International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 05 | May 2024 www.irjet.net p-ISSN: 2395-0072

BLOCKCHAIN BASED DOUBLE LAYER DATA PROTECTION WITH ENHANCED ACCESS CONTROL MECHANISM

RAMYA R[1], JAYANTHI K[2]

[1]M.TECH-Student, DEPARTMENT OF CSE, PRIST UNIVERSITY, THANJAVUR, TAMILNADU, INDIA.
[2]M.E-Assistant Professor, DEPARTMENT OF CSE, PRIST UNIVERSITY, THANJAVUR,TAMILNADU,INDIA.

Abstract— an expanded version of cloud computing is called fog computing. A current research trend is fog computing, which offers the potential for effective network services. Fog computing is a decentralized platform that is fundamentally different from cloud computing, while it shares many similarities with cloud computing in terms of data computation, storage, and application services. The security and privacy of medical records are crucial in the age of digital healthcare. The goal of this project is to create and put into place a complete system for the safekeeping and restricted access of medical records. The fog server's data processing and storage capacity are fully utilized by the TLS (Three Layer Security) framework. Three layers make up the architecture: the local machine, the fog server, and the cloud server. The solution creates a strong foundation to protect sensitive healthcare data by utilizing cloud storage for scalability, a fog computing layer for real-time processing, and Advanced Encryption Standard (AES) encryption for data secrecy. It also uses blockchain technology to protect the integrity of the data. In order to control user and data owner permissions and guarantee authorized access, a trusted authority system is implemented. In accordance with the least privilege concept, role-based access control (RBAC) procedures are used to limit data access based on user roles. Users can access certain medical records through a secure file request and verification process that follows strict authentication guidelines. In addition to addressing the present security issues, the suggested solution lays the groundwork for future adaptability to new threats in the ever-changing field of healthcare information management.

Index Terms — Cloud server, Fog server, data storage, Index creation, Blockchain process, Role based access control, Data request, Verification process, Data sharing, User revocation. **Introduction**

FOG COMPUTING

A dispersed computing infrastructure known as fog computing places data, processing power, storage, and applications in between the cloud and the data source. Fog computing, like edge computing, brings the benefits and strength of the cloud closer to the point of creation and action for data. Since both refer to moving processing and intelligence closer to the point of data creation, many people confuse the terms edge computing and fog computing. Though it may also be done for security and compliance purposes, efficiency improvement is the main motivation for this.

Just as fog focuses on the network's edge, the fog metaphor originates from the meteorological term for a cloud near the ground. The phrase is frequently linked to Cisco; it is thought to have been created by Ginny Nichols, the product line manager for the corporation. Fog computing is available to the general public and is registered under the term Cisco Fog Computing.

CLOUD COMPUTING

Undoubtedly, the cloud computing concept holds great potential for business computing. It highlights the need for crucial infrastructure to support a new kind of service delivery that offers the advantage of cost savings through pooling computing and storage resources. In the fiercely competitive and demanding information technology sector, cloud computing is currently a truly enormous technology that surpasses all previous computing technologies. Undoubtedly, the cloud computing concept holds great potential for business computing. It highlights the need for crucial infrastructure to support a new kind of service delivery that offers the advantage of cost savings through pooling computing and storage resources. In the fiercely competitive and demanding information technology sector, cloud computing is currently a truly enormous technology that surpasses all previous computing technologies.

Resource management is better handled by cloud computing since it relieves the user of the burden of finding resources for storage. A user can ask the cloud provider for more storage if needed, and when they're done using it, they can either release the storage by simply ceasing to use it or shift the data to a longer-term, less expensive storage option. This further enables the user to utilize more dynamic resources efficiently because they are free from the worry of storage

and the expense of using both new and old resources for the firm.

