RESEARCH ARTICLE OPEN ACCESS

Bio-Cryptography Based Secured Data Replication Management in Cloud Storage

P. Elango and K.Kuppusamy
Dept.of Computer Science and Engineering, Alagappa University, Karaikudi

ABSTRACT
Cloud computing is new way of economical and efficient storage. The single data mart storage system is a less secure because data remain under a single data mart. This can lead to data loss due to different causes like hacking, server failure etc. If an attacker chooses to attack a specific client, then he can aim at a fixed cloud provider, try to have access to the client's information. This makes an easy job of the attackers, both inside and outside attackers get the benefit of using data mining to a great extent. Inside attackers refer to malicious employees at a cloud provider. Thus single data mart storage architecture is the biggest security threat concerning data mining on cloud, so in this paper present the secure replication approach that encrypt based on biocrypt and replicate the data in distributed data mart storage system. This approach involves the encryption, replication and storage of data.

Introduction

Providing robust data to users is an important and difficult task for outsourced data providers. Cloud computing is revolutionizing many of our ecosystems, including healthcare. Compared with earlier methods of processing data, cloud computing environments provide significant benefits, such as the availability of automated tools to assemble, connect, configure and reconfigure virtualized resources on demand. These make it much easier to meet organizational goals as organizations can easily deploy cloud services. However, the shift in paradigm that accompanies the adoption of cloud computing is increasingly giving rise to security and privacy considerations relating to facets of cloud computing such as multi-tenancy, trust, loss of control and accountability [1].

Cloud Storage

Cloud storage is defined as the storage of data online in the cloud, wherein a company's data is stored in and accessible from multiple distributed and connected resources that comprise a cloud. Cloud storage can provide the benefits of greater accessibility and reliability, rapid deployment,

strong protection for data backup, archival and disaster recovery purposes; and lower overall storage costs as a result of not having to purchase, manage and maintain expensive hardware. There are many benefits to using cloud storage, however, cloud storage does have the potential for security and compliance concerns that are not associated with traditional storage systems. Representation of cloud storage environment is shown in the figure 1.

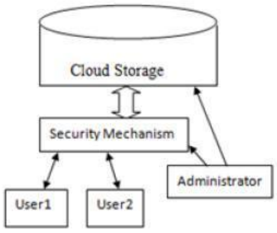

Figure 1 Cloud Storage Mechanism

Data Replication in Distributed storage

For avoiding the disadvantage of storing all data of a client to the single data mart, data can be split into chunks and distributed among multiple data marts. In a distributed environment, an attacker chooses a specific client but the distribution of data into multiple data marts, this makes attacker job more difficult. In purposed system, the distributed environment represents by different data marts that placed on different places. Different data marts store the client's information after encryption, replication. Data warehouse also stores the full copy of client's information for increasing the availability of information. If any data mart lost the part of client's information due to hardware and software failure then it can copy of information from backup data warehouse.

Related Work

Juels et al. [3] described a formal "proof of retrievability" (POR) model for ensuring the remote data integrity. Their scheme combines spot-cheking and error-correcting code to ensure both possession and retrievability of files on archive service systems. Shacham et al. [4] built on this model and constructed a random linear function based homomorphic authenticator which enables unlimited number of queries and requires less communication overhead. Bowers et al. [5] proposed an improved framework for POR protocols that generalizes both Juels and Shacham's work. Later in their subsequent work, they extended POR model to distributed systems. However, all these schemes are focusing on static data. The effectiveness of their schemes rests primarily on the preprocessing steps that the user conducts before outsourcing the data file F. Any change to the contents of F, even few bits, must propagate through the error-correcting code, thus introducing significant computation and communication complexity. Ateniese et al. [6] defined the "provable data possession" (PDP) model for ensuring possession of file on untrusted storages. Their scheme utilized public key based homomorphic tags for auditing the data file, thus providing public verifiability. However, their scheme requires sufficient computation overhead that can be expensive for an entire file. In their subsequent work they escribed a PDP scheme that uses only symmetric key cryptography. This method has lower-overhead than their previous scheme and allows for block updates, deletions and appends to the stored file, which has also been supported in our work. However, their scheme focuses on single server scenario and does not address small data corruptions, leaving both the distributed scenario and data error recovery issue unexplored. Awerbuch and Curtmola et al. [2, 7] aimed to ensure data possession of multiple replicas across the distributed storage system. They extended the PDP scheme to cover multiple replicas without encoding each replica separately, providing guarantee that multiple copies of data are actually maintained.

