International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 6 ǁ 2021 ǁ PP. 15-19

Secure Encrypted Medical Health Record Sharing In Cloud Using Duplicate Storage System

N.Mohamed Salmaan,S.Barath Kumar,S.Gogul
Department of Information Technology, A.V.C. College of Engineering, Mannampandal, Mayiladuthurai.

Abstract-*In cloud secure personal data sharing is the important issues because it creates several securities and data confidentiality problem while accessing the cloud services. Many challenges present in personal data sharing such as data privacy protection, flexible data sharing, efficient authority delegation, computation efficiency optimization, are remaining toward achieving practical fine-grained access control in the Personal Information Sharing system. Personal records must be encrypted to protect privacy before outsourcing to the cloud. Aiming at solving the above challenges, here propose an efficient data sharing mechanism for Personal Data Sharing, which not only achieves data privacy, fine-grained access control and authority delegation simultaneously. Proposed methodology is presented to secure patients' MHR (Medical Health Record) in the healthcare cloud using the duplicate generation technique with a two server based computing facility. Duplicate server serves as a second gallery to contain duplicate MHR that appear to the attacker as if it is the original MHR. When user uploading a file on original server, corresponding duplicate file will be stored on another server. In this method, the decoy files are called when an attacker is detected as accessing the system, in our proposed methodology the duplicate files are retrieved from the beginning to ensure better security. In proposed approach RSA algorithm is implement to encrypt the medical records.*

KEY WORDS: *Data sharing mechanism, cloud computing,secured outsourced computation,attribute based encryption, electronic medical record.*

Date of Submission: 25-05-2021 Date of acceptance: 07-06-2021

I. INTRODUCTION

Proposed system adopt two different public cloud servers to achieve secure outsourced computation, such as outsourced key generation/encryption/re-encryption key generation/ decryption. Actually, one public cloud server (e.g., public cloud 2) is sufficient for outsourced decryption, but not enough for other operations, because all the secret will be exposed to the unique cloud server. The access control model consists of five entities: private key generator (PKG), public cloud 1, public cloud 2, data owners and data consumers. Proxy Re-encryption is used to re-encrypt the data before sending it to the data consumer. Here propose an efficient data sharing mechanism for Personal Data Sharing, which not only achieves data privacy, fine-grained access control and authority delegation simultaneously, but also optimizes the computation efficiency and is suitable for resource constrained servers. Most of the data consumers are honest, while few of them are corrupt and will leakage their secret keys in the collusion. On the contrary, PKG and data owner are assumed to be fully trusted. Besides, public cloud 1 and public cloud 2 cannot collude with each other. The non-collusive assumption is reasonable, because the client can demand that two cloud servers cannot reveal users' information by contract. In proposed work, PR-ABE (Attribute Based Encryption with Proxy Re-encryption) technique implements to provide secure encryption of medical data. To improve the access control, here partial key sharing scheme will be implement. Using this, data owner can send partial secret key for the requested user. This approach overcomes the key guessing attack in data retrieval process. Proposed system will be implementing using PHP as front end and SQL is for back end process. This approach has modules like Framework Creation, Medical files uploading, Data Encryption, duplicate Storage, File access and alert system. Input process has file storage and output was provide secure to medical files using two cloud.

II. LITERATURE SURVEY

Tiwari et al propose a ciphertext-policy attribute-based proxy re-encryption scheme. In the proposed scheme, we design an efficient fine-grained revocation mechanism, which enables not only efficient attribute-level revocation but also efficient policy-level revocation to achieve backward secrecy and forward secrecy. The SecCloudSharing protocol develops a ciphertext-policy attribute-based proxy re-encryption (CP-AB-PRE) scheme to delegate a data owner controlled policy revocation process to a cloud server. The cloud server transforms a ciphertext associated with an access policy to another ciphertext under a new access policy without revealing the secret key of the data owner and the message. A KGC is a global central entity that initializes the system by publishing the public parameters. The KGC initializes the attribute revocation process by generating a

