High security mechanism: fragmentation and replication in the cloud with auto update in the system

Shrutika Khobragade, Rohini Bhosale, Rahul Jiwane
Department of Computer Engineering, Mumbai University, Pillai HOC College of Engineering and Technology, India

Article Info

Article history:

Received Jan 22, 2020
Revised May 10, 2020
Accepted May 25, 2020

Keywords:

Auto update mechanism
Cloud security
File fragmentation
File replication
Version control

ABSTRACT

Cloud computing makes immense use of internet to store a huge amount of data. Cloud computing provides high quality service with low cost and scalability with less requirement of hardware and software management. Security plays a vital role in cloud as data is handled by third party hence security is the biggest concern to matter. This proposed mechanism focuses on the security issues on the cloud. As the file is stored at a particular location which might get affected due to attack and will lost the data. So, in this proposed work instead of storing a complete file at a particular location, the file is divided into fragments and each fragment is stored at various locations. Fragments are more secured by providing the hash key to each fragment. This mechanism will not reveal all the information regarding a particular file even after successful attack. Here, the replication of fragments is also generated with strong authentication process using key generation. The auto update of a fragment or any file is also done here. The concept of auto update of files is done where a file or a fragment can be updated online. Instead of downloading the whole file, a fragment can be downloaded to update. More time is saved using this methodology.

Corresponding Author:

Rahul Jiwane
Department of Computer Engineering
Mumbai University
Pillai HOC College of Engineering and Technology, Rasayani, Dist. Raigad, Maharashtra, India
Email: rjiwane@mes.ac.in

1. INTRODUCTION

Cloud computing encloses more use of networking sites and other forms of interpersonal computing. However, there are a large amount of resources on cloud storage, data or software applications which have been accessed online. It plays an important role in the privacy and security of the data. As cloud computing is a flexible, cost- effective and authenticated delivery platform for providing business consumer IT services on the internet. Cloud computing presents an added level of risk as essential services are often outsourced to a third party, which makes it difficult to maintain data security and privacy, demonstrate consent and also support data and service availability. The cloud computing paradigm has reformed the control and management of the information technology infrastructure [1]. Cloud computing is characterized by on-demand self-services, resource pooling, elasticity, ubiquitous network accesses and measured assurance of the services [2], [3]. However, the benefits of imperceptible management (from user's perspective), low cost, easy access and greater resilience come with increased security concerns which have to be taken care of.

Erstwhile, computer software was not written with security in mind but because of the increasing frequency and sophistication of malicious attacks against information systems, modern software design methodologies include security as a primary objective. With cloud computing systems seeking to meet multiple objectives, such as cost, performance, reliability, maintainability, and security, trade-offs have to be made. Any cloud server is vulnerable to an attacker with unlimited time and physical access to the server. Additionally, physical problems could cause the server to have down time. This would be a loss of availability, which is one of the key principles of the security triad confidentiality, integrity, and availability (CIA).

The data which is outsourced to a public cloud must be secured. Unauthorized data access by other users and different processes (can be accidental or deliberate) should be prevented [4]. As stated above, any weak entity may lead the whole cloud at risk. In such a framework, the security mechanism should significantly increase a hacker's effort to retrieve a probable amount of data which may get lost even after a successful intrusion or attack in the cloud. Moreover, the reasonable amount of loss (due to data leakage) must also be minimized. Mei *et al.* [5] says that the scheme of fragmentation and replication to allocate files over multiple servers can lead to any attack as the files will be stored at the particular location. Juels and Opera [6] presented a technique called as Iris file system which ensures the freshness, integrity and availability of data in a cloud. Kappes *et al.* [7] where the deliberate attack of censorious information in case of improper sanitization cannot be handled. It stores the file based on blocks which may lead to an improper sanitization.

The outsourced environment where the use of a trusted third party provides the security services in the cloud is advocated in addressing cloud computing security issues [2]. They used the public key infrastructure (PKI) to enhance the trustworthiness in the authentication, integrity and confidentiality of data. At the user level, the use of tamper-proof devices, such as smart cards was used as the storage of the keys. Tang *et al.* [8] have utilized the public key cryptography and trusted third party for providing data security in cloud environments. However, the author has not used the PKI infrastructure to reduce the overheads. The system needs to be more secure and should be accessed by only authorized person. The data stored in the cloud need to secure as well as proper encryption keys have to be used for the verification of file which is stored in it.

