IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Securely Eradicating Duplication by Generating File Tags and Tokens over Hybrid Cloud Using Security Algorithm

Madhuri Ghodke[1], Priyanka Bais[2], Abhirupa Saha[3], Poonam Singh[4], Prof. G.M. Gaikwad[5]

B.E, Information Technology, Sinhgad Institute of Technology, Lonavala, India [1, 2, 3, 4]

Professor, Information Technology, Sinhgad Institute of Technology, Lonavala, India[5]

Abstract: A composition of two or more clouds i.e. private and public that remains distinct entities but are bound together, providing advantages of number of deployment models is called Hybrid Cloud. Eliminating duplicate copies of repeating data to save storage can be managed by data deduplication which is a specialised data compression technique. Existing system shows that each user will be issued private keys for their corresponding privileges. These private keys can be used for generating file token for duplicate checking. However, during file uploading, the user needs to use file tokens for sharing with other users with privileges. To compute these file tokens, the user needs to know the private keys. This restriction leads the authorized deduplication system unable to be widely used and limited. This failure can be overcome by implementing block level deduplication which eliminates duplicate blocks of data that occur in non-identical files. New deduplication algorithms supporting authorized duplicate check in hybrid cloud using token number and privilege key are used such as SHA1 and AES algorithms. SHA-1 produces a 160-bit hash value known as message digest. AES algorithm will be used to convert a given plaintext of 256 bit into cipher text of 256 bits. Thus we will implement a deduplication construction supporting authorized duplicate check in hybrid cloud architecture.

Keywords: Data Owner Module, Encryption and Decryption module, Remote User Module, Cloud Server Module.

I. INTRODUCTION

Hybrid cloud is a consolidation of two or more clouds (private, community or public) that remain distinct entities but is intent together, providing the advantages of number of deployment models. Data deduplication is a specialized data compression scheme for removing duplicate copies of repeating data to save storage. Today's cloud service providers provide us with immense storage and computational facilities at a very low cost. Instead of keeping the redundant data it makes us available some more space that can be used for further storage. To make data management scalable in cloud computing, deduplication has been one of the commonly used technique and has attracted more and more attention recently. Data deduplication is a functional data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to provide better storage exertion and can also be applied to network data transfers to reduce the number of bytes that should be sent. Rather preserving multiple data copies with the same content, deduplication removes extravagant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can also take place at file level or the block level.

For file level deduplication, it eliminates duplicate copies of the clone file. Deduplication can also arise at the block level, which eliminates duplicate blocks of data that be tide in non-identical files. In spite of data deduplication brings a lot of benefits, security and privacy concerns appears as users' sedative data are susceptible to both inside and outsider attacks .Traditional encryption, while sustain data affinity, is incompatible with data deduplication. Specifically, for traditional encryption various users require to encrypt their data along with their own keys. Thus, identical data copies of different users will lead to distinct cipher texts, making deduplication impractical. Convergent encryption has been proposed to enforce data confidentiality while making deduplication appropriate. It encrypts/decrypts a data copy with a convergent key, which is retrieved by measuring the cryptographic hash value of the data copies content. After key generation and data encryption, users possess the keys and post(send) the cipher text to the cloud. Since the encryption operation is deterministic and is borrowed from the data content, identic data copies will generate the same convergent key and hence the same cipher text. To inhibit unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user certainly owns the equivalent file when a duplicate is found. After the proof, subsequent users with the same file will be dispense a pointer from the server without needing to upload the same file.

A user can download the encrypted file having the pointer derived from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption removes the restrictions for the cloud to perform deduplication over the cipher texts and also the proof of ownership prevents the unauthorized user to access the file.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

II. LITERATURE SURVEY

A. Fast and Secure Laptops

A lot many people store their office and personal data on desktops and laptops. These often have weak or irregular connectivity and are exposed to vulnerable theft or any hardware failure. Thus the traditional backup systems don't go well with such type of backup recovery and thus it is inefficient to use such techniques. The algorithm described in this survey is beneficial since it uses the common data between the users to increase the storage space and speed up the backup. It supports client-end per user encryption which is needed for confidential personal data that supports a unique feature that allows the detection of common sub-trees. Backing up of data has been an important concept ever since the use of computer devices which store the confidential data. An extensive use of personal laptops is the new challenges and hence the traditional systems do not meet these challenges and hence face the real risks. Generally backups are made to some local disks but are not stored offsite they are not encrypted and thus are not vulnerable to theft. Example: Personal information in an organization is stored in plain text thus can be read by any other employee. Thus such backup leads to infrequent and irregular scheduling of data. Disadvantage: In this system if there is a change in particular node it may affect the change globally hence making it difficult to recognize the fault occurred.

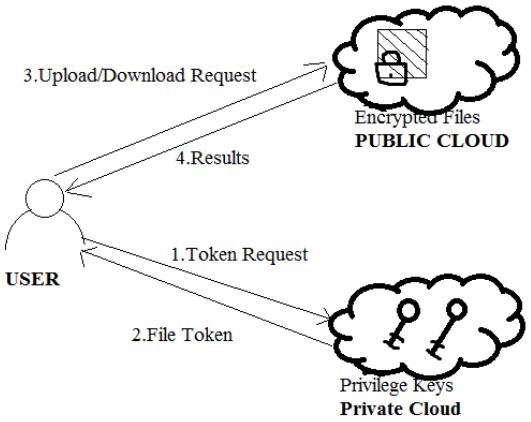
B. Proof of Ownership in Existing System

Here the ownership to upload or download a file is given. This ownership depends on a user's position like a student or a teacher. Each user will have to proof his or her own identity. PoW is implemented by an algorithm where a sender and receiver need to prove its identity. This entire process is mainly done by a prover and a verifier in which a verifier send a data copy to the server and checks for ownership of respective user. Disadvantage: A lot of storage is consumed. For example if a student uploads a file PQR and if a teacher uploads the same file on the server, since different privileges are provided to student and teacher deduplication will not be performed.

