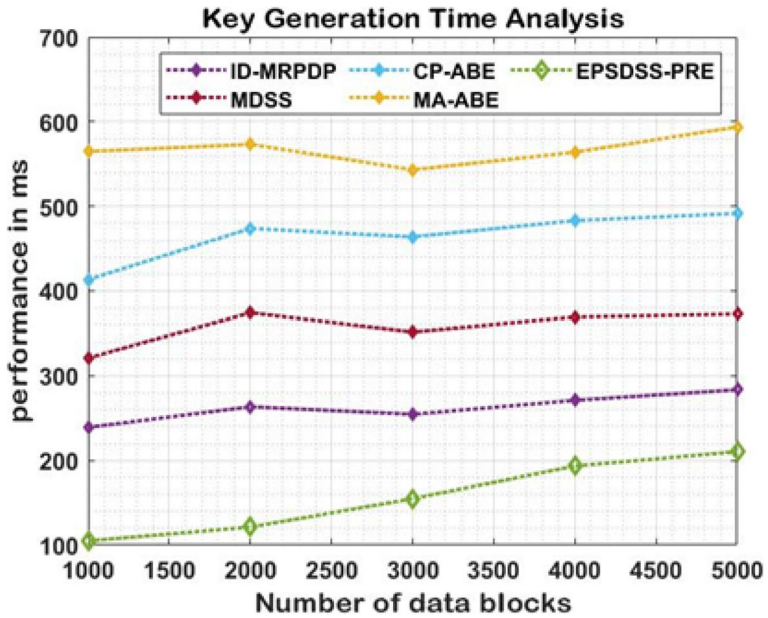

Fig. 5 Key generation time analysis for EPSDSS –PRE method with existing systems

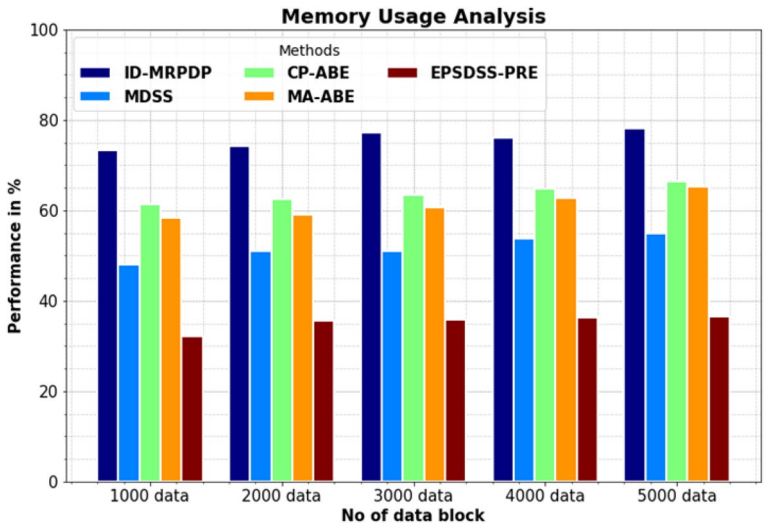

Fig. 6 Memory usage analysis for EPSDSS –PRE method with existing systems

Table 6 Memory usage analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	73.198	48.127	61.293	58.310	32.190
2000	74.274	50.943	62.429	59.148	35.519
3000	77.219	51.084	63.492	60.732	35.854
4000	76.043	53.741	64.841	62.843	36.273
5000	78.215	54.927	66.318	65.175	36.531

