ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Secure Authorised Deduplication in Private Cloud

Shilpa Giri

PG Student, Dept. of CSE, GHRCEM, Wagholi, Pune, India

ABSTRACT: Organization requires large space for storage purpose. Data de-duplication is used to reduce the storage space requirement by the organizations. With the help of data de-duplication we save only one copy of the data and replace all other copies with a pointer which points to the original data file. The proposed system does the de-duplication which is block level de-duplication. In existing system, file level de-duplication check the token which is generated for the file and check on the storage for the same token, if the token is already present in the public cloud then instead of uploading the file the user gets the file pointer of the saved file for their reference and use. When the file level de-duplication shows the results as file is unique then it just save the pointer for that file. File level de-duplication is already done but it has a problem that it is useful only when both the files are unique. In proposed system block level de-duplication is used to solve this issue, it divides the file into the blocks and then perform the de-duplication check. De-duplication concept is mainly used in companies for the purpose of backup of data and for disaster **recovery** Applications, but now in proposed system it is used for the purpose of free up the space of primary storage. The concept of private cloud is used to avoid the duplication of data and to maintain the confidentiality of sensitive data in the cloud. To provide a better result secure ABE-based private cloud storage architecture is used which allows the users to store the sensitive data related to the organization in a private cloud.

KEYWORDS: De-duplication; authorized duplicate check; confidentiality; private cloud

I. INTRODUCTION

Cloud computing is the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.

It provides unlimited virtualized resources across the network for the users, and hides the platform and implementation details. Now a day the cloud service providers provide highly available storage space as well as the parallel computing resources at very low cost. As the cloud computing is growing rapidly, the more amount of data is stored in the cloud and many users can share it with specified privileges, which shows the access rights of the data. The data on the cloud is continuously growing and it's very difficult to maintain that data for the cloud storage providers. Recently the de-duplication of data provides a perfect solution to this problem, it offers scalability in cloud computing. The de-duplication is a technique used to reduce the duplicate data copies on storage. It is a unique data compression technique used to eliminate the redundant data on the cloud and improves the storage utilization. De-duplication saves only one copy of the file having the same content of data like other files and all the other files refer to that file for the data content. It means only one physical copy of data is available on the cloud and all the others are pointers which point towards the original file. De-duplication is done at file level means check for the whole data content of the file and eliminates the duplicate files or at block level means check for the same chunk of data content in non- identical files to remove the duplicate blocks of data. Data de-duplication has a lot of advantages but the security and privacy is a big challenge as the insider and outsider attacks can damage the sensitive data of the users. Users generally use the encryption/ decryption techniques to provide the security to their data but the old encryption techniques are not provided de-duplication. In old encryption techniques users encrypt the data with their own key that's why the same file gives the different cipher texts,so it's impossible to do the deduplication. To solve this problem a new encryption technique is used to do the cipher text which allows de-duplication for the data known as Group key encryption. Using this Group key techniqueit encrypts/ decrypts the file. Key generation and data encryption provides the data key to the user and store the cipher text to the cloud. As it uses deterministic operation which is derived from the data content same files in the same group will generate the same encryption key and therefore the cipher text is also same. To

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

control or prevent the unauthorized access, a secure proof of ownership protocol [1] is required, which provides a proof that the user also owns the file, if the duplicate is found. After this, instead of uploading the same file again, the server provides a pointer of the same file to that corresponding user. Now using this pointer the user can download the encrypted file from the server, which can be decrypted by their convergent keys.