III.PROPOSED WORK

Concurrent encryption has been proposed to accomplish data confidentiality while making deduplication feasible. It encrypts/decrypts a data copy using convergent key which is obtained by computing the cryptographic hash value of the content of the data copy. After the key is generated and data is encrypted, user retains the key and sends the cipher text to the cloud. As the encryption process is deterministic and is derived from the data content identical data copies will generate same convergent keys and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is needed to provide the proof that the user indeed owns the same file when the duplicate is found. After the proof, subsequent users will be provided a pointer from the server without the need to upload the same file. A user can download the encrypted file from the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys.

A. System Architecture



B. System Modules

DATA OWNER MODULE:
Data Owner login validations.
Upload Files.
Manipulates Encrypted files.
Differential Authorization.

ENCRYPTION AND DECRYPTION MODULE:
Generate signs.
Encrypts and uploads files.
Decrypts and downloads files.
Data confidentiality.

REMOTE USER MODULE:
Accessing Files.
Remote User login validations.

CLOUD SERVER MODULE:
Authorized Duplicate Check.
Accessing files.

C. User Module

In this module, Users will have authentication and security to access the information is presented in the ontology system. Users must get themselves registered in order to get access or to search any data.

Secure Deduplication System: Here the file F and the given privilege will be used to determine the File tag F in order to support authorized deduplication. To show the difference with traditional notation of the tag, we call it file token instead. To support authorized access, a secret key kp is bounded with a privilege p to generate a file token. Let ϕ' F; p = TagGen (F, kp) denote the token of F which is only allowed to access by user with privilege p. In other words, users having the privilege key p will only be able to compute the file token ϕ' F; p. Because of this, if any file has been uploaded by a user with aduplicate token ϕ' F; p, then a duplicate check sent by another user will be successful if and only if he also has the file F and privilege p. Such as token generation function can be easily implemented as H (F, kp), where Cryptographic Hash function is determined by H.

DOI 10.17148/IJARCCE.2016.5339

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Security of Duplicate Check: As we consider several types of privacy which we need protect, that is, unforgeability of duplicate-check token: There are mainly two types of adversaries' i.e. external adversary and internal adversary. As shown below, the external adversary might be viewed as an internal adversary without any privilege. If a user has privilege p, it requires that an adversary can't forge and produce valid duplicate token with any other privilege key p' on file, where p does not equals p'. Furthermore, it also requires that the adversary does not make any request of token with its own privilege from the private cloud server, it cannot forge and produce a valid duplicate token with p on any F which have been queried upon.

Send Key: As soon as the key request is received, the sender either accept or decline it. With the received key and request id which was created at the time of sending the key request the receiver can decrypt the message.

D. Abbreviations

Acronym	Description
S-SCP	Cloud storage service provider
PoW	Proof of Ownership
pku,sku	Public and secret key pair of User
kF	Convergent encryption key for fil1e F
ϕ′F,p	Token of file F with privilege p111

E. Algorithms Used in Proposed System

1. SHA-1 and MD5

1. The file name which is the input message is broken up into chunks of 512-bit blocks and padding of message is done so that it is divisible by 512.
2. The working of padding is: 1 is initially appended to the end of the message.
3. After this zeros as are required to make the length of the message up to 64 bits which should be less than the multiple of 512.
4. The leftover bits are filled with a 64-bit integer which represents the length of the original message, in bits.
5. The MD5 algorithm uses 4 state variables who are a 32 bit integer (unsigned long). These variables are sliced and diced and are the message digest. The variables are initialized as follows:

a = 0x65894321
b = 0xRTYUHJ43
c = 0x98BADCFE
d = 0x10325476.

6. The main part of the algorithm is it uses these functions, the state variables and the message as input, which transform the state variables to message digest from their initial state. The rounds are performed for every 512 bits. Then the generated message digest would be stored in variables (a, b, c, and d). For hexadecimal form you are used to seeing, output the hex values of all the state variables, least significant byte. After the digest if, for example the generated values are:

a = 0x01234567;
b = 0x89ABCDEF;
c = 0x1337D00D
d = 0xA5510101

Then the message digest would be: 65894321RTYUHJ430DD03713010151A5 which is the required hash value of the input value.

2. AES

1. Mainly used to protect our data.
2. Output depends on the input i.e. 128/192/256 bit with value 1 and 0.
3. Consists of rounds 10, 12, 14 for 128, 192, 256 bits respectively.
4. In this paper AES will be used to generate file token number.

F. Advantages

1. Block level implementation i.e. if a same file exists in the storage then our system will divide the contents of file into blocks to check deduplication.
2. Saves storage space.
3. Secure data outsourcing.
4. Increase backup speed.

IV. CONCLUSION

In this report of project we discussed its architecture and its work flow. We will make one application based on deduplication which will thus help us in checking the duplication on the server and thus storage can be saved. We will show that our authorized duplicate check scheme has minimum overhead as compared to other traditional approaches.

ACKNOWLEDGMENT

We are heartily thankful to our Guide Prof. G.M. Gaikwad to be our constant support during this phase of research and development for being there for us whenever we needed him. We also thank our parents for believing in us and being patient with us.

REFERENCES

[1] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou. A Hybrid Cloud Approach for Secure Authorized Deduplication.
[2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.
[3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secured deduplication. In EUROCRYPT, pages 296– 312, 2013.
[4] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.