Proposed work

The system has four components (a) Data Owner: The data owner is responsible for originating the file data to be stored on the cloud. It can be a user-level program, a file system on a personal computer, a mobile device or plug-in of a client information. Firstly, the data owner will request to the key manager for decryption of a blinded version of the encrypted data key. Then key manager will attempt policy Check. If the associated policy is satisfied, the key manager will decrypt the data key and return the blinded version of the original data key. Finally data owner will recover the data key from blinded version. Hence the content of actual data key remains unknown to the key manager and to any attacker also. (b) Key Manager : The key manager is responsible for maintaining policy-based control keys used in the encryption of data keys. Upon request of data owner, key manager performs encryption, decryption, renewal and revocation of the control key. We can deploy key manager as minimally trusted third-party service. Minimally trusted means, the key manager will reliably remove the control keys of revoked policies. Here the files associated with revoked policies will remain inaccessible because the control keys are removed. In this sway file assured deletion is achieved. (c) Storage Cloud: Storage cloud is responsible for storing and maintaining data on behalf of data owner. It is maintained by third-party cloud providers. Here there is no necessity of any protocol, hardware and implementation changes on the storage cloud to support this system. (d) Policy Revocation: The policy revocation operations do not involve interactions with the cloud. Suppose policy Pi is revoked by the data owner. Then in such a case, the key manager will completely erase the private control key di and secret prime numbers pi and qi . So Si cannot be recovered by using Si ei. As a result data key K and hence file F cannot be recovered. Thus we can say that the file associated with policy Pi is assuredly deleted. In such a way, policy revocation operation doesn't include interaction with the cloud

The following figure 2 shows the overall working architecture of the proposed biocryptogrphy based secured data replication management system in cloud storage.

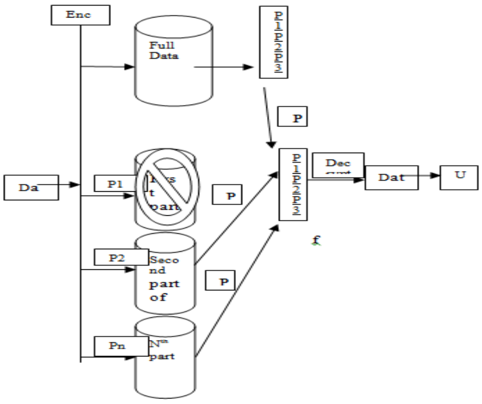

Figure 2 Biocryptography based data replication system architecture

In this system client send data to cloud provider for storing it. The cloud provides receive data from client and perform advance encryption on it. After performing encryption full copy of data stores on data warehouse for backup. After full backup, performing replication divide the data in parts according to the availability of data marts. In purposed system use three data marts (S1, S2, S3) for increasing privacy and availability of client's data. The client's data store on backup warehouse and then divide the data in three parts P1, P2, P3 and store on respective data marts S1, S2, S3. If any data mart lost the part of client's data then it can reload from backup warehouse. In this way replication of client's data on different data mart increase the availability of information as well as enhance the security of information. This makes difficult job of the attackers, both inside and outside attackers. The insider attacker refers as employee that works under organization which is responsible securing and storing the client's information. If any data mart hack by an attacker then it can access the only part of information, for full information there is need to apply attacks on other data marts. The data mart is crashes or down also impact on the availability of information. The purposed system also removes that drawback. If any data mart is crashes or down then client's request also able to extract the data from backup warehouse. In this scenario data mart S1 is fail and not responding the user request. In this case the part of information P1 is lost. The purposed system allow user to extract the information from backup ware house. The availability of data mart also affect on security of information. In case of large no of data marts the data divide in more parts and store different parts in different data marts. Each data marts have very small part of information. If any data mart is hacked by attacker then it can take only small part of information.

