Proficient Public Attestation of Data Veracity for Storage of Cloud Across Dual Protection

T Ramya
Computer Science and Engineering, ARM College of Engineering and Technology, Chennai, India.

Abstract

The data privacy and service availability in cloud computing are the key security issue. In this project we preserve our data in the cloud storage and would make available for multiple users upon various effective verifications done by auditor and the data can be stored in the cloud and updated dynamically. When a data is uploaded to the server the original data splits into three parts and stored in three different locations on the server using Erasure Code algorithm, a unique key code is generated to each split data for auditing. After key generation all the three parts of the single file will be encrypted by AES (Advanced Encryption Standard) algorithm. Each users are given some level of access as per the requirement (View , Read only , Read write) To download the stored file Owner authorization is essential. During the time of download a key is generated(by using code based key generation) and will be sent to file owner , the encrypted data will be decrypted and spliced from three dissimilar locations, If anyone tries to hack the data at cloud end it is impossible to break the two different blocks as the security scheme of our system is Strongest and this system allows the clients to check and monitor whether their outsourced data is remain pristine without downloading the whole data using own auditing based Token generation the system ensures more security , when a data is edited or modified the anonymous change will be made in the original key value, by this client can easily find out the data is being changed or modified.

I. INTRODUCTION

Distributed computing has been visualized as the accompanying creation information development (IT) plan for endeavors, due to its broad summary of unparalleled inclinations in the IT history: on-ask for self-advantage, ineliminable framework to get the zone of self-choosing resource pooling, quick resource adaptability, usage based assessing and transference of peril.

While distributed computing make these compensation more captivating than some other time in ongoing memory, it also passes on new and testing security risks to customers' utilized data. As Cloud organization providers are doing the administration for the components and thecustomers data's are outsourcing truly last control more than the fate of their data.

One fundamental piece of this standpoint changing is that data are being outsourced. From customers' perspective, including together individuals and IT, securing the data remotely to the in a adaptable on-ask for strategy bring engaging focal points: landing of the weight for storage space organization, big data access and avoidance of advantages staff frameworks of help,programming and hardware costs. As an issue of first implication, despite of the way that the structures beside the more powerful and trustworthy than individual enlisting devices, they are still before the broad assortment of both inside and outside risks for data respectability.

1.1. Scope of The Project

As rapid systems and omnipresent Internet get to wind up accessible as of late, numerous administrations are given on the Internet to such an extent that clients can utilize them from anyplace whenever. Information vigor is a noteworthy prerequisite for capacity frameworks. There have been numerous proposition of putting away information over.

1.2. Objectiveof The Project

Specialist organizations are separate authoritative element, information outsourcing is really giving up client's definitive control over the destiny of their information. Thus, the rightness of the information in the is being put in danger because of the accompanying reasons. Above all else, despite the fact that the foundations under the are substantially more ground-breaking and solid than individualized computing gadgets, they are as yet confronting the wide scope of both inward and outside dangers for information respectability.

1.3. Acronyms and Abbreviations

Abbreviations such as AES, DML, J2EE, JSP, ASCII, EC, PKI, EC, TPA, and SQL

II. CONTENT OF THE THESIS

2.1. User Plug in

In our system provides a user friendly interface to interact with our system. We have user management module to authenticate the valid user and make the system is secure. User can have create the account and our system provide self-dashboard that contains the options for file upload, download and check the user activities. Our System allows only authentic user to upload and download the files. In our system Provides easy environment to access the file storage with secure manner.

Our System provides the easy way to search the file, so the user don't want to remember the exact file name of all the uploading files, we have provide the keyword for each file while upload the file.

2.2. File uploading

Storage server provide data security to a message such that each server stores a message when storing the data. By using Erasure coding technique each message have to encode of k symbols into a code word on n symbols.

A message code word symbol has to store in different server. Storage server that corresponds to an error of erasure code word symbol. The message can be retrieved from the code word which is available in the storage servers by decoding

2.3. Secret key generation

Initially the secret key will be generated at the time of file uploading, each file is uploaded, will have unique secret key. This key will be used to identify every file. The key which we are using is a three digit number and we will make it use for both uploading and downloading process.

If the user wants to download the file they gives the download request and the secret key of the file will be sent to the owner of the file. Once they get the request maybe they can share the file.

2.4. File Sharing

In our system we can share the file to authorized registered user by providing credentials, we can enable the sharing option the necessary authority provides the permission to the shared user whether they can edit or view the particular file. User can view the shared file with in the system without downloading the file and also they can edit the file which will not affect the original file.

2.5. File Auditing

File auditing the process of check the file activity whether the files original content was changed or not. This will provides the information to file owner by generating the token, where the token are generated by using the ASCII valuesof the characters in the file and these characters are stored in the Database while uploading the file. If a shared user edit the file and saves it, during this process a new token will be generated and stored in the Database. If the initial token and the current token aren't the same then a notification will be sent to the file owner.