2. PROPOSED SYSTEM MODEL

A new proposed model ensures the security of the data which is stored on the cloud. This system provides the better solution to increase the security as well as performance level. The cloud security increases by the control of third-party administrative control. The data which needs to be secured is in the form of files. Diverse amount of files are stored on cloud so here a particular file is uploaded and then fragmentation process is done. Each fragment of that file is secured with hash key which is generated randomly. Fragments are generated based on equal size and each fragment of that particular file is placed at a different location. Here controlled replication is maintained where each fragment is replicated only once to improve the security. When an attacker/ hacker hacks that specified file, he will not reveal all the information of that file. Various attacks such as data recovery, cross VM attack, improper media sanitization, VM escape can be handled by this methodology. In this proposed system, a file is uploaded on the cloud which needs to be secured as it is third party outsourced data. A file is generally stored on a cloud at a particular location which is not secured as any attacker can easily access or attack that particular file and get the information using various malicious attack or intrusion. So instead of storing a single file at a particular location, a file is fragmented according to the size and placed at the different location so that if any attacker access to that particular file it will not reveal all the information of that whole file. Security is the major concern where a file must be purely secured, so to maintain this, each key is used for different fragment of a single file. Replica of file is also maintained to get availability of lost file or old files.

2.1. Design goals of cloud computing

Design goals of cloud computing are stated to provides good authentication system which allows only authorized users to login and process. Improves security as well as improves the performance. Fragmentation of file is created to and each fragment is stored at different location. Controlled replication is developed to decrease the chance of data loss, increases the performance, availability and reliability. To provide a file to the client whenever there will be run time error in network.

As illustrated in Figure 1, in the design, first registration process is done by giving the information about the user. Authorized user will login to the system. Proposed system provides best way for the secured files which is very susceptible for attacks. User needs to upload the file on the system by providing file id and file name. The uploaded file gets fragmented in such a way that fragments do not include any meaningful information as a particular if fragmented and each fragment needs its own encryption key. Fragmentation is done by using the equal size of file and if not of equal size then remaining bytes of file is stored at the next fragment. This helps to keep away the attacker from finding the location of each fragment as each fragment is stored at different location. Here each key is generated for each fragment from the content of the file so that an attacker will not get the data of each fragment. Fragments will be placed at different nodes as it will be difficult for the attacker to access on single file. After the fragmentation process that fragments get replicated on different nodes in such a way that the access time will be low which also increases the performance. Proposed system provides the controlled replication which is required to manage the ideal performance and more security.

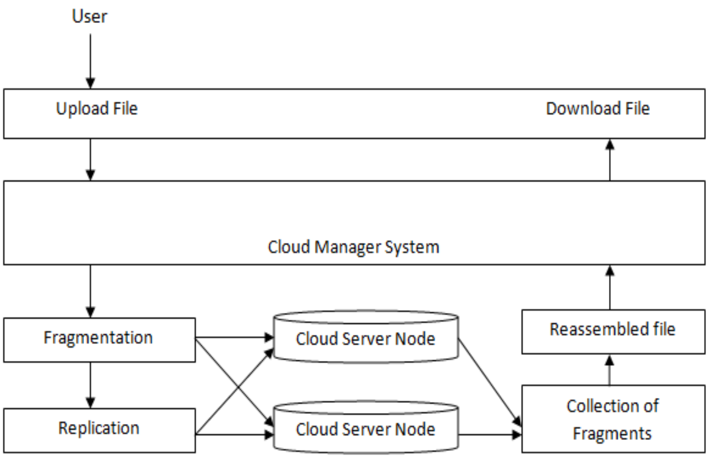

Figure 1. Proposed architecture

3. PROPOSED SYSTEM TECHNIQUE

The steps of algorithm are as follows:

3.1. Fragment placement

The initial step is select the file foe fragment. Select the particular file which needs to be fragmented. The file gets fragmented based on the size. Each fragment is stored at the different location i.e. on different nodes. Repeat the process until all fragments assign to the node. File size should be more than 20 KB.

3.2. General flow algorithm

S = {I, P, R, O}

Where,

I is set of initial input to the system.

I = {i1, i2, i3}

i1 = File given by the user.

i2 = Download request from user.

i3 = Download request from client.

P is set of procedure or function or processes or methods.

P = {p1, p2, p3, p4, p5, p6, p7, p8}

p1 = Registration and authentication.

p2 = Uploading a file on cloud server.

p3 = Fragmentation of file with separate hash key for each fragment received from user.

p4 = Replication of that file.

p5 = Download request from user.

p6 = Download request from client.

p7 = Collection and reassemble of fragments.

p8 = Downloading the original file.

R is a set of rules or constraints.

R= {r1}

r1 = File accessed from Replication.

O is a set of outputs.

O = {o1}

o1 = Downloading the original file.