random number corresponding to the revoked attribute on the request of the cloud server and sends it to the A-KACs. Each A-KAC updates the revoked attribute of the user by combining with the random number generated by the The KGC initializes the system functionality by publishing the public parameters and also controls the functionality of the cloud server and A-KACs. A cloud server is configured as a data storage server and a proxy mediator server to provide data services to the users. SecCloudSharing overcomes the problem of single-point performance bottleneck by defining a collaborative key distribution configuration. It provides policy-level revocation and attribute-level revocation techniques, which in turn support forward and backward secrecy of data. It uses the hash derivation function to check the correctness of the valid access policy (executed by the proxy-mediator server and user) by validating the authenticity of the data owner. We conducted a detailed security and performance analysis to verify that our scheme is secure and efficient. The security analysis showed that our scheme could effectively resist to individual and colluded malicious users, as well as the honest-but-curious cloud server. Further, the performance analysis showed the superiority of our scheme over the traditional CP-ABE–based access control schemes for public cloud storage. Accessibility information (contains a global user id, a user attributes set, and an attribute set group id) to the user, where the attribute set group id is communicated in secure manner.

Zhang, Y et al propose an online/offline MA-ABE scheme, which realizes both the online/offline secret key generation and the online/offline encryption while supporting a fully large attribute universe. In the offline phase, one global-identity authority and multiple attribute authorities doing the majority of the work to issue attribute secret keys before knowing users' global identity and attributes. The data owner can perform most of the encryption computation tasks before knowing the actual message and access structure. Furthermore, the online phase can rapidly assemble the final decryption key and ciphertexts when related specifications become known. Particularly, global identity authority and attribute authorities need not to cooperate in the whole process. In the proposed system, the computation required for the generation of user global-identity secret keys, the generation of user attribute secret keys and the encryptions of messages are split into an offline phase and an online phase. In the offline phase, GA and AAs do the majority of the work to issue attribute secret keys before knowing users' global identity and attributes. The data owner can perform most of the encryption computation tasks before knowing the actual message and the access structure. Furthermore, the online phase can rapidly assemble the final decryption key and ciphertexts when related specifications become known. The technique of online/offline digital signature (OOS) is used by AAs to efficiently generate a signature on users' attribute secret keys. GA further generates users' global-identity secret keys, and hence, the decryption key for users only when the online signature is valid. Theoretical analysis and performance comparisons indicate that the proposed OO-MAABDS system is extremely suitable for resource constrained users in mobile cloud computing.

Alderman et al propose a rigorous definitional framework for a CES that enforces read-only information flow policies (which encompass many practical forms of access control, including role based policies). This framework (i) provides a tool by which instantiations of CESs can be proven correct and secure, (ii) is independent of any particular cryptographic primitives used to instantiate a CES, and (iii) helps to identify the limitations of current primitives (e.g. key assignment schemes) as components of a CES.

Li et al propose a new attribute-based data sharing scheme suitable for resource-limited mobile users in cloud computing. The proposed scheme eliminates a majority of the computation task by adding system public parameters besides moving partial encryption computation offline. In addition, a public ciphertext test phase is performed before the decryption phase, which eliminates most of computation overhead due to illegitimate ciphertexts. For the sake of data security, a Chameleon hash function is used to generate an immediate ciphertext, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts.

In order to realize secure attribute-based data sharing (ABDS) suitable for resource-constrained mobile users, we introduce a new online/offline ABE scheme that eliminates a majority of the computation task by adding system public parameters besides moving the encryption computation overhead on the data owner's side to the offline phase. A public ciphertext test phase is performed before the decryption phase, which eliminates most of the computational cost resulted from illegitimate ciphertexts. In other words, the public ciphertext test allows a user to check at a low cost whether a potential equation holds for components of a given ciphertext before performing the expensive decryption phase. The technique of Chameleon hash function is used to generate an immediate ciphertext, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts. In this way, the proposed scheme is proven CCA2 secure, which is widely recognized as a standard security notion. Theoretical analysis and experimental results indicate that the proposed ABDS system is extremely suitable for resource limited mobile users in cloud computing.