II. RELATED WORK

A. Secure Deduplication

Recently secure data deduplication has been very popular in cloud computing from research community. For the cloud storage deduplication system Yuan et al. [2] proposed an integrity check method which decreases the storage size of the tags. Bellare et al. [3] explained by transforming the predictable message into unpredictable message we can protect the data confidentiality, which helps to enhance the security and confidentiality of the data deduplication. To generate the file tag for duplicate check, he introduced the third party called key server. Stanek et al. [4] provides a new encryption structure which provides differential security for popular means the commonly used data and unpopular data means rarely used data. For most commonly used data which is generally not much sensitive, traditional conventional encryption technique is used. And another two layered encryption scheme with more strong security while supporting deduplication is proposed for rarely used data. Li et al. [5] distributes the keys across multiple servers after encrypting the files which addressed the key management issue in block-level deduplication.

B. Convergent Encryption

Convergent encryption John R. Douceur, John R. Douceur [6] Reclaiming Space from Duplicate Files in a Server-less Distributed File System provides de-duplication with data privacy. NesrineKaaniche ,Maryline Laurent . [8] A Secure Client Side De-duplication Scheme in Cloud Storage Environments explains The increasing need for secure cloudstorage services and the attractive properties of the convergent cryptography lead us to combine them, thus, defining an innovative solution to the data outsourcing security and efficiency issues. The solution is based on a cryptographic usage of symmetric encryption used for enciphering the data file and asymmetric encryption for meta data files, due to the highest sensibility of these information towards several intrusions.

C. Proof of Ownership

Jin Li, Yan Kit Li, Xiaofeng Chen. A Hybrid Cloud Approach for Secure Authorized De-duplication explains Data de-duplication is one of main data compression techniques for removing duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth.Pietro and Sorniotti proposed another efficient PoW scheme by choosing the projection of a file onto some randomly selected bit-positions as the file proof.

D. Twin Cloud Architecture

Recently, Stefan N¨urnberger, Ahmad-Reza, Twin Clouds: Secure Cloud Computing with Low Latency shows Cloud computing assures a cost effective enabling technology to outsource storage and massively parallel computations. However, present approaches for provably secure outsourcing of data and arbitrary computations are either based on tamper-proof hardware or fully homo morphic encryption.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

III. PROPOSED SYSTEM

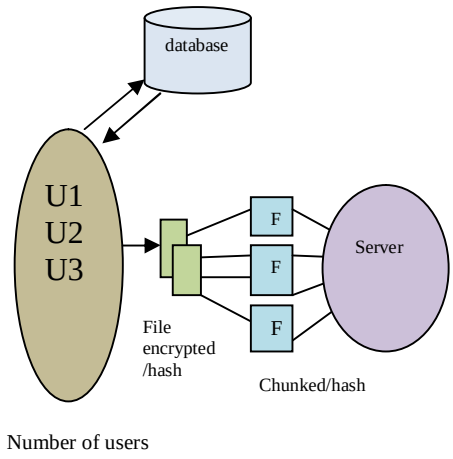

Fig 1. System architecture

In the proposed system,number of groups are created(For eg. G, G2, G3….Gn) with number of users (For eg. U1, U2, U3….Un) in each group. These users are registered in database for authentication purpose. When user login into system the authentication is checked and if the user is valid user then he is able to upload or download the data from the cloud.

In upload file process the filedivides into number of chunks and hash is generatedfor each chunk. After this the meta data (file name, hash of file,chunk files, hash of chunks)are uploaded at server.

When another user from the same group want to upload the file then the hash for that file is generated and that hash is checked with stored hash values at server. If hash is present then there is no need to upload the file but the link of that file is stored with the user, otherwise file is uploaded at the server.

IV. ALGORITHM USED

1. AES Algorithm

Step 1: Key Expansion - Using Rijndael's key schedule Round keys are derived from the cipher key.
Step 2:If Distance To Tree(u) > Distance To Tree(DCM) and First-Sending(u) then
Step 3: Initial Round - Add Round Key where Each byte of the state is combined with the round key using bitwise XOR.
Step 4: Rounds

- Sub Bytes : non-linear substitution step
- Shift Rows : transposition step
- Mix Columns : mixing operation of each column.
- Add Round Key

Step 5:Final Round: It contain Sub Bytes, Shift Rows and Add Round Key

2. SHA-1 Algorithm

Step 1: attach Padding Bits….