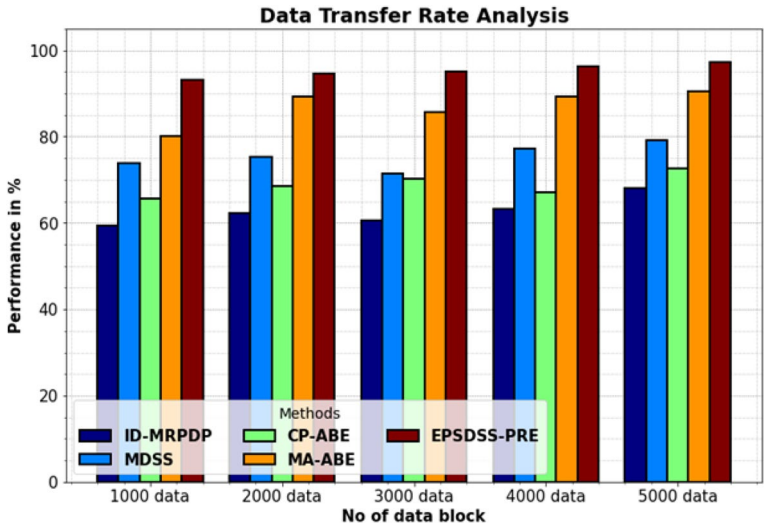

Fig. 7 Data transfer rate analysis for EPSDSS –PRE method with existing systems

Table 7 Data transfer rate analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	59.42	73.95	65.74	80.21	93.12
2000	62.35	75.28	68.43	89.32	94.53
3000	60.59	71.42	70.32	85.64	95.15
4000	63.14	77.23	67.18	89.21	96.21
5000	67.94	79.04	72.54	90.43	97.32

models, which have data transfer rate values of 67.94%, 79.04%, 72.54%, and 90.43%, respectively.

Error rate analysis

A comparison of the EPSDSS-PRE strategy's error rate to various existing methods is shown in Fig. 8 and Table 8. The graph illustrates how the deep learning approach has an improved efficiency with minimum error rate. In contrast to the error values of 43.902%, 76.392%, 63.184%, and 59.143% for the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, the EPSDSS-PRE model has a minimum error rate of 24.194% for 1000 data blocks. However, the EPSDSS-PRE model performed better with various data sizes. The EPSDSS-PRE model has an error rate of 29.432% under 5000 data blocks, compared to the ID-MRPDP, MDSS, CP-ABE, and MA-ABE models, which have an error rate of 52.178%, 80.483%, 69.483%, and 64.392%, respectively.

Conclusion

In conclusion, when integrated with cloud-based decentralized trust management systems, the suggested data selective sharing scheme offers a practical and verifiably secure method for protecting sensitive data.

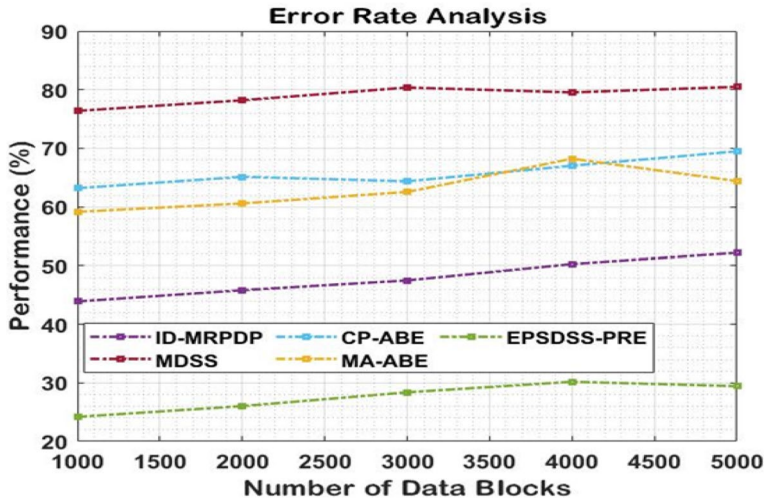

Fig. 8 Error rate analysis for EPSDSS –PRE method with existing systems

Table 8 Error rate analysis for EPSDSS –PRE method with existing systems

No of data block	ID-MRPDP	MDSS	CP-ABE	MA-ABE	EPSDSS-PRE
1000	43.902	76.392	63.184	59.143	24.194
2000	45.821	78.194	59.128	60.583	26.032
3000	47.491	80.372	64.363	62.549	28.384
4000	50.273	79.541	67.027	68.216	30.185
5000	52.178	80.483	69.483	64.392	29.432

