An Effective Approach for Key Exposure Resistance in Cloud using De Duplication and Tile Bitmap Method

Sneha Singha
Department of Computer Engineering,
Jayawantrao Sawant College of Engineering,
Pune, India. Savitribai Phule Pune University,
Handewadi Road, Hadapsar
Pune-411028, India

S. D. Satav
Department of IT Engineering, Jayawantrao
Sawant College of Engineering, Pune, India.
Savitribai Phule Pune University,
Handewadi Road, Hadapsar
Pune-411028, India

ABSTRACT
Since a lot of data is dynamically updated and stored in today's scenario, the previous methods used for checking static data integrity can no longer be applied to analyze the integrity of the stored dynamic data in the cloud. In the existing system, the authors had focused on key management in a built in key exposure resilient system. In this paper, we have introduced the concept of de duplication strategy of data wherein the built-in key exposure resilient system will check the duplicacy of data and eliminate the redundant one using MD5 hashing. This will enforce space management in an efficient manner. Also, tile bitmap technique is used wherein the intrusions can be detected without any tampering of data to maintain its integrity.

General Terms
Cloud Computing, Cloud storage

Keywords
Cloud computing, Third Party Auditor, Key-exposure resistance, De-duplication, Reverse Circle Cipher, Tile Bitmap

1. INTRODUCTION
Cloud computing provides us a path by which we can easily access all the applications and lot of content or data globally available. Also, it helps us to create ay application or modify the same as required. Firstly we will see as to what a cloud means. Cloud is simply a network of applications.A.lternatively, we can say that cloud is something which is remotely located. Cloud delivers different services over network, i.e.,on public networks or on private networks which could be in the form of an infrastructure, platform or a software.

Cloud storage is a model where data is stored uniformly and maintained which is made available to end users over a large scale network. Storage outsourcing into the cloud is very much cost beneficial and also assists in intricacy of large-scale data storage for long term use. So even if any kind of interruption happens at the client's site, say a natural calamity or an accident, the data which has already been uploaded in the cloud will be available for access which the client will be able to download later as required. However, such kind of act violates the service level requirements of the data owner as his data is being tampered. Moreover, the huge amount of data and owner's limited computational capabilities further makes the task of storage auditing in a cloud environment expensive and even bothering for individual clients. There upon, the Third Party Auditor (TPA) was introduced which was used to audit the data. Even though many key management methods were proposed, still there was a lack of security.

Apart from key management issues, the proposed system puts forward an idea of efficiently using the storage space available in cloud space for increased storage and network efficiency. In this paper, the idea of an effective approach for key exposure resistance using de-duplication and tile bitmap method is presented, which eventually eases the process by taking input as the user data and performs the operation by using de-duplication strategy and tile bitmap method for effective cloud storage and data integrity. For further proceeding of the paper, Section 2 is dedicated for Literature survey. Section 3 defines the Problem statement. Section 4 presents the Proposed system. Section 5 presents the Results and Discussions and Section 6 narrates the Conclusion and the Future work.

2. LITERATURE SURVEY
As the previous section reveals various methodologies for enabling cloud storage auditing, but still there is a huge gap to meet the perfection. Thus, as a step towards this, this paper has gathered many previous concepts so that a new and efficient system can be proposed. The detailed studies are as follows. In [1], the authors have focussed on a new feature of cloud storage auditing. They have examined on how to lessen the damage of the client's key exposure in cloud storage auditing, and give a new solution for the same. Binary tree structure and pre-order traversal technique is adopted to update the client's secret keys.

In [2], a thorough survey of various methods of cloud storage auditing is performed. Few existent methods have been analyzed and the challenges faced have been described so that an efficient protocol can be made. When data is stored, the different version of the data is also stored uniformly. Thus, for the minimization of storage overhead, [3] "delta encoding" was adopted wherein the differences between the versions was noted. A specific type of delta encoding, skip delta encoding was adopted to improve the added cost of storing and retrieving the data.

K. Yang et al. [4] introduced a framework for auditing data storage in the cloud and also proposed an efficient privacy-preserving auditing protocol. Furthermore, it was extended to support dynamic operations like addition, deletion or modification of data. [5] explains the method of dynamic audit service to verify the integrity of a non trustable and outsourced storage which is based on the fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection.