2.6. Mail alert process

The secret key has been sent to the corresponding user Email when uploading and downloading process of the user, the user gets the secret key and then apply the secret key to encrypted content. After the encryption this can be send to the storage server and decrypt it by using the same key to download the corresponding data file in the storage server. The secret key generation using the share key gen algorithm (SKA, t, m).This algorithm share the SKA (secret key) of the user to set of key server

2.7. File Downloading process

File downloading is the process of get the corresponding secret key from the user email of the file. After that we can decrypt the file data. During the process of downloading again new encryption key generated and stores in servers such that storage servers perform the encryption Operation. The length of forwarded message and the computation of new encryption are taken care of by storage servers. Proxy Schemes significantly reduce the overhead of the data forwarding function in a secure storage system.

III. TECHNOLOGY IMPLEMENTATION

Erasure Coding:Erasure coding (EC) is a data protection and storage process through which a data object is separated into smaller components/fragments and each of those fragments is encoded with redundant data padding. EC transforms data object fragments into larger fragments and uses the primary data object identifier to recover each fragment.

EC is primarily derived from a mathematical equation: n= k + m

Where
"k" = the original data
"m" = the additional data padding
"n" = the resulting erasure coded data

This same equation can be applied to the data to recover the original amount of data. Erasure coding is primarily used in applications that have a low tolerance for data errors. This includes most data backup services and technologies including disk arrays, object-based cloud storage, archival storage and distributed data applications.

Erasure coding performs best in cases of sequential data writes. The erasure coding engine immediately writes the original data to remote disks as the data streams in. It computes the coding parts on the fly and writes them along. If written in random order, write performance for data degrades severely. The coding engine needs to read all data of the coding group first, recomputed the coding parts, and then write out the modified original data along with the coding data. That amplifies any random write by m – 1 reads and $m$writes. Because erasure coding stores the original data as-is, reading erasure coded data behaves just like reading replicated data. And it has no caveats around access patterns.

All this boils down to one essential rule: For sequentially written data (usually written only by one writer and only once) use erasure coding. That'll give you the benefit of increased data safety and minimal blow-up. For everything else use replication. Quobyte's policy engine lets you choose between the two down to the level of individual files.

Algorithm

Step 1: Storing data in a third party's cloud system causes serious concern over data confidentiality.

Step 2: Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority.

Step 3: A threshold proxy re-encryption scheme and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated.

Step 4: The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back.

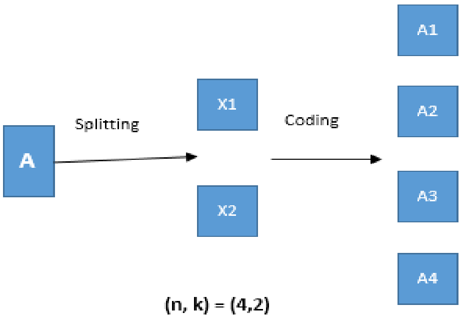
Step 5: The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages.

Step 6: It’s stored the number of copies of a message dispatched to storage servers and the number of storage servers queried by a key server.

IV. PSEUDOCODE

```
Begin;
→ow and pwd;
Based: = the privileges based on the entry system in
the cloud computing
ownname =ow&&pwd=password
Then
→If (skey==cfile)
        Files upload i;
i→sp1,sp2,sp3;
→Encryption & decryption with DES
r→encsp1, encsp2, encsp3
w→decsp1, decsp2, decsp3
file downloading  fd;
serfile from db& server
→if(fd==serfile)
Skey→send to user mail(otp).
addori→(sp1+sp2+sp3)
download the file.
      ori;
      Else
   Cancel the file;
End;
```

Without really wondering how do actually add files, the following example illustrates one particular case of designing a (4, 2) code.



AES is based on a design principle known as a substitution–permutation network, and is efficient in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, Rijndael per se is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits.

KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

Initial round key addition:

AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor.

9, 11 or 13 rounds:

SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

MixColumns—a linear mixing operation which operates on the columns of the state, combining the four bytes in each column.

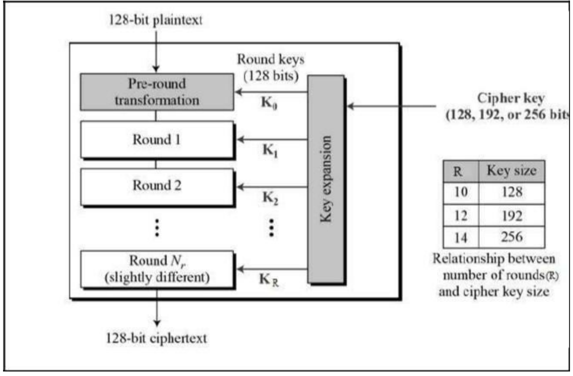
AddRoundKey

Final round (making 10, 12 or 14 rounds in total):

SubBytes

ShiftRows

Addroundkey



R	Key size
10	128
12	192
14	256

Figure 1. The Schematic of AES Structure

4.1. *Encryption Process*

For encryption, you can either enter the plain text or an image file or a .txt file that you want to encrypt. Now choose the block cipher mode of encryption. ECB (Electronic Code Book) is the simplest encryption mode and does not require IV for encryption. The input plain text will be divided into blocks and each block will be encrypted with the key provided and hence identical plain text blocks are encrypted into identical cipher text blocks.CBC mode is highly recommended and it requires IV to make each message unique.