Message is "padded" with a 1 and as many 0's as important to bring the message length to 64 bits lesser than an even multiple of 512.

Step 2: attach Length....

64 bits are added to the end of the padded message. These bits hold the binary format of 64 bits showing the length of the original message.

Step 3: Prepare Processing Functions….

SHA1 requires 80 processing functions defined as:

f(t;B,C,D) = (B AND C) OR ((NOT B) AND D)(0 <= t <= 19)
f(t;B,C,D) = B XOR C XOR D (20 <= t <= 39)
f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)
f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Step 4: Create Processing Constants....

SHA1 needs 80 processing constant words defined as:
K(t) = 0x5A827999
(0<= t <= 19)
K(t) = 0x6ED9EBA1
(20 <= t <= 39)
K(t) = 0x8F1BBCDC
(40 <= t <= 59)
K(t) = 0xCA62C1D6
(60 <= t <= 79)

Step 5: Initialize Buffers….

SHA1 needs 160 bits or 5 buffers of words (32 bits):
H0 = 0x67452301
H1 = 0xEFCDAB89
H2 = 0x98BADCFE
H3 = 0x10325476
H4 = 0xC3D2E1F0

Step 6: Processing Message in 512-bit blocks (L blocks in total message)….

This is the an important task of SHA1 algorithm which loops through the padded and added message in 512-bit blocks. Input and predefined functions:

M[1, 2, ..., L]: Blocks of the padded and added message f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): 80 Processing Functions K(0), K(1), ..., K(79): 80 Processing Constant Words H0, H1, H2, H3, H4, H5: 5 Word buffers with starting values

V. MATHEMATICAL MODEL

In the Sub-Bytes step, every byte $a_{i,j}$ in the *state* matrix is changed with a Sub-Byte S($a_{i,j}$) using an 8-bit substitution box, the Rijndael S-box. This task gives the non-linearity in the cipher. The S-box used is generated from themultiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks which are based on easy algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also selected to avoid any confirm points (and so is a derangement), i.e., S($a_{i,j}$) ≠ $a_{i,j}$, and also any opposite fixed points, i.e., S$(a_{i,j}) \oplus a_{i,j}$ ≠0*FF. While performing the decryption, Inverse SubBytes step is used, which needs first taking the affine transformation and then finding the multiplicative inverse (just alteringthe steps used in SubBytes step).

The Shift Rows step

In the shift row step byte in every row of the state are changed cyclically to the left. The number of places each byte is shifted differs by each row.

For blocks of sizes 128 bits and 192 bits, the shifting pattern is similar. Row n is shifted left circular by n-1 bytes. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have little bit different offsets). For a 256-bit block, the first row is not changed and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only are applied for the Rijndael cipher when used with a 256-bit block, as AES fails to use 256-bit blocks. The neccessarity of this step is to fail the columns being sequentially not dependent, in which case, AES degenerates into four independent block ciphers.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

The mix columns step

In the mix column step every column of the state is multiplied with a confirm polynomial c(x).In the mix column step , the four bytes of each column of the state are mixed using a invertible linear transformation.The mix column function takes four bytes as input and outputs four bytes, where each input byte affects four output together with shift rows ,mix column provide distribution in the cipher. During this operation each column is changed using fixed matrix.

The add round key step:-

In add round key step every byte of state is added with a byte of round sub key Using XOR operation⊕. Here the sub key is adding with the state for each round sub key is generating from the main key using rijndaels key schedule, each sub key is in the similar size as the state. The sub key is added by adding each byte of the state with the corresponding byte of the sub key using bitwise XOR.