[6] introduces a mechanism of storage integrity auditing which permits the end users to compute the cost along with achieving fast data error localization, i.e it identifies if any server misbehaves. However, for an efficient auditing, a much more secure cloud storage system was proposed which supported privacy-preserving public auditing and the results were extended so that TPA could perform audits for multiple users at the same time and also execute it efficiently. Thus, in all the above works the cloud storage auditing is tried to make more efficient in various ways.

As it is already known that the public key and the secret key play an important role in the encryption and the decryption of the data. If the secret key is exposed, it may lead to data forging and can get in hands of any unauthorized user. [7] narrates an idea of public key encryption which uses the concept of Binary Tree Encryption (BTE) wherein there is a master public key associated with the tree. Every node has a corresponding secret key and to encrypt a message destined for a particular node, one uses both public key and the name of the target node. The ciphertext as a result can then be decrypted using the secret key of the target node.

A secure cloud storage system supporting privacy-preserving public auditing was proposed [8]. The TPA was extended to perform audits for multiple users simultaneously and efficiently. [9] proposed Reverse Circle Cipher which uses 'circular substitution' and 'reversal transposition' and this helps in the handling of both confusion and diffusion. This encryption technique can be utilized for stand alone systems for personal data security or real time packet transfer for network security.

When any kind of crime is committed, forensic analysis is done. Data is being investigated as to when and what was tampered. In this paper [10], the authors have presented Tiled Bitmap algorithm which employs a logarithmic number of hash chains and checking the hash chain values produces a binary number. So, this algorithm is able to compute the pre-image of bitwise AND functions of that number. Thus, this produces a candidate set which distinguishes all the potentially corrupted granules.

3. PROBLEM STATEMENT

Cloud is a big platform to store and to retrieve the data in huge capacity. There is a greater possibility of duplication of the data and due to this, the huge storage space is used unnecessarily. So, in order to enhance the storage space available in the cloud for better network efficiency and protect and maintain the integrity of data, the proposed system put forwards an idea of maintaining the same in a key exposure resilient system.

4. PROPOSED METHODOLOGY

4.1 Overview

Cloud is a big platform to store and to retrieve the data in huge capacity. There is a greater possibility of duplication of the data and due to this the huge storage space is used unnecessarily. Also, due to the availability of many data access entities in cloud, there has always been a threat of data theft which happens by the exposure of the key of the file or data.

So, in order to overcome these issues, the proposed system puts forward an idea of avoiding these duplications on the basis of maintaining the hash tags of the files before encryption and conducting advance searches for the existing files using some powerful concepts. Also, tiled bitmap technique is adopted which is used to detect the intrusions in data by the external or internal threats on exposure of the key to the third parties.

4.2 Proposed system architecture

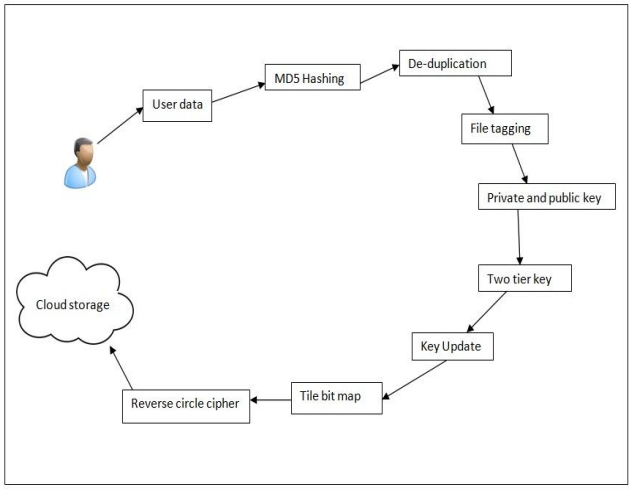

Fig. 1: System architecture

4.3 System description

Here let's describe the framework for key exposure resistance in cloud using de duplication and tile bitmap method in the following steps:

1) MD5 Hashing: In this step, as the user uploads the data into the cloud first the data contents are read in the form of string and then fed to the MD5 hashing algorithm. [10] The MD5 message-digest algorithm is a the most commonly used cryptographic hash function which produces a 128-bit (16-byte) hash value, usually expressed in text format as a 32 digit hexadecimal number. MD5 is utilized in a variety of cryptographic applications, and is also commonly used to verify data integrity.

   Once the hash key is derived from the MD5 algorithm, then every file is labeled according to the hash key which acts as the primary key in the database pattern.

2) De duplication**:** In this step, hash keys which were generated in the last steps are used to check any replication of the files. If so, the system automatically avoids that to upload to the server and then the data is re-labeled according to the user name and file name and this is known as file tagging.