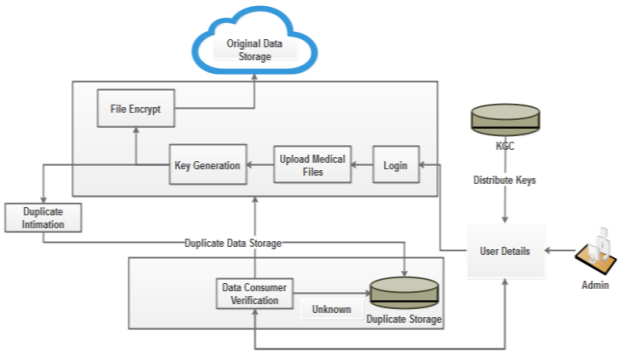
Castiglione et al propose a construction for hierarchical and shared key assignment which uses as building block a threshold broadcast encryption scheme. We denote such a construction as the Threshold Broadcast Encryption Based Construction (TBEBC). Threshold Broadcast Encryption: A broadcast encryption scheme allows a sender to broadcast an encrypted message to a set of users in such a way that only legitimate users can decrypt it. Broadcast encryption schemes can be either publickey or symmetric-key based. In the

symmetric-key setting, only a trusted authority can broadcast data to the receivers. Conversely, in the public-key setting, a public key published by a trusted authority allows anybody to broadcast a message. In a threshold public key broadcast encryption scheme (TBE) a message is encrypted and sent to a group of receivers, in such a way that the cooperation of at least t of them (where t is the threshold) is necessary in order to recover the original message. Such schemes have many applications in situations where one wants to avoid that a single party has all the power/responsibility to protect or obtain some critical information. In those schemes, the sender of the message who wants to protect some information may want to decide who will be the designated receivers in an ad-hoc way, just before encrypting the message, and also decide the threshold of receivers which will be necessary to recover the information.

Alderman et al propose a new framework for Publicly Verifiable Outsourced Computation with Access Control (PVC-AC). In the context of VC, we will use cryptographic access control in somewhat unusual ways. Rather than storing static encrypted data and ensuring that only authorized users hold the relevant decryption key, we will encrypt dynamic messages within a protocol execution. In particular, to enforce policies restricting the computations that may be outsourced, a delegator must use an appropriate key to encrypt input data. Without the appropriate encryption, the input will be discarded by the server. The enforcement of policies for performing computations is achieved by distributing keys to servers that can be used to decrypt encrypted inputs. Without decryption, the server will be unable to read the input data and evaluate the function. The enforcement of (read) policies on outputs uses cryptographic access control in a more conventional fashion; results are published and protected via encryption with an appropriate key.

G. Rathi et al propose a device that implements records classification based totally on the sensitivity levels of statistics i.e. for higher touchy statistics higher degree of encryption will be enforced and lower sensitive information will use decrease level of encryption. The machine allows the physician to upload the report after which health practitioner is asked his mystery key wherein the device makes use of this key along with the physician and affected person records to create a device generated key to encrypt the record. To gain pleasant grained and scalable information get entry to manipulate for clinical information saved in cloud servers, recommend Attribute Based Encryption (ABE) strategies including key coverage characteristic primarily based encryption; position based totally encryption, and so on. To encrypt each patient's scientific report file. For this here describe an approach which allows garage that is comfortable and patient's fitness facts with controlled sharing. Explore key-coverage characteristic primarily based encryption to gain patient get admission to manipulate policy such that everybody can down load the records, however most effective legal consumer can view the medical facts. An excessive degree of patient privateness is maintained using a couple of cryptographic algorithms implemented on the diverse styles of statistics. This system has a double layer protection in which the EHRs are stored in the cloud. Encryption/ Decryption will be done in one layer and in the other layer; Splitting/ Merging of the ciphertext will be done. Thus, data security can be improved in cloud computing. As the proposed system is in the development stage, the actual results will be shared in future publication.

III. ARCHITECTURE



In our project we take electronic medical record.First we need to import the medical record in our environment. Next, we do a data encryption which is used to change the readable file into unreadable format for preventing unauthorized access of record.the encyrption process is done by AES algorithm.then decyrption process is done by reverse of aes algorithm.After completing the process we create the duplicate file of original record to store in the duplicate storage.when unauthorized or attacker access the module by unknown credentials redirected to duplicate storage.