Optimization of the cipher

On systems with 32-bit or larger words, it is possible to fasten execution of this cipher by including the Sub Bytes and Shift Rows steps with the Mix Columns step by changing them into a order of table lookups. This requires four 256-entry 32-bit tables, and uses a total of four kilobytes (4096 bytes) of memory — one kilobyte for every table. A round can then be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the Add Round Key step.If the resulting four-kilobyte table size is very large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of cyclic rotates. Which Uses a byte-oriented approach, it is possible to include the Sub Bytes, Shift Rows, and Mix Columns steps into a single round operation.

VI. RESULT AND DISCUSSION

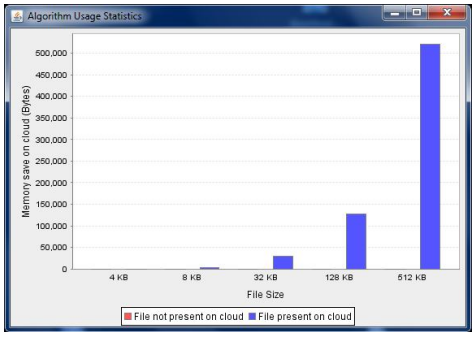

Fig 1:-Graph file size vs memory of cloud

Fig 1 Shows the graph of existing system, in that whole file gets uploaded, though the file is present with some changes. Red bar shows the files whichare present in thealreadyuploaded files in the cloud.

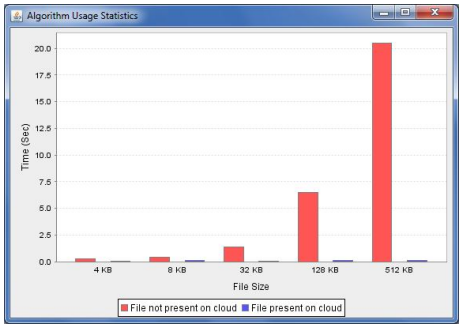

Fig 2:-Graph file size vs time of execution.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Fig 2 shows the graph of proposed system .Where red bar shows the time taken to store the file which is not present on the server, whereas blue bar shows the file present on cloud. In our proposed System the file is divided into chunks, so the chunk which is already present is not uploaded only remaining chunks which are not present are uploaded which saves lots of time of user

VII. CONCLUSION

In this paper, the proposed system will achieves goal by changing and exclusively combining techniques Advanced Encryption Standard (AES). Present System shows secure ABE-based private cloud storage architecture that allows an organization to store data securely in cloud. In ABE, association of data and attribute contains a public key component and set of attribute and message are associated with encrypting public key component. Based on the proposed ABE scheme, develop a secure cloud data storage architecture using a private cloud infrastructure.

REFERENCES

1. 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.
2. G. Timothy and M. M. Peter, "The nist definition of cloud computing," vol. NIST SP - 800-145, September 2011.
3. M. Bellare, S. Keelveedhi, and T. Ristenpart, Dupless: Server- aided encryption for deduplicated storage, In USENIX Security Symposium, Harlow, 2013.
4. J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, A secure data deduplication scheme for cloud storage, In Technical Report 2013.
5. J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, Secure deduplication with efficient and reliable convergent key management, In IEEE Transactions on Parallel and Distributed Systems 2013.
6. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, Reclaiming space from duplicate files in a serverless distributed file system, In ICDCS, Harlow, pages 617-624, 2002.
7. M. Bellare, S. Keelveedhi, and T. Ristenpart, Message-locked encryptionand secure deduplication, In EUROCRYPT pages 296-312, 2013.
8. NesrineKaaniche, Maryline Laurent A Secure Client Side Deduplication Scheme inCloud Storage Environments.Institut Mines-Telecom, Telecom SudParis, UMR CNRS 5157 SAMOVAR.
9. P. Anderson and L. Zhang, Fast and secure laptop backups with encrypted de-duplication, In Proc. of USENIX LISA, 2010.
10. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, A secure cloud backup system with assured deletion and version control, In 3rd International Workshop on Security in Cloud Computing, 2011.