3.3. Use of rijndael algorithm

Rijndael encryption algorithm proposes a new encryption technique for encryption and decryption purpose. It is advanced AES algorithm is used to encrypt sensitive information. It is symmetric key encryption algorithm to be used to encrypt information. It is the best combination of security, performance, efficiency, easy implementation and flexibility, high speed and versatility across a variety of platforms. Run efficiently on large computers, desktops and small devices like smart cards. The block and key can in fact be chosen independently from 128, 160, 192, 224, 256 bits. Rijndael is simple to implement and uses very little system memory.

3.4. Use of SHA-512 algorithm

We are using SHA-512 algorithm for performing hashing task. Secure hash algorithm (SHA). The United States national security agency is designed hashing algorithm.SHA-512 is faster than SHA-256 on 64-bit machines is that has less rounds per byte (80 rounds for 128-byte blocks) compared to SHA-256 (64 rounds for 64 byte blocks), However, storing a SHA-512 bit hash is expensive. The SHA-512 time to generate the hash value and the number of cycles per bytes are efficient comparing to the others. In SHA-512, the number of cycles per bytes somewhat more compared to other hashing functions, but at the same time the time to generate the hashing value is much smaller than others. So the SHA-512 hash function is efficient and also secure hashing algorithm.

3.5. Use of random key generation algorithm

Random Key generation generates random values. It is used for encryption and decryption purpose. Here the keys are randomly generated using a random number of generator or pseudorandom number generator that produces random data. In our system random number key generation is used at the time of downloading data for the user. It makes much harder for a hacker or attacker to guess the key.

3.6. Properties of proposed system

Properties of proposed system are follows:

- Uses the one time password verification scheme.
- Privacy of data is maintained from third party organization.
- Files are divided into fragments and stored at the various locations.
- Key generation is done for each fragment.
- Best encryption algorithm used.
- Version control is used for retrieving old files.
- Auto update and auto replicate of files are done.
- Controlled replication for immense security.
- Implementation of real cloud.

There are different modules of system as shown in Figure 2. They are as follows:

Upload: This module consist of uploading the file on the cloud. When the file is uploaded, a file id is generated for each new file. Then the name to that file can be given by the user which is unique.

Edit: This module is for editing of the particular fragment of a file. The details of all the uploaded files are shown in this table. When any file is uploaded it is divided into various fragments and encrypted and later stored in various nodes. And each fragment has its hash value which is sent to the users email id during registration. To edit we need to select any fragment and edit that file, but to retrieve that content hash value has to be entered.

Edit full file: This module consist of the update of whole file. Here we can upload the updated file and rename the file as new file. The old file is stored in as a backup.

Download: This module is used for downloading the file which is in the encrypted form. A single fragment can be downloaded or a single fragment too. When we download the data using authorized key it is first decrypted and then the fragments are combined from various nodes used and the file is presented in its original form. This is the procedure which is implemented to upload the file then fragmentation, replication and download of original file is done.

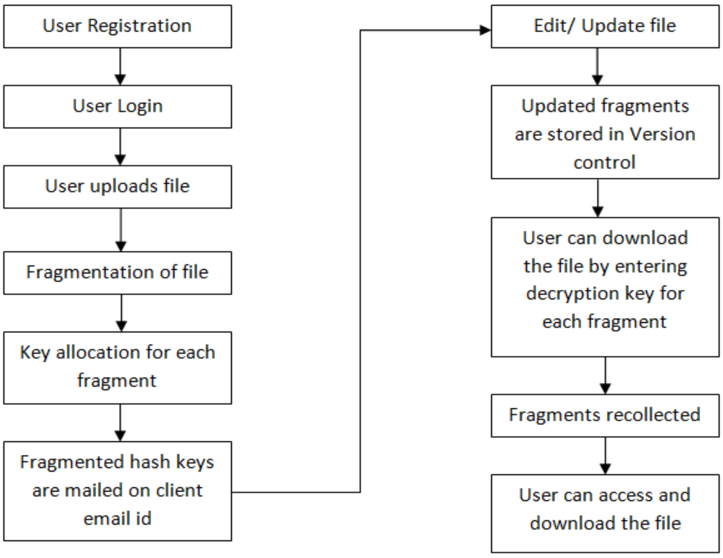
User Registration
User Login
User uploads file
Fragmentation of file
Key allocation for each fragment
Fragmented hash keys are mailed on client email id
Edit/ Update file
Updated fragments are stored in Version control
User can download the file by entering decryption key for each fragment
Fragments recollected
User can access and download the file

Figure 2. Proposed system flow

4. RESULTS AND DISCUSSION

The Figure 3 shows the gap analysis between existing and proposed system. The existing system gives the file to the client whenever request is granted. In the existing system there was no traditional cryptographic technique used for security of the data and no recovery of data is maintained.