ACCESS CONTROL IN DATA SHARING

Limiting access to the resources and the system itself is the first step towards securing a system's data and resources. Access control, however, encompasses more than merely limiting the individuals (subjects) who have access to specific computer and network resources. Access control also controls files, users, and other resources. It regulates a user's access rights to resources (objects) or files. Access control systems need a number of processes to be completed before granting access to resources or objects in general, including identity, authentication, authorization, and accountability. Researchers and technologists understood early on in the development of computing and information technology how important it was to keep users from interfering with one another on shared systems. Numerous models of access control were created. The primary criterion for granting users access to the system or its resources was their identity. It was known as Identification Based Access Control (IBAC) strategy. But when networks and users increased, it was discovered that IBAC was insufficient to support such rapid expansion. Owner/group/public access control was one of the advanced concepts discussed. It turned out that distributed systems also had issues with IBAC. Controlling who has access to what resources and the system became difficult and error-prone. Role Based Access Control (RBAC) is a novel technique that was introduced. A user's access to the system is determined by Role-based Access Control (RBAC) depending on their job role. The least privilege principle essentially determines the role that a user is assigned. The position is defined with the fewest functionality or permissions required to complete the task at hand. If a role's privileges change, permissions can be added or removed. But when RBAC was applied to different administrative domains, issues surfaced. Additionally, it was challenging to agree on the privileges that should go along with each role. As a result, Attribute Based Access Control (ABAC), a policybased access control system, was created. Access to ABAC is based on attributes, like national number or date of birth that the user may demonstrate they possess. However, it can be quite difficult to agree upon a set of attributes, particularly when working across agencies, domains, and companies. Every access control technique depends on the user's authentication both on the site and when making the request. They are occasionally referred to as authenticationbased access control. All these techniques necessitate close connectivity between domains. This is done in order to specify roles or qualities, or to integrate identities. Furthermore, assigning subsets of an administrator's privileges is challenging with all of these methods.

Reducing the possibility of illegal access to logical and physical systems is the aim of access control. A key element of security compliance programs is access control, which guarantees that security technology and access control guidelines are in place to safeguard private information, including client data. The infrastructure and protocols of the majority of enterprises restrict access to files, applications, computer systems, networks, and sensitive data, including intellectual property and personally identifiable information. Because of their complexity, access control systems can be difficult to maintain in dynamic IT settings that combine cloud services and on-premises systems. Technology providers have switched from single sign-on systems to unified access management, which provides access restrictions for onpremises and cloud settings, following a few high-profile hacks.

related work

Hahn, et.al,…[1] Using our unique CPABE structure, propose a cloud-based IoT data management strategy that is both secure and effective. The suggested plan has effective bandwidth and storage management as well as the capacity to track down traitors who disclose their secret keys without authorization. Through the use of a user-specific transformation key, the cloud server may handle a large portion of the computational cost associated with decryption. Because the suggested access mechanism of the IoT data operates as follows: the cloud (1) authenticates the identity of the key holder, (2) uses the key holder's transformation key to partially decrypt the ciphertext, and (3) returns the partially decrypted result, this key is essential in preventing unauthorized access by a shared (or leaked) key holder. The key holder, who can get the plaintext from the partially decrypted ciphertext only if he is the original owner of the attribute key, is intimately linked to the transformation key. Consequently, the plaintext is not accessible to the other users who are in possession of the shared (or leaked) keys illegally. Users, the cloud, and the key authority make up the suggested plan. The authority to generate system-wide master, public, and decryption keys is vested in and authorized by the key authority. Users may function as message senders who encrypt messages or as message recipients who decrypt them. It is presumed that users possess malevolent intent, enabling them to distribute their private keys illegally. When it comes to transformation key management and partial decryption, the cloud genuinely complies with protocol, but it's interested in knowing whose communications are encrypted.

Abdollahi, et.al,…[2] Provide a CP-ABE technique based on the linear secret sharing scheme (LSSS) that requires four pairs to be made during the decryption process. The number of pairings in our system remains same as the number of needed attributes in the decryption is increased. The four components of the proposed concept are the cloud, data owners (DOs), data users (DUs), and trusted authority (TA). The generation of DU keys is the responsibility of the trusted authority. Every DU asks for a secret key based on a particular set of characteristics. Following authentication, the TA transmits the DU's outsourced key, okdu, to the cloud and the DU's secret key, skdu; to the DU. A portion of DU's key that is typically altered during revocation procedures is the outsourced key. Furthermore, it will be demonstrated by security proof that the system's security is unaffected by these outsourced keys (see to our security model in Section 4.3). A DO transmits the resulting ciphertext to the cloud using the public parameters P P for encryption. As a result, the ciphertexts and the outsourced keys are associated with two databases on the cloud. A DU initially notifies the cloud of a ciphertext. With the outsourced key okdu, the cloud partially decrypts data and generates the transform ciphertext tct. Lastly, the decryption is finished by the DU using the transform ciphertext (tct) and the secret key (skdu).

Fugkeaw, et.al,…[3] Create and implement a lightweight access policy updating system with granular access control for outsourced PHRs. Regarding PHR sharing, a patient or other data owner has the option to disclose their data to specific parties alone. Use symmetric encryption for data encryption in this case since it offers better encryption performance and enhances data access and policy updating performance. The symmetric key is encrypted using the CP-ABE technique. The cost of updating the policy only impacts the encrypted symmetric key because we encrypt it using the CP-ABE approach. Therefore, it is not necessary to re-encrypt every ciphertext. This lowers the computation cost at the proxy side greatly. Here, a proxy re-encryption method and CP-ABE are combined to provide lightweight policy updates. The suggested method makes use of a parallel processing methodology to facilitate effective data re-encryption. The data owner can easily amend the policies kept in the outsourced data storage according to our suggested plan. Additionally, CP-ABE's cryptographic details are apparent to users, making the tool usable.

Yang, et.al,…[4] Provide a unique solution that allows for effective access control for huge data in the cloud with dynamic policy updating. Our main goal is to create an externalized process for upgrading ABE system policies. The study being proposed focuses on finding a solution to the policy updating issue in ABE systems and suggests a productive way to outsource policy updating. Data owners just submit a policy updating query to the cloud server, and the cloud server may update the policy of the encrypted data without having to decrypt it, saving them the trouble of retrieving and re-encrypting the data. By fully utilizing previously encrypted data with outdated access controls in the cloud, our strategy not only meets all of the aforementioned requirements but also minimizes the calculation effort for the data owner and prevents the transfer of encrypted data back and forth. Each authority has autonomy from the others and is in charge of overseeing the characteristics of users within its jurisdiction. Additionally, it creates a secret key for each user based on their attributes and a secret/public key pair for each attribute in its domain. Users can access data stored on the cloud server through data access services. Updating ciphertexts from outdated access policies to current ones is another duty of the server. Before storing the data in the cloud, the data owners set the access policies and encrypt the information in accordance with them. Additionally, they request that the server change the clouds encrypted data's access restrictions. Every user has a unique global user identity that they can use to access the ciphertexts from the server at will. Only when the attributes meet the access policy specified in the ciphertext may the user decrypt it.

Yu, Jiguo, et.al,…[5] Provide ciphertext-policy encryption (CPABE) and key policy attribute-based signature (KPABS) in LH-ABSC, a lightweight ABSC scheme. The decision-making process on who can directly decrypt data for data owners is led by CPABS. In the meantime, the signature is connected to the attribute set of the data owner and can be utilized to confirm the legitimacy of the data. Specifically, LH-ABSC meets public verification requirements and has a consistent signature size, both of which are critical for Internet of Things devices. Our hybridpolicy ABSE scheme is designed by combining the ciphertext-policy ABE and the key-policy ABS. Specifically, our KPABS signature length is constant, meaning it doesn't rely on how many attributes are used, which lessens the communication load in the Internet of Things. We outsource signature, verification, and decryption to fog nodes due to the enormous pairing and modular exponentiation computations, which reduces the computational cost of resource-constrained IoT devices while maintaining low latency. ABSC is capable of effectively achieving both message confidentiality and ciphertext enforceability at the same time. Furthermore, KPABS works better when a central manager—such as the CEO of a company—wants to have authority over anonymous signatures inside a specific approved structure. As is typical for Internet of Things scenarios, CPABE can accomplish fine-grained one-to-many data exchange while preserving the data owner's right to determine who can decrypt the data.

Furthermore, our signature approach fulfills public verification and has a constant signature size, both of which are critical for Internet of Things systems.

background of the work

Sensitive information, such as medical data, needs to be fully protected in all respects. Because it offers encryption capabilities and access control enforcement, ciphertext-policy attribute-based encryption (CP-ABE) is a good choice for access control for data that is outsourced. In order to encrypt the data in CP-ABE, the data owner can designate an access policy made from the collection of attributes that is logically modeled using the logical gates AND, OR, and MofN. The requested ciphertext can only be decrypted with the satisfied secret key. There are various difficulties to be resolved for a feasible implementation of CP-ABE in the cloud-based IoT healthcare setting. First, because CP-ABE involves pairing and exponentiation procedures, its cryptographic cost is computationally high. In particular, the ciphertext encryption and decryption of CP-ABE is not appropriate for the data owner or user who utilizes devices with limited resources, like mobile or Internet of Things devices. The user can effectively decrypt the ciphertext with the use of the suggested work's CP-ABE decryption outsourcing. Even though there are a number of studies that suggest data consumers engage in partial CPABE decryption, most of them oppose outsourcing encryption. Furthermore, unless the data owner plans to encrypt the data using a device with limited resources, lightweight encryption is not necessary.

Cloud and Fog combined Access Control in Data Sharing Process

The suggested solution provides EMR sharing with policy update functionality that is safe, light-weight, and effective while maintaining privacy. The suggested plan makes use of blockchain technology to enable auditing and control over decentralized access authentication. This comprises various treatment records and secure healthcare data aggregation based on IoT data encryption. A dependable authority that manages the granting of permissions to users and data owners is in charge of this complex system, which strikes a carefully calibrated balance between security and accessibility. The system's primary function is to securely store medical records while streamlining data organization and retrieval through the use of various indexing algorithms. Sensitive medical record data is encrypted using the trusted Advanced Encryption Standard (AES) cryptographic technique to protect its confidentiality and integrity. In order to bring computing capabilities closer to the data source, medical data processing is tactically carried out in a fog layer. This maximizes resource usage while enhancing real-time processing capabilities. Encrypted medical records reside on a cloud server as an extra security measure, strengthened by blockchain technology's imperviousness. By using blockchain technology, data is saved with tamper-proof integrity and an unchangeable record of all transactions and access requests is created. This adds a transparent and auditable aspect to data management while also strengthening the system's overall security posture. Data users in this system take on several roles, each of which has certain access privileges. The process of verification is overseen by a thorough Role-Based Access Control (RBAC) system, which examines the user's identity and given role when requesting files. The notion of least privilege is reinforced by requiring adherence to preset RBAC rules in order to access requested data later. In addition to strengthening the system against unwanted access, the integration of AES encryption, fog layer processing, cloud storage with blockchain, and RBAC also creates an open and auditable framework for healthcare data management.

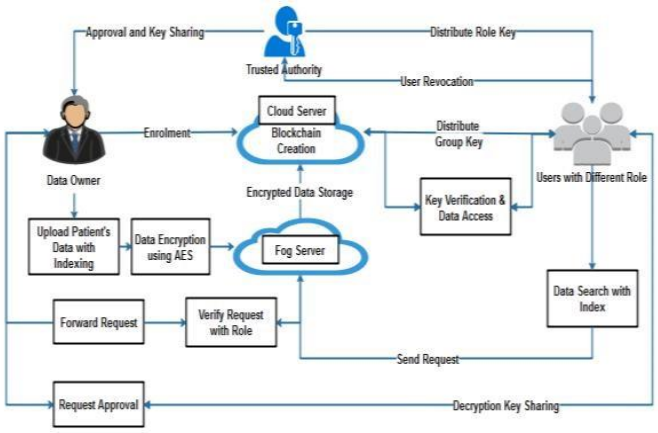

Fig 1: Proposed Framework

Framework Creation

Build a local cloud and offer reasonably priced ample storage services in this module. Users can upload and exchange data in the cloud once they have space available from the cloud. Cloud storage can be implemented in this task with a high level of security. Users do not, however, fully trust the cloud because it is quite likely that the CSPs are not within their trusted domain. A secure data sharing framework that is being proposed allows group members and data owners to communicate. Proposed system model comprises different types of entities: Cloud Service Providers (CSPs), Fog Nodes (FNs), Blockchain, Trusted Authority (TA), Data Owners (DOs), and Data Users (DUs).

Permissions for data owners and users are still largely determined by the trusted authority. This guarantees that access rights are appropriately controlled.

Data Upload with Index

Several indexing techniques are used to safely store medical information, facilitating effective data organization and retrieval. It is possible to divide the medical data using distinct index structures. Information on diseases, medical records, and personal health data are used in this index. The patient's data is uploaded by the data owner, and it is automatically divided according to each index. This improves the medical records' general accessibility and administration.

Data Encryption

The medical records are encrypted using AES to guarantee data integrity and confidentiality. Strong protection against unauthorized access is offered by this encryption layer. Medical records' confidentiality and integrity are protected by AES encryption, which also offers a robust barrier against illegal access and data breaches. The encryption procedure used on the fog layer. By carrying out data processing operations, the fog layer moves computational power closer to the data source. Performance may be enhanced as a result, particularly for needs involving realtime processing. Finally, blockchain technology is used for increased security and encrypted medical records are kept on a cloud server. Blockchain protects the data's integrity, making it impervious to manipulation.

Role Based Access Control

During file requests, the user's identification, allocated role, and verification procedure are managed by the Role-Based Access Control (RBAC) system. The access rights of data users are determined by their respective jobs. During the access request procedure, the user's role must be confirmed by the RBAC system. This fosters a more regulated and safe atmosphere and lowers the possibility of unwanted access. By building a data structure for indexing, users can search patient data using their id and keyword, making it possible to search through big datasets more quickly and effectively. Depending on the type of data and the specifications of the search query, choose the relevant index. Indexing can speed up the process of running search queries and increase the system's overall effectiveness.

Request Approval

Depending on their jobs, data users can ask for particular files. The identity of the user and the assigned role are both validated during the verification process. Access to the requested data is either permitted or refused in accordance with the RBAC rules after the user's identity and role have been confirmed. By doing this, users are guaranteed access to only the information required for their roles. The for layer handled the processing of this request. Users can access certain medical records through a secure file request and verification process that follows strict authentication guidelines. Following the completion of the fog layer verification process, the request will be forwarded to the designated data owner. Ultimately, the data owner grants the requested user access rights. The data user will then receive the secret key from the system.

Verifiable Data Access

The user authenticates via a secure means to start the data access process. The user requests access to particular data that is kept on the cloud server after completing the authentication process. The user's request is examined by the cloud server, which also confirms the associated secret key. This secret key may be supplied independently via a secure channel or produced during the authentication procedure. The verification makes that the user has the right secret key, which is needed to access the requested data. Strong encryption techniques, such AES, were first used to encrypt the data in order to guarantee confidentiality during transmission and storage. The cloud server securely sends the plain, decrypted data to the verified user after verifying the data integrity. Now that the data is in an understandable format, the user can examine or work with it in accordance with their responsibilities and access rights.

METHODOLOGY

AES ALGORITHM

Data is encrypted and decrypted using a block cipher by the symmetric Advanced Encryption Standard (AES) algorithm. The standard defines three key sizes: AES-128, AES-192, and AES-256. The algorithm consists of the following steps: **Key Expansion:** The 128-bit, 192-bit or 256-bit encryption key is expanded into a key schedule of 10, 12, or 14 round keys, respectively. The round keys are derived from the original encryption key using a key schedule algorithm.

Initial Round: The first round key is used to XOR the plain text after it has been separated into 128-bit blocks. **Rounds:** The encryption process consists of a set of rounds (10, 12, or 14) that operate on the state of the cipher. Each round consists of four transformations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.

SubBytes: Every byte in the state is swapped out with an equivalent byte from an S-box. This stage creates ambiguity and aids in avoiding linear cryptanalysis.

ShiftRows: The state's rows are all moved cyclically by a predetermined amount of steps. There are three row shifts: one step for the second row, two steps for the third row, and three steps for the fourth row.

MixColumns: A fixed polynomial is used to multiply each state column. In addition to providing diffusion, this step aids in preventing differential cryptanalysis. **AddRoundKey:** The state and the round key for the current round are XORed.

Final Round: The final round is the same as the previous rounds except that it does not include the MixColumns transformation.

Output: The resulting cipher text is the final state of the cipher.

The encryption procedure is reversed during decryption. The final round key is XORed with 128-bit blocks of the ciphertext. The rounds are then carried out using the inverse of each transformation, in reverse order. The original plain text is what remains in the end.

BLOCKCHAIN CREATION

1.A Block containing information about current transactions.

2. Each data generates a hash.

3. A hash is a string of numbers and letters.

4. Transactions are entered in the order in which they occurred.

5.The hash depends not only on the transaction but the previous transaction's hash.

6.Even a small change in a transaction creates a completely new hash.

7.The nodes check to make sure a transaction has not been changed by inspecting the hash.

8.If a transaction is approved by a majority of the nodes then it is written into a block.

9.Each block refers to the previous block and together make the Blockchain.

10.A Blockchain is effective as it is spread over many computers, each of which has a copy of the Blockchain.

Conclusion

Future data sharing will be possible on the cloud due to the rapidly increasing demand for data sharing. A review of safe data sharing in a cloud computing context was proposed and presented. Data owners outsource their data to cut costs. Because cloud service providers are third-party providers, data owners do not have control over their data. This method offers an effective keyword search procedure along with index building. Here, the cloud service provider can grant access rights to the data owner. Next, upload the files to the cloud using a variety of keywords. All searchable keywords have an index term that is set using an indexing process. The privacy and security concerns are the crux of the cloud data sharing problem. This study discusses a number of methods, including AES encryption, group data sharing, and user revocation, to promote privacy and secure data sharing. The study finds that in order to implement privacy and security in group sharing, a safe anti-collision data sharing method for groups' offers more efficiency, supports access control mechanisms, and ensures data confidentiality.

REFERENCES

[1] Hahn, Changhee, Jongkil Kim, Hyunsoo Kwon, and Junbeom Hur. "Efficient iot management with resilience to unauthorized access to cloud storage." IEEE Transactions on Cloud Computing 10, no. 2 (2020): 1008-1020.

[2] Abdollahi, Sina, Javad Mohajeri, and Mahmoud Salmasizadeh. "Highly Efficient and Revocable CP-ABE with Outsourcing Decryption for IoT." In 2021 18th International ISC Conference on Information Security and Cryptology (ISCISC), pp. 81-88. IEEE, 2021.

[3] Fugkeaw, Somchart. "A lightweight policy update scheme for outsourced personal health records sharing." IEEE Access 9 (2021): 54862-54871.

[4] Yang, Kan, Xiaohua Jia, Kui Ren, Ruitao Xie, and Liusheng Huang. "Enabling efficient access control with dynamic policy updating for big data in the cloud." In IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pp. 2013-2021. IEEE, 2014.

[5] Yu, Jiguo, Suhui Liu, Shengling Wang, Yinghao Xiao, and Biwei Yan. "LH-ABSC: A lightweight hybrid attribute-based signcryption scheme for cloud-fog-assisted IoT." IEEE Internet of Things Journal 7, no. 9 (2020): 7949-7966.

BIOGRAPHY

Mrs.RAMYA R,attained her B.E. in Computer Science and Engineering. she completed her B.ed in 2022. she is pursuing her M.Tech (Part Time) in Computer Science and Engineering at PRIST University, Thanjavur, Tamil Nadu, India.

Mrs. K Jayanthi has been working as an Assistant Professor in Computer Science and Engineering at PRIST University, Thanjavur, Tamil Nadu, India.