3) Key Generation: In this step, current time will be taken and a hash key will be created using MD5 technique and then this hash key is subjected to fetch random key based on the algorithm mentioned in algorithm 1.

Algorithm 1: Key Generation algorithm

Input: Instance Date and time in String

Output: Key

Initialisation

1: Get the instance time and date in String Called "D_s"

2: Remove Special Symbols from D_s (Like /, -)

3: Get the MD5 Hash key of D_s in String as H

4: Assign sum=0, Key =””

5: For i=0 to length of D_s

6: Sum =sum + ASCII of D_s[i]

7: End For

8: Random integer R=sum MOD 7

9: While Key length is less than 7

10: Select Random character from H on index R

11: And concatenate to key

12: Rotate H by one character

13: End While

14: Return Key

Stop

4) Reverse Circle Cipher: Proposed system makes use of reverse circle cipher, an encryption algorithm for imposing a strong security policy. Reverse circle cipher is secured as compared to others because it makes use of private key for encryption purpose. Once the input string is obtained it is divided into blocks of 10 characters. Then these individual blocks are rotated by their respective index and then fed to the encryption module. Encryption module accepts the rotated string and based on the ASCII value of each of the character encryption is performed. Detail implementation procedure for reverse circle cipher algorithm is explained in the below algorithm 2.

Algorithm 2: Reverse Circle Cipher algorithm

Input: File Text T and Key K
Output: Encrypted Text T_E
Initialisation
1: Create a vector called DIV and initialize count=0, initialize string B to empty
2: For i=0 to length of T
3: Keep joining characters from T into String B, and count++
4: If count =10
5: Add B to DIV, set count=0 and empty B
6: End For
7: For i=0 to size of DIV
8: String B_s= DIV[i]
9: Rotate B_s by one character, initialize sum =0
10: For j=0 to length of K
11: sum =sum + ASCII of K[j]
12: END For
13: Val=sum%20
14: For j=0 to length of B_s
15: ASCII of B_s[j] + Val
16: Replace a new character
17: End For
18: Concatenate B_s to a string T_E
19: Return T_E
20: End For
Stop

5) Keys Updation: In this step, a validation time will be set. Based on that, iterations continue. On every iteration, the system captures the hash keys of the data and it is then compared with the previous one for any intrusions and this process is named as tiled bitmap signatures.

Once the intrusions are identified, then the data in the past iteration will be replaced with the current to maintain the data integrity in the cloud storage. Thus, the intruded file keys are immediately changed and the updated keys are shared across all the concerned users.

5. RESULTS AND DISCUSSIONS

To show the effectiveness of the proposed system some experiments are conducted on java based windows machine using Netbeans as IDE and Apache Tomcat as web server. And developed systems are put under hammer in many scenarios to prove its authenticity as mentioned in below tests.

5.1 Key Complexity

To measure the performance of the system, the bench mark is set by considering the system with more number of operating nodes (i.e. users).To determine the performance of the system, the check is done on how many relevant keys are been generated on the rise of the number of users in the scenario. So the available result is shown in figure 2.

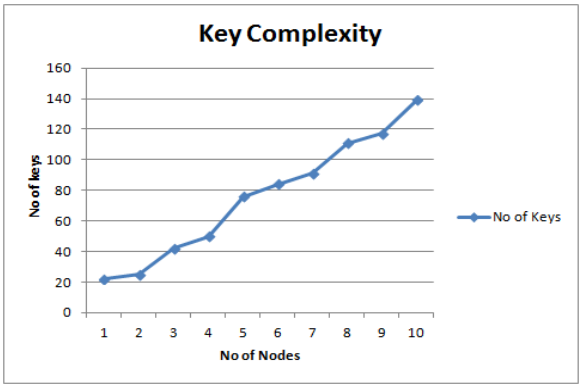

Fig. 2 : Key Complexity

The plot in figure 2 clearly indicates that the number of keys generated are always directly proportional to the number of the active users in the web system. This actually shows a good behavior of the model in cloud system.

5.2 Key Space Complexity

In any system where random keys are been generated are especially under the lenses for their space complexity. Again, key space is playing a vital role in the complete scenario as space required for the keys are always needed to be linearly dependent on the number of generated keys, which is successfully achieved by this system as shown in the figure 3. That is eventually a good sign for the key space complexity.