IV. METHODOLOGY

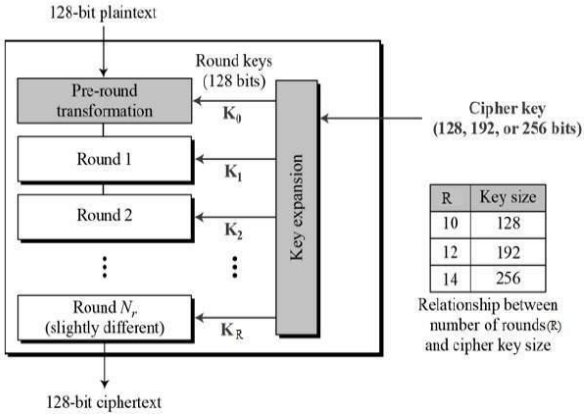
AES Encryption:

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

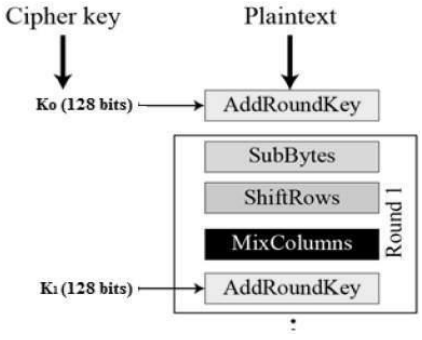


Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes.

Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.



Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.

- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Decryption Process

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order −

- Add round key
- Mix columns
- Shift rows
- Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

V. RESULTS

According to exisiting system,security of medical records in cloud is less than our proposed system.Our proposed system has a duplicate storage make a security of medical records is high.Attackers can able to access a dublicate storage.if get a medical records,it will be dummy file on original file name.

VI. CONCLUSION:

In this project proposed a new mechanism is proposed to protect the healthcare data in the cloud. This system has a double layer protection in which the EHRs are stored in the cloud. Encryption/ Decryption will be done in one layer and in the other layer, duplicate files will be created and stored. To this end, two cloud storages are generated for different purpose. The original medical files are kept secretly in the cloud and the duplicate cloud is used as duplicate file storage. Therefore, instead of retrieving the duplicate medical files only when any unauthorized access is discovered, the user, by default, accesses the duplicate files in cloud 2. The original server is only accessible by a user after verifying the authenticity of the user. Thus, the original multimedia data become more secure by setting the default value of the duplicate storage, while the original medical files are kept in a secure hidden cloud.

REFERENCES

[1]. Tiwari, Deepnarayan, and G. R. Gangadharan.(2018)"SecCloudSharing: Secure data sharing in public cloud using ciphertext-policy attribute-based proxy re-encryption with revocation." International Journal of Communication Systems 31, no. 5 : e3494.

[2]. Zhang, Y., Zheng, D., Li, Q., Li, J., & Li, H. (2016). Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing. Security and Communication Networks, 9(16), 3688-3702.

[3]. Alderman, James, Jason Crampton, and Naomi Farley.(2017)"A framework for the cryptographic enforcement of information flow policies." In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, pp. 143-154.

[4]. Li, Jin, Yinghui Zhang, Xiaofeng Chen, and Yang Xiang.(2018) "Secure attribute-based data sharing for resource-limited users in cloud computing." Computers & Security 72 : 1-12.

[5]. Castiglione, Arcangelo, Alfredo De Santis, Barbara Masucci, Francesco Palmieri, Aniello Castiglione, Jin Li, and Xinyi Huang.(2015) "Hierarchical and shared access control." IEEE Transactions on Information Forensics and Security 11, no. 4 : 850-865.

[6]. Alderman, James, Christian Janson, Carlos Cid, and Jason Crampton. (2015)"Access control in publicly verifiable outsourced computation." In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, pp. 657-662.

[7]. G. Rathi, Abinaya. M, Deepika. M, Kavyasri. T,(2015)" Healthcare Data Security in Cloud Computing" International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3.

[8]. R. Josephius Arunkumar , R. Anbuselvi. (2017)"Enhancement of Cloud Computing Security in Health Care Sector ", International Journal of Computer Science and Mobile Computing A Monthly Journal of Computer Science and Information Technology ISSN 2320–088X IMPACT FACTOR: 6.017 IJCSMC, Vol. 6, pg.23 – 31.

[9]. Kushan Shah, Rui, and Ling Liu.(2010) "Security for Healthcare Data on Cloud." In Cloud Computing (CLOUD), IEEE 3rd International Conference on, pp. 268-275.

[10]. Raval, Divya, and Smita Jangale.(2016)"Cloud based Information Security and Privacy in Healthcare." International Journal of Computer Applications (IJCA), ISSN : 0975-8887.