Key generation using Biometric characteristics

The following figure 3 shows the key generation process in the proposed work. Let sender A and receiver B both have a common secret key (they have generated secret key using similar technique of Diffie Hellman method). In this proposed algorithm, we will exchange finger print image of A and B by encrypting with secret key. The algorithm leaves no room for false image transmission by third party. After exchange of images, sender A will decrypt the finger print image of receiver B and then will merge finger print of A with finger print of B. After that he will calculate hash on combined image using hash algorithms which is negotiated between sender and receiver. A 128 bit key is generated out of this key generation process.

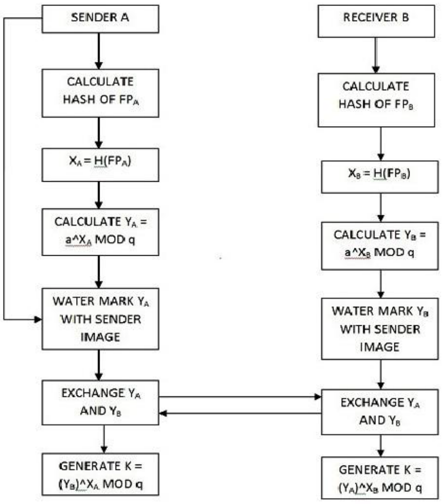
SENDER A
RECEIVER B
CALCULATE HASH OF FP_A
CALCULATE HASH OF FP_B
X_A = H(FP_A)
X_B = H(FP_B)
CALCULATE Y_A = a^X_A MOD q
CALCULATE Y_B = a^X_B MOD q
WATER MARK Y_A WITH SENDER IMAGE
WATER MARK Y_B WITH SENDER IMAGE
EXCHANGE Y_A AND Y_B
EXCHANGE Y_A AND Y_B
GENERATE K = (Y_B)^X_A MOD q
GENERATE K = (Y_A)^X_B MOD q

Figure 3 Overview of key generation

This generated key will be used to generate a random sequence of keys using the pseudo random generator. A random key according to the length of the message is obtained after randomizing all the keys in the key space generated. These random keys are used to encrypt the message of arbitrary length. Sender will watermark the 128 bit key and a random sequence (seed value + other parameter) in the sender finger print image. Sender sends watermark image and encrypted message to the receiver. Similarly, receiver B generates a 128 bit key by applying a hash on combined image of finger print of sender A and receiver B. Now, receiver will be de watermark the received watermark image from the sender and get 128 bit key + a random sequence + FPA (finger print of A). Receiver pick 128 bit key and compare it with the generated 128 bit key. If it is same, then he can be assured that the key has not been altered in the way and authenticate the user also. After that he will pick random sequence and apply on the 128 bit key and generate a random sequence of keys, by which he will decrypt the message that was encrypted earlier by sender

Cloud implementing Biometric Cryptography

The following figure 4 shows the implementation of Biometric Cryptography in cloud environment.
Sender A encrypts the message by using secret key and sends this message to receiver. Receiver also encrypts finger print by using secret key and sends to sender.