By allowing users to share data while controlling access permissions selectively, the project addresses privacy concerns and provides robust protection against unauthorized disclosure. Leveraging the power of cloud computing and decentralized trust management, the scheme offers scalability and reliability while ensuring the integrity and confidentiality of shared data. Overall, this research contributes to advancing certain data-sharing practices in cloud environments, paving the way for enhanced privacy and data protection in the digital age. This study created a cloud-based, decentralized trust management system called the Efficient Provably Secure Data Selection Sharing Scheme (EPSDSS). The suggested EPSDSS solution uses sophisticated cryptographic methods, including attribute-based encryption (ABE) and proxy re-encryption (PRE), to establish fine-grained access control over shared data. A decentralized trust management system ensures the dependability and responsibility of participating entities, hence mitigating the risks associated with centralized trust models. The proposed EPSDSS-PRE system would allow data owners to regulate granular access to their data while allowing users to modify data gathering without identifying their preferences. The EPSDSS recognizes shared data and creates short fingerprints for data that can successfully avoid detection before cloud storage in our scheme. Furthermore, the DTMS analyses user trustworthiness and improves user behaviour administration based on this computation. Using data merging, we lower the overall data selection and the stress on users and the cloud. Our technology is secure enough to survive forgery assaults from three different attackers. Future work could focus on further enhancing the performance of the Efficient Provably Secure Data Selective Sharing (EP-SDSS) scheme. This could involve exploring new cryptographic techniques, data structures, or algorithms to improve the scheme's efficiency, such as reducing computational overhead or communication costs.

Authors' contributions

Conceptualization, S.V. and M.P.; methodology, S.N..; software, A.R. and S.V.; validation, S.N., S.V. and A.R.; formal analysis, S.N.; investigation, S.V..; resources, S.V.; data curation, S.N.; writing—original draft preparation, M.P.; writing—review and editing, S.N., S.V.; visualization, M.P.; supervision, S.N.; project administration, S.V.; All authors have read and agreed to the published version of the manuscript.

Funding
The authors received no specific funding for this Research.

Availability of data and materials
The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Ethics approval and consent to participate
Not Applicable.

Competing interests
The authors declare no competing interests.

Author details
[1]Department of Information Technology, R.M.D. Engineering College, Kavaraipettai, Tamil Nadu, India. [2]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. [3]Department of Computer Science and Engineering, RMK Engineering College, Chennai, India. [4]Faculty of Electrical & Computer Engineering, Jimma Institute of Technology, Jimma University, Jimma, Ethiopia.

Received: 8 September 2023 Accepted: 13 March 2024
Published online: 10 April 2024