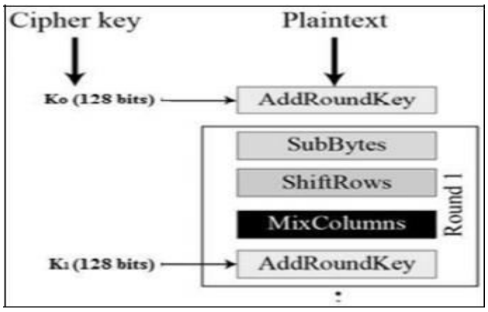


Figure 2. Encryption Process

4.2. *Byte Substitution*

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.

The SubBytes phase of AES involves splitting the input into bytes and passing each through a Substitution Box or S-Box

4.3. *Shift rows*

Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are reinserted on the right side of row. Shift is carried out as follows The result is a new matrixconsisting of the same 16 bytes but shifted with respect to each other.

4.4. 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.

4.5. 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 cipher text. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

4.6. Decryption Process

AES decryption has also the same process. Bydefault, it assumes the entered text be in Base64.The input can be Base64 encoded or Hex encoded image and .txt file too. And the final decrypted output will be Base64 string. If theintended output is a plain-text then, it can be decoded to plain-text in-place.

V. Performance Of Evaluation And Results

A Successful uploading of file to the cloud by splitting it into equal three halves based on its size is done. The files are splits using the erasure code method and successfully encrypted and uploaded to the cloud. The dual protection of data stored over the cloud will be preserved more efficiently than the existing system.

The performance of the existing and proposed system is depicted through the histogram chart.

VI. Conclusion

In addition, the analysis and experimental results show this system is provably secure and efficient. Moreover, unlike the existing schemes, the proposed systemrequires at least *t* group managers to work together to trace the identity of the misbehaving user.

We additionally extend our protection safeguarding open evaluating convention into a multi-client setting, where the TPA can play out different examining errands in a cluster way for better productivity. In imminent we will enhance the execution.

In this framework we utilized just content records; In future we will incorporate the picture, sound, video documents. In our framework the OTP sent to proprietor mail id, coming up the customer will get the OTP on portable by utilizing the versatile number Current auditing protocols are all based on the assumption that the client's secret key for auditing is absolutely secure. However, such assumption may not always be held, due to the possibly weak sense of security and/or low security settings at the client[1]. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud[2]. an auditing framework for cloud storage systems is designed and an efficient and privacy-preserving auditing protocol is proposed. [3]— Remote data integrity checking is of crucial importance in cloud storage. It can make the clients verify whether their outsourced data is kept intact without downloading the whole data [7]."

Provable data possession (PDP) is a probabilistic proof technique for cloud service providers (CSPs) to prove the clients' data integrity without downloading the whole data [6].

A novel public auditing mechanism for the integrity of shared data with efficient user revocation in mind. By utilizing proxy re-signatures" [5]., public auditing on the integrity of shared data with these existing mechanisms will inevitably reveal confidential information identity privacy to public verifiers [4].

VII. References

[1] Jia Yu, Kuiren, Cong Wang, Vijay Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," in Proceedings of SecureComm. ACM, Vol 10 2015, pp. 1167 - 1179.

[2] Qian Wang, Cong Wang, Kuiren, Wenjing Lou And Jin Li, "Enabling public auditability and data dynamics for storage security in cloud computing," EEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847 - 859, 2011.

[3] Cong Wang ; Qian Wang ; Kui Ren ; Wenjing Lou, "Privacy-preserving public auditing for data storage security in cloud computing, 2010 Proceedings IEEE INFOCOMvol. 5, pp. 567–578, 2010.

[4] Boyang Wang, Baochun Li, Hui Li, "ORUTA: Privacy-preserving public auditing for shared data in the cloud," IEEE 5th International Conference, vol. 46, no. 4, 2012.

[5] Boyang Wang, Baochun Li, Hui Li, "Public auditing for shared data with efficient user revocation in the cloud,"in IEEE Transactions on Services Computing,vol 8,, pp. 92-106, 2015.

[6] Huaqun Wang And Yuqing Zhang, "On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 4, pp. 264-267, 2014.

[7] K. Yang and X. Jia, "Identity-based distributed provable data possession in multicloud storage," IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 328–340, 2014.

[8] Kan Yang And Xiaohuajia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," in IEEE Transactions on Parallel and Distributed Systems, vol 24, pp. 1717-1726,2013.