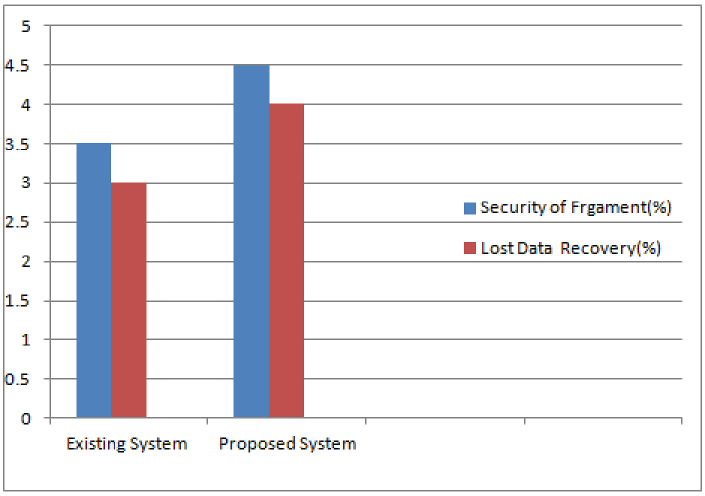

Figure 3. Result analysis

So, details of Figure 3 are as follows:

- Security of fragment: In the existing system there was no cryptographic technique used but in this proposed system special cryptographic technique is used to increase the security of data for each fragment.
- Lost data recovery: In the existing system, if the replica of a fragment is deleted, there was no recovery system but in this proposed system even if any fragment is deleted, we can retrieve using version control.

As from the above details there are two parameters which differ from the existing system.

Table 1. Comparison of two systems

Sr. No.	Existing System	Proposed System
1	Only the data is secured.	Security is given to data as well as traditional cryptographic technique is used.
2	Even if the file or fragment is lost or whole block is lost, it cannot be recovered.	Recovery of file or fragment is done in version control.
3	Download of whole file and update is done. As it is time consuming.	Instead of downloading whole file we can update a particular fragment online itself or we can download required fragment only to update it. This saves time.
4	No separate hash key is used for fragments.	Separate key is maintained for each fragment to make it more secure.

5. CONCLUSION

A system is proposed for security of users' data when the data is stored in the cloud. The proposed scheme works for the security of users' data when data is stored into the cloud. Cloud computing growth raises the security concern due to its core technology. So, this system provides a better solution to achieve the security as well as performance by using techniques such as secure cryptographic scheme, fragmentation and replication. Fragmentation is used to protect data from single point disaster. Replication can be useful for maintaining availability, reliability and performance in failure situations. But the extra replication can also result in high storage cost or drops in systems overall performance due to extreme use of bandwidth. So, here controlled replication is used. This scheme utilizes rijndael algorithm to create an encryption key that the user gets while requesting to data owner to file accessing. This scheme uses SHA-512 algorithm for generating hash key for each fragment after division.

REFERENCES

[1] K. Hashizume, D. G. Rosado, E. Fernandez-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing, "*J. Internet Serv. Appl.*, vol. 4, no. 1, pp. 1-13, 2013, doi: 10.1186/1869-0238-4-5

[2] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583-592, 2012, doi: 10.1016/j.future.2010.12.006.

[3] M. Hogan, F. Liu, A. Sokol, and J. Tong, "NIST cloud computing standards roadmap," *NIST Special Publication*, 2011, [Online] Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=909024.

[4] N. Khan, M. L. M. Kiah, S. U. Khan, and S. A. Madani, "Towards Secure Mobile Cloud Computing: A Survey," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278-1299, 2013, doi: 10.1016/j.future.2012.08.003.

[5] A. Mei, L. V. Mancini and S. Jajodia, "Secure dynamic fragment and replica allocation in large-scale distributed file systems," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 9, pp. 885-896, Sept. 2003, doi: 10.1109/TPDS.2003.1233711.

[6] A. Juels and A. Opera, "New Approaches to Security and Availability for Cloud Data," *Communications of ACM*, Vol. 56, No. 2, pp. 64-73, 2013, doi: 10.1145/2408776.2408793.

[7] G. Kappes, A. Hatzieleftheriou and S. V. Anastasiadis, "Virtualization-aware access control for multitenant filesystems," *2014 30th Symposium on Mass Storage Systems and Technologies (MSST)*, 2014, pp. 1-6, doi: 10.1109/MSST.2014.6855543.

[8] Y. Tang, P. P. C. Lee, J. C. S. Lui and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion," in *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 903-916, Nov.-Dec. 2012, doi: 10.1109/TDSC.2012.49.