ISSN- 2394-5125 VOL 9, ISSUE 05, 2022

P-MOD: Protected Priority-Based Multi tiered Managerial Information Sharing in Cloud Technology

T Jaya Sri,[1] V V Deepthi Meghana[2], G Lakshmi Prasanna[3], Ch Pawan Kalyan[4], M Vani[5]

[1] Asst. Professor, Department of Computer Science and Engineering

[2,3,4,5] Student, Department of Computer Science and Engineering

[1,2,3,4,5]QIS College of Engineering & Technology

Abstract

In summary, cloud computing has modified how businesses handle data storage, accessibility, and collaboration. A steady stream of large data sets are being uploaded to the cloud and shared around a tier of administrators. Consisting of a large number of users, each of whom has unique permissions and roles.

Finding a safe and effective data access structure has become a key research challenge as more and more data storage is moved to the cloud. In this research, we present a method for the management and sharing of large datasets called Privilege-based Multilevel Organizational Data-sharing (P-MOD), which combines a privilege-based access structure with an attribute-based encryption mechanism. Our suggested privilege-based access structure facilitates the management of healthcare records utilising mobile healthcare devices by reducing the difficulty of building hierarchies as the number of users increases. It may also help businesses use big data analytics to get a more complete picture of population dynamics. If the DBDH assumption holds, then P-MOD is shown safe against an adaptively selected plaintext attack. Full simulation and performance evaluations on the actual U.S. Census Income data set show that P-MOD reduces computing complexity and space requirements compared to the current methods.

INTRODUCTION

RBAC is built on hypothetical options for positions. To adequately encapsulate the rights granted to each user of the system, an ever-increasing number of

RBAC roles would be required. Role explosion [5] describes how managing a large number of rules may become a time- and energy-consuming process.

Consider the situation where people upload their PHR to the cloud where they may be viewed by hospital staff and healthcare professionals to better grasp the significance of this research. Patients often want their doctors to have full access to their phrs (including the more private sections like their medical histories), whereas an administrator is given just restricted, less private access (e.g. Date of birth). That may be accomplished if the patient establishes a set of criteria by which different hospital staff members are granted access to their medical records. Next, the patient must specify the permissions at each tier in order to outline the informational materials available to each data consumer. To fully appreciate the complexity of phrs, one must first acknowledge that patients hold a wide range of conservative values. Some patients, for instance, may only want to share certain aspects of their PHR with certain doctors and keep others out.

In this work, we suggest a strategy for exchanging data across many organisational levels that is based on the concept of privilege. It expands upon ideas introduced into address the challenges of information exchange in hierarchically complicated organisations. This paper's primary contributions are summed up as follows.

Separating large data sets into smaller chunks is a common problem, and we show you many ways to andto enable data sharing in hierarchical contexts, you should provide a privilege-based access structure.

Using the Decisional Bilinear Diffie-Hellman (DBDH) assumption, we formally verify P-security MOD's and demonstrate that it is immune to plaintext assaults that are selected adaptively.