References

1. Xu M, Buyya R (2019) Brownout approach for adaptive management of resources and applications in cloud computing systems. ACM Comput Surv 52(1):1–27
2. Zhu Y, Zhang W, Chen Y, Gao H (2019) A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment. EURASIP J Wirel Commun Netw 247:1–18. https://doi.org/10.1186/s13638-019-1605-z
3. Abdulsalam Y, Shailendra S, Shamim H, Ghulam M (2019) IoT big data analytics for smart homes with fog and cloud computing. Future Gener Comput Syst 91:563–573
4. Zewei L, Chunqiang H, Ruinian L, Tao X, Jiguo Y, Hui X (2023) A privacy-preserving outsourcing computing scheme based on secure trusted environment. IEEE Trans Cloud Comput 11(3):2325–2336
5. Sheng C, Gexiang Z, Pengfei L, Xia Song Z, Ferrante N (2019) Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain. Inf Sci 485:427–440
6. Kuan F, Zijian B, Mingxi L, Vasilakos AV, Wenbo S (2020) Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial IoT. Futur Gener Comput Syst 110(1):665–674
7. Goswami P, Faujdar N, Debnath S, Khan A, Singh G (2024) Investigation on storage level data integrity strategies in cloud computing: classification, security obstructions, challenges and vulnerability. J Cloud Comput 13:45
8. Bian G, Fu Y, Shao B, Zhang F (2022) Data integrity audit based on data blinding for cloud and fog environment. IEEE Access 10:39743–39751
9. Yan L, Ge L, Wang Z, Zhang G, Xu J, Hu Z (2023) Access control scheme based on blockchain and attribute-based searchable encryption in cloud environment. J Cloud Comput 12(1):61
10. Zhang X, Wang X, Gu D, Xue J, Tang W (2022) Conditional anonymous certificateless public auditing scheme supporting data dynamics for cloud storage systems. IEEE Trans Netw Serv Manage 19(4):5333–5347
11. Sermakani AM (2020) Effective data storage and dynamic data auditing scheme for providing distributed services in federated cloud. J Circuits Syst Comput 29(16):205–259
12. Jouini M, Rabai L (2019) A security framework for secure cloud computing environments. In: Cloud security: concepts, methodologies, tools, and applications. p 249–263
13. Kan Y, Xiaohua J, Kui R, Ruitao X (2014) Enabling efficient access control with dynamic policy updating for big data in the cloud in Proc. INFOCOM: 2013–2021
14. Zhu N, Cai F, He J, Zhang Y, Li W, Li Z (2019) Management of access privileges for dynamic access control. Clust Comput 22(4):8899–8917
15. Wang P, Yue Y, Sun W, Liu J (2019) An attribute-based distributed access control for blockchain-enabled IoT. In: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, Barcelona, pp 1–6
16. Liu B, Xiao L, Long J, Tang M, Hosam O (2020) Secure digital certificate-based data access control scheme in blockchain. IEEE Access 8:91751–99176
17. Yu J, Zhang H, Li S, Mao L, Ji P (2019) Data sharing model for internet of things based on blockchain. J Chin Mini-Micro Comput Syst 40(11):2324–2329
18. Xiang G, Jia Y, Yan C, Huaqun W, Jianxi F (2021) Checking only when it is necessary: enabling integrity auditing based on the keyword with sensitive information privacy for encrypted cloud data. IEEE Trans Dependable Secure Comput 19(6):3774–3789
19. Fan K, Pan Q, Zhang K, Bai Y, Sun S, Li H, Yang Y (2020) A secure and verifiable data sharing scheme based on blockchain in vehicular social networks. IEEE Trans Veh Technol 69(6):5826–5835
20. Yang C, Tan L, Shi N et al (2020) AuthPrivacyChain: a blockchain-based access control framework with privacy protection in cloud. IEEE Access 8:70604–70615
21. Wang C, Cheng X, Li J, He Y, Xiao K (2021) A survey: applications of blockchain in the Internet of vehicles. EURASIP J Wirel Commun Netw 77:1–16
22. Tian Y, Haowen T, Jian S, Pandi V, Brij BG, Varsha A (2023) Efficient identity-based multi-copy data sharing auditing scheme with decentralized trust management. Inf Sci 644:119255
23. Gupta R, Kanungo P, Dagdee N, Madhu G, Sahoo KS, Jhanjhi NZ, Masud M, Almalki NS, AlZain MA (2023) Secured and privacy-preserving multi-authority access control system for cloud-based healthcare data sharing. Sensors 23(5):2617
24. Li W, Wu J, Cao J, Chen N, Zhang Q, Buyya R (2021) Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions. J Cloud Comput 10(1):1–34
25. Yan X, Ni H, Liu Y, Han D (2019) Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR. Comput Sci Inf Syst 16:831–847
26. Najafi A, Bayat M, Haj Javadi H (2021) Privacy preserving attribute-based encryption with conjunctive keyword search for e-health records in cloud. ISC Int J Inf Secur 13:87–100
27. Ruan C, Hu C, Zhao R, Liu Z, Huang H, Yu J (2023) A policy-hiding attribute-based access control scheme in decentralized trust management. IEEE Internet Things J 10(20):17656–17665
28. Li Y, Yong Y, Ruonan C, Xiaojiang D, Mohsen G (2020) IntegrityChain: provable data possession for decentralized storage. IEEE J Sel Areas Commun 38(6):1205–1217
29. Yang K, Shu J, Xie R (2022) Efficient and provably secure data selective sharing and acquisition in cloud-based systems. IEEE Trans Inf Forensics Secur 18:71–84
30. Lei Z, Anmin F, Guomin Y, Huaqun W, Yuqingg Z (2022) Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics. IEEE Trans Depend Secure Comput 19(2):1118–1132
31. Paulraj D, Neelakandan S, Prakash M, Baburaj E (2023) Admission control policy and key agreement based on anonymous identity in cloud computing. J Cloud Comput 12(1):1–18
32. Paulraj D, Sethukarasi T, Neelakandan S, Prakash M, Baburaj E (2013) An Efficient Hybrid Job Scheduling Optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment. PLoS One 18(3):e0282600

Publisher's Note