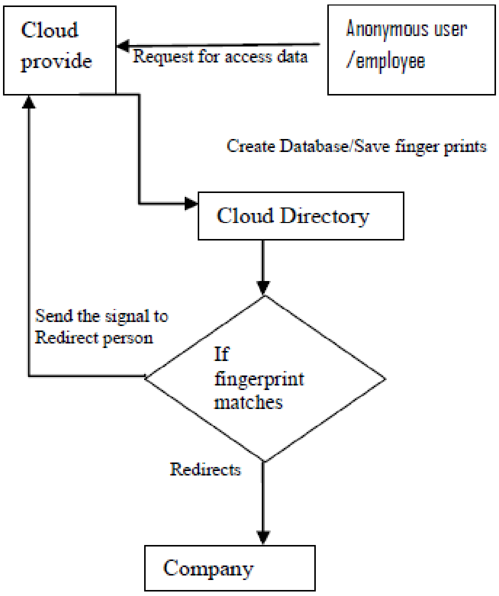

Figure 4 Biometric Cryptography implementation in Cloud

Now, the sender decrypts the finger print of B and merges it with its own finger print, calculate hash on partial portion of merged image which generate a 128 bit key. After taking hash on partial portion, Random keys are generated by this master key using random sequence (seed value + other parameter). A pseudo random generator usually extends to the solution space of hash values or it may even consider the pattern space of finger print to randomize the pattern itself. Now, we use these random keys to encrypt the message of arbitrary length. Calculated hash on master key and random sequence (seed value + other parameter) provides the requisite information of this algorithm. After calculating all these values, sender watermarks(master key + random sequence) and hash of (master key + random sequence). Sender A sends watermarked image and encrypted message to receiver B. At receiver side, after dewater marking receiver, finds 128 bit key, random sequence, image of finger print of A and hash of (128bit key and random sequence). Receiver apply hash on dewater marked random sequence and master key and compare it with the received hash value for checking the integrity of the message. The sender authentication gets executed when receiver compare dewater marked sender image by previous image. After this receiver use pseudo random generator to generate keys and message is obtained after decryption.

RESULT AND DISCUSSIONS

The following Table and Figure 5 shows the comparison of the existing approach with the proposed approach data replication using biocryptography based security with the failure rate of 20%, sensitivity of 92.60% and specificity of 93.20%.

	Data Replication without Security	**Data Replication using BioCryptography based security**
Failure Rate	60%	20%
Sensitivity	75.30%	92.60%
Specificity	80.70%	93.20%

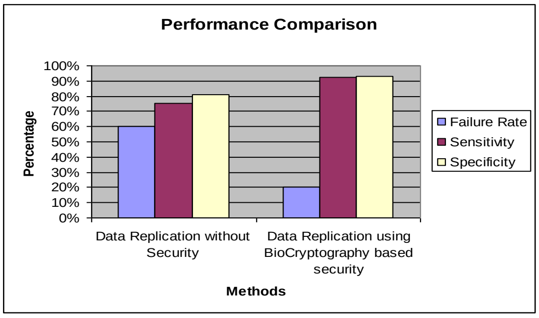

Figure 5 Comparison of Data Replication without security approach and the Biocryptography based security approach

Conclusion

In this paper, the general principle of new approach to perform secure replication on stored information is outlined. This is a dominant technique which will provide better results for security and availability of information. This secure replication technique can be used in order to build a secure and reliable distributed storage. We expect the enhancement done in this technique will increase the quality by different data mart host with cloud provider and store information according to its sensitivity. This new technique that we have developed can be applicable in different cloud providers companies and financial organizations etc.

References:

1. Buyya, R., et al., Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst., 2009. 25(6): p. 599-616

2. B. Awerbuch, Y. Bartal and A. Fiat, “Optimally-Competitive Distributed File allocation”, 25th Annual ACM STOC, Victoria, B.C., Canada, 1993, pp. 164-173

3. A. Juels and B.S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS’07), pp. 584-597, 2007.

4. H. Shacham and B. Waters, “Compact Proofs of Retrievability,” Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT’08), pp. 90-107, 2008.

5. Bowers K D, Juels A, Oprea A. Hail: A high-availability and integrity layer for cloud storage. In Proceeding of ACM Conference on Computer and Communications Security. Chicago: ACM, 2009. 187–198

6. Ateniese G, Burns R C, Curtmola R, et al. Provable data possession at untrusted stores. In Proceedings of the 2007ACM Conference on Computer and Communications Security. Alexandria: ACM, 2007. 598-609.

7. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MR-PDP: Multiple-Replica Provable Data Possession,” in 28th IEEE ICDCS, 2008, pp. 411–420.