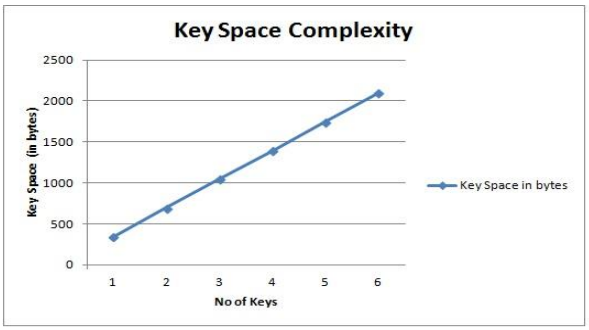

Fig. 3 : Key Space Complexity Analysis

5.3 Character Assignment for Encryption

The graph in figure 4 is drawn between the number of file character that are being used for the encryption and decryption v/s number of different characters that are using by the algorithm.

Moreover, the algorithm used in this system takes more characters to replace than the system that has been proposed by the author [9]. As the author [9] uses the characters on completion of the rotation, this makes the algorithms to take little less character than this proposed method.

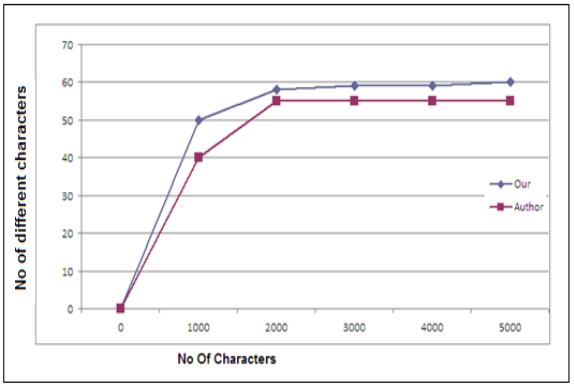

Fig. 4 : No of File character v/s No of Using different characters for the encryption and decryption

6. CONCLUSION

In this paper, the study is done on how to eliminate the duplicate files and avoid these on the basis of maintaining the hash tags of the files before encryption and conducting advance searches for the existed files using some powerful concepts. Also, tiled bitmap technique is adopted which detects the intrusions in data by the external or internal threats on exposure of the key to the third parties.

There are lot of enhancements which can be done in future. In future work, we can focus de duplication on other types of files like images, audio, video, etc where the number of pixels can be used as one factor. The same cloud storage procedure can be applied on various types of files and work can be enhanced and made more efficient. Also, the proposed method can be extended for outsourcing computation in the near future.

7. ACKNOWLEDGMENT

It is a pleasure for me to thank many people who have supported me in completion of this paper work. Firstly, I would like to thank my Guide, Prof. S.D.Satav for his support during the entire work. He highlighted the key areas in this topic that helped get the right content in the topic. Also I extend my thanks to our HOD, principal, teachers and college to provide us the facilities in the department and their guidance.

8. REFERENCES

[1] Jia Yu and Kui Ren, "Enabling Cloud Storage Auditing with Key-Exposure Resistance," in IEEE transactions on information forensics and security, VOL XX, NO1., 2015.

[2] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and opportunities," World Wide Web, vol. 15, no. 4, pp. 409-428, 2012.

[3] B. Chen and R. Curtmola, "Auditable Version Control Systems," 2014 Network and Distributed System Security Symposium, 2014.

[4] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel and Distributed Systems, Vol. 24, No. 9, pp. 1717-1726, 2013.

[5] Y. Zhu, H.G. Ahn, H. Hu, S.S. Yau, H.J. An, and C.J. Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE Trans. on Services Computing, vol. 6, no. 2, pp. 409-428, 2013.

[6] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, Vol. 62, No. 2, pp. 362-375, 2013.

[7] R. Canetti, S. Halevi and J. Katz, "A forward-secure public-key encryption scheme," Advances in Cryptology- EUROCRYPT'03, pp. 255-271, 2003.

[8] Imran Ahmad and Hitesh Gupta, "Privacy preserving public auditing and data integrity for secure cloud storage" International conference on cloud, big data and trust 2013, Nov 13-15.

[9] Ebenezer R.H.P. Isaac, Joseph H.R. Isaac and J. Visumathi, "Reverse Circle Cipher for Personal and Network Security" Information Communication and Embedded Systems(ICICES), 2013 International conference, pp 346-351, Feb 2013.

[10] Kyriacos E. Pavlou and Richard T