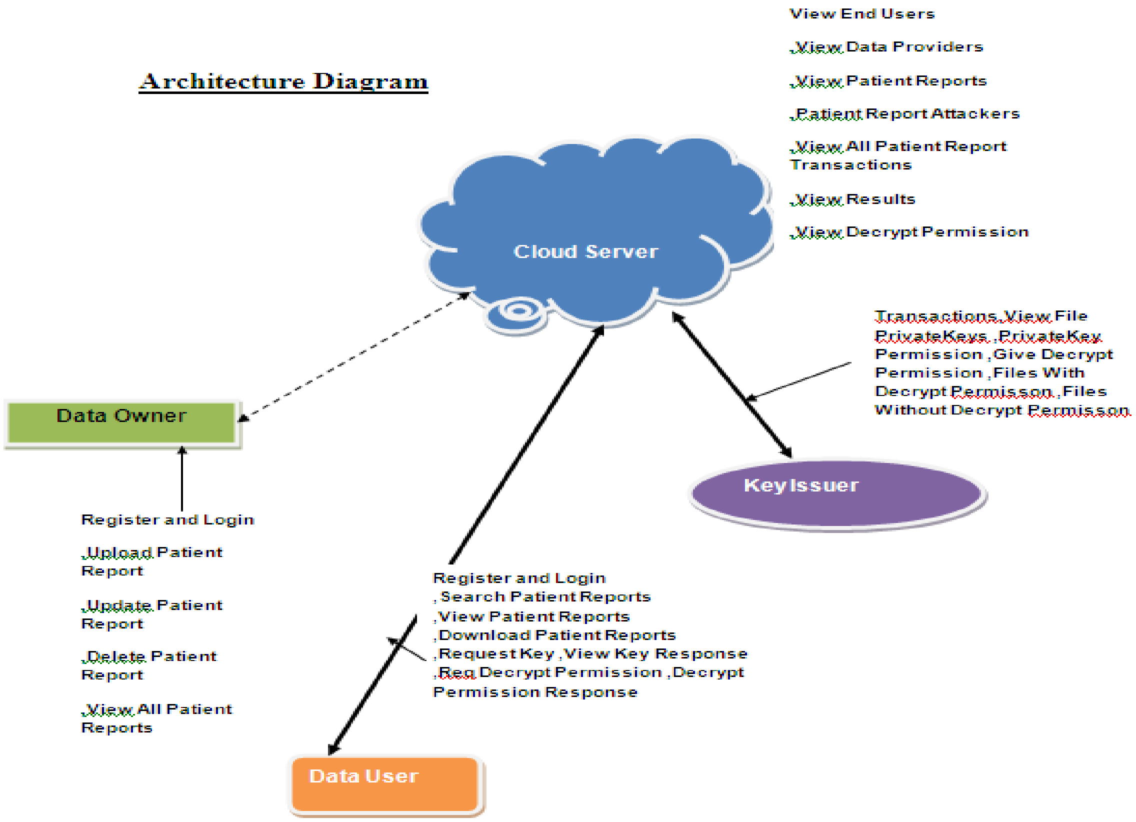
• We provide a performance evaluation of P-MOD and evaluate it against three existing] with comparable hierarchical aims.

Using the actual data from the United States Census on individual incomes, we develop P-MOD and run extensive simulations under a variety of scenarios

[10]. Our findings are compared to simulations of the identical circumstances run on two different methods [7], [9].

Here is how the remainder of the paper is laid out. The relevant literature is discussed in provides an overview of the study by introducing and briefly discussing the most important underlying principles.

Section IV then goes on to detail the problem formulation, which includes an outline of the design objectives and a model of the resulting system. A comprehensive description of the P-MOD system that has been suggested may be found in Section V. Next, we use the difficulty of the Decisional Bilinear Diffie-Hellman (DBDH) issue to explicitly demonstrate PMOD's security in Section VI. The effectiveness of P-MOD is evaluated in Section VII. Section IX concludes with some last thoughts on what this research has shown.


Architecture Diagram
View End Users
,View Data Providers
,View Patient Reports
,Patient Report Attackers
,View All Patient Report Transactions
,View Results
,View Decrypt Permission
Cloud Server
Transactions,View File PrivateKeys ,PrivateKey Permission ,Give Decrypt Permission ,Files With Decrypt Permisson ,Files Without Decrypt Permisson
Key Issuer
Data Owner
Register and Login
,Upload Patient Report
,Update Patient Report
,Delete Patient Report
,View All Patient Reports
Register and Login
,Search Patient Reports
,View Patient Reports
,Download Patient Reports
,Request Key ,View Key Response
,Req Decrypt Permission ,Decrypt Permission Response
Data User

Related work

For cloud-based data sharing, [11] developed Fuzzy Identity-Base Encryption (Fuzzy IBE) as a flexible encryption method. The ciphertext is stored in the cloud, where only authorised users may access it. Users encrypt their data with the intention of only allowing authorised others access to it, and must thus seek a private key from a key-issuer in order to decode the data.

The private key of the data user and the ciphertext are both associated with characteristics in fuzzy IBE [12], making it a subset of function encryption. Each person or thing has its own unique set of characteristics, or attributes, that may be described in detail. Attributes provide you more leeway in terms of data access permissions since they may be any variable. Assigning descriptive metadata to cloud-stored ciphertext and private keys is made possible by this technique. In order to decode the ciphertext, the data user's private key must include at least some characteristics that are identical to those embedded in the ciphertext. While this technique makes it simple to characterise complicated systems in terms of attributes, it loses efficiency when used to represent either huge systems or a high number of characteristics.

More flexibility in data exchange was required, thus attribute-based encryption (ABE) techniques were developed. Attributes and authorization rules are both included into these systems.

A system's access policy is a set of declarative statements that link attributes to specify which users have access and which do not. Each ciphertext is annotated with a set of descriptive characteristics in Key-Policy Attribute-Based Encryption (KP-ABE) [13], while each private key is combined with an access policy in Ciphertext Policy Attribute-Based Encryption (CP-ABE) [7]. In order to decode the ciphertext, authorised data users must first get a private key from the keyissuer. Keys are issued by an entity that also incorporates the access policy within the key itself. When the access policy included in a user's private key matches the ciphertext's associated set of descriptive characteristics, the user is able to decode the ciphertext. KP-ABE is more adaptable than Fuzzy IBE and allows for more granular access control. The data owner, however, must have faith in the key issuer that it will only give out private keys to authorised individuals. This constraint arises because, in the end, the data owner gives up control over which users have access to their data.

Conversely, Role-Based Access Control (RBAC) [14] is seen as conceptually comparable to CP-ABE. In this way, the data owner may decide which users have access to decipher specific ciphertexts. This occurs when the data owner embeds the access structure into the ciphertext itself. It ensures that the keyissuer's private key has only the characteristics that match those held by the data user. In order to improve efficiency and adaptability, other CP-ABE systems [15]-[19] were developed afterwards.

When data users are not organised in a hierarchy and may be treated as individuals, attribute-based encryption techniques like Fuzzy IBE, KP-ABE, and CP-ABE are preferable (i.e. No relationships). However, in the case of huge multilevel organisations, they both suffer from the same problem: they are very computationally difficult. In order to decrypt a single file using one of these methods, a significant number of characteristics (at varying depths) must be added to the encryption.

The Hierarchical Identity-Based Encryption (HIBE) [20] scheme and the Cipher Text-Based Encryption (CP-ABE) [7] were eventually combined in [8], [21] to form Hierarchical Attribute-Based Encryption (HABE).

HABE allows for granular security in hierarchical settings. There is a root master responsible for producing and dispersing parameters and keys, many intermediate domain masters who delegate keys to lower-level domain masters, and a large number of end users. Like the HIBE system, keys in this one are created using a hierarchical structure. When expressing an access policy, HABE makes use of the disjunctive normal form, which combines all characteristics managed by the same domain authority into a single conjunctive sentence. When many domain authorities are in charge of managing identical characteristics, this system becomes impractical. A complicated organisation with various domain authorities may find it difficult to synchronise attribute management. Subsequently, in [22]–[25], for instance, more hierarchical scheme examples were provided.

One of the newest hierarchical solutions is File Hierarchy Ciphertext Policy Attribute-Based Encryption (FH-CP-ABE) [9]. The paper presents a tiered authorization system for handling a hierarchical organization's dissemination of information with varying degrees of confidentiality. It was suggested that a single access structure may be used to depict both the organisational hierarchy and its access regulations. A root node, intermediate nodes, and leaf nodes make up this access structure. Root and transit nodes both take the form of gates (i.e. AND or OR). Attributes held by data consumers are represented by the leaf nodes.

System analysis

EXISTING SYSTEM

Later, attribute-based encryption (ABE) systems came into existence to provide more flexibility during information exchange. Attributes and security rules are both included into these systems. To define which users have access to the system and which do not, administrators employ access policies, which are statements that link characteristics. Two distinct methods for introducing ABE schemes—Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext Policy Attribute-Based Encryption (CP-ABE)—have been developed (CP-ABE). In [10], KP-ABE was first described. Each ciphertext in KPABE is given a set of metadata labels, and each private key has its own access policy built in. Users with proper access to the data will need to get a private key from the key-issuer in order to decode the ciphertext. Keys are issued by an entity that also incorporates the access policy within the key itself. An access policy included into a user's private key will determine whether or not the user is able to decode a ciphertext. KP-ABE has greater wiggle room than Fuzzy IBE and can provide fine-grained access control. The data's owner, however, must have faith in the key issuer to limit the distribution of private keys exclusively to those who have been authorised to view the data. Ultimately, the data owner loses the ability to decide which people will be permitted access to their data, which is a significant restriction.

□

Another method, CP-ABE, was subsequently suggested in [7]. It has been compared to RBAC [11] in terms of its general ideas behind the system. With CP-ABE, however, the data owner may decide which users have access to decipher specific ciphertexts. This occurs when the data owner embeds the access structure inside the ciphertext. It ensures that the key-private issuer's key has only the characteristics that match those of the data user. Later on, a number of improved CP-ABE systems [12–15] were devised, each with the potential to increase adaptability and efficiency.

Fuzzy IBE, KP-ABE, and CP-ABE are some of the most used attribute-based encryption algorithms, and they excel in situations where data users do not form a hierarchical structure and may act autonomously of one another (i.e. No relationships). However, they both suffer from the same issue—high computational complexity—when applied to big multilevel organisations. In order to decrypt a single file using one of these methods, a significant number of characteristics (at varying depths) must be added to the encryption.

Later, in [17], Hierarchical Attribute-Based Encryption (HABE) was presented, which merged the Hierarchical Identity-Based Encryption (HIBE) [16] scheme with the Cipher Text-Based Encryption (CP-ABE) [15,16]. Within a hierarchical framework, HABE is able to implement granular permissions. There is a root master responsible for producing and dispersing parameters and keys, many intermediate domain masters who delegate keys to lower-level domain masters, and a large number of end users. The keys in this scheme are created using the same hierarchical method as the HIBE scheme. In order to define an access policy, HABE employs a disjunctive normal form, whereby all attributes are governed by the same domain authority into a single conjunctive sentence. When many domain authorities are responsible for managing identical attribute copies, this technique becomes impractical to apply. When dealing with big organisations with various domain authorities, synchronising attribute management might become a difficult problem. In [18], [19], for instance, other hierarchical structures are shown as examples.

Disadvantages

In the current system, one of the biggest problems for data owners is how to safely and effectively assign different levels of access permissions to different datasets.

Attribute-based encryption is not available in the current implementation.

PROPOSED SYSTEM

The suggested system proposes a Privilege-based Multilevel Organizational Data-sharing scheme (P-MOD) to address the difficulties associated with coordinating information flow in hierarchically-diverse workplaces. The technique first suggests splitting a file into many sections with varying degrees of importance. Then, symmetric encryption is performed on each individual piece. The information that is really sent to consumers is the encryption keys themselves. The concept goes on to provide an access structure that categorises the users of an organization's data into distinct roles.

Each tier has its own access tree, which specifies what capabilities each tier's data users have. Users at various levels in the hierarchy are given varying degrees of access to the data file by first encrypting the whole file once. The encryption and decryption procedures are based on an Attribute Based Encryption (ABE) framework, which permits very fine-grained control over access.

Advantages

The significance of the proposed system may be better grasped by visualising a scenario in which people upload their Public Health Records (PHR) to the cloud, where they would be accessible to hospital staff and healthcare practitioners.

Data users may successfully decode a cypher text if the collection of descriptive qualities associated with the cypher text fulfils the access policy embedded into their private keys, hence the system is effectively protected.

Implementation

• Data Owners

Here, the data provider will transfer their encrypted patient data to the remote Cloud server. The user adds an extra layer of protection by encrypting the file before uploading it to the server. A user that has the ability to decrypt a file may then execute the following tasks: You can do things like see all patient reports, edit existing ones, delete them entirely, and upload new ones.

In the Cloud Server

The primary function of the cloud server is to facilitate data storage for the Data Owners. Users may see End Users, Data Providers, Patient Reports, Patient Report Attackers, All Patient Report Transactions, Results, View Decrypt Permission, and Private Key Permission once they have encrypted and stored their data files on the Server.

Data User

A secret key is required in order to access the data file in this component. Search Patient Reports, View Patient Reports, Download Patient Reports, Request Key, View Key Response, Req Decrypt Permission, Decrypt Permission Response, and more are all possible using this interface.

Key Issuer

The following procedures are carried out by the key issuer while in this section. Files with decryption permission, files without decryption permission, files with private key permission, files with private key permission, and files with decryption permission.

CONCLUSION

With so many positives associated with cloud computing, it's no surprise that many big, hierarchical businesses have begun moving their data storage and sharing operations there. The primary focus of this work is to address the key security concerns of data owners when it comes to cloud-based data sharing. Up next, the most extensively used and studied There is a short discussion of data sharing strategies, highlighting the limitations of each. This work presents a Privilege-based Multilevel Organizational Datasharing system (P-MOD) to overcome these issues, allowing for the safe and effective sharing of data in the cloud. P-MOD divides a file into several sections, each of which may be accessed only by users with the appropriate permissions. Data users' access rights determine which parts of the file they may access. Assuming the DBDH assumption holds, we formally demonstrate that P-MOD is safe from an ad hoc selected plaintext attack. We compare P-performance MOD's and simulation results to those of the three most prominent methods and find that P-MOD dramatically reduces computational complexity while decreasing storage space. To facilitate the development of attribute-based, safe data management and smart contracts in the future, we present a system that we believe will do just that.

REFERENCES

[1] Ponemon Institute, "Sixth Annual Benchmark Study on Privacy and Security of Healthcare Data,"

[2] More than half of American firms are already using cloud computing, according to research by R. Cohen, author of "The cloud reaches the mainstream." Retrieved from According to "The NIST definition of cloud computing"),

[3] 2010 Economic Analysis of Role-Based Access Control, A. C. Oconnor and R. J. Loomis, National Institute of Standards and Technology, Gaithersburg

[4] "Role explosion: Acknowledging the issue." by A. Elliott and S. Knight in Software Engineering Research and Practice,

[5] , E. Zaghloul, T. Li, and J. Ren presented "An attribute-based distributed data sharing method."

[6] "Ciphertext-policy attributebased encryption," by J. Bethencourt, A. Sahai, and B. Waters;

[7] Computers & Security,

ISSN- 2394-5125 VOL 9, ISSUE 05, 2022

[8] G. Wang, Q. Liu, J. Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data on cloud servers."

[9] Xie, "An efficient file hierarchy attribute-based encryption system in cloud computing," IEEE Transactions on Information Forensics and Security,

[10] The UCI machine learning repository, by M. Lichman (2013)

[11] Anand Sahai and Brian Waters, "Fuzzy identity-based encryption," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer,.

[12] Functional Encryption: Definitions and Challenges, D. Boneh, A. Sahai, and B. Waters, Theory of Cryptography Conference,

[13] Attribute-based encryption allows for more nuanced control over who can view encrypted data.

[14] rained access control of encrypted data," Proceedings of the 13th ACM Conference on Computer and Communications Security, D. F. Ferraiolo and D. R. Kuhn in 2009, "Role-based access restrictions" is an example of this.

[15] "Provably secure ciphertext policy ABE," written by L. Cheung and C. Newport, [16]"Secure data retrieval based on ciphertext policy attribute-based encryption (CP-ABE) system for the dtns," tech. Rep., Citeseer

[17] Completely Secure Ciphertext-Policy Hiding CPABE, J. Lai, R. H. Deng, and Y. Li, International Conference on Information Security Practice and Experience

[18] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partly disguised access patterns," in Proceedings of the 7th ACM Symposium on Information, Computer, and Communications Security

[19] CP-ABE with constant-size keys for lightweight devices. F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan

[20]International Conference on the Theory and Application of Cryptology and Information Security

[21] "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," G. Wang, Q. Liu, and J. Wu,

[22] European Symposium on Research in Computer Security, pp. 587-604, Springer, 2009, "Attribute-sets: a realistically driven addition to attribute-based encryption" by R. Bobba, H. Khurana, and M. Prabhakaran.

[23] Hasbe is a hierarchical attribute-based method for flexible and scalable access control in cloud computing, as described

[24] To encrypt short ciphertexts using a hierarchical attribute-based policy, "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," Information Sciences,

[25] Mobile Networks and Applications, J. Li, Q. Wang, C. Wang, and K. Ren, "Enhancing attribute-based encryption using attribute hierarchy."

[26] In the Proceedings of the twenty-first annual ACM Symposium on Theory of Computing, M. Naor and M. Yung wrote about universal one-way hash functions and their cryptographic applications

[27] An expressive, efficient, and provably secure implementation of ciphertext policy attribute-based encryption. B. Waters

[28] J. Wang's "ciphertext-policy attribute based encryption java toolkit."

[29] "JPBC: Java pairing based cryptography," by A. De Caro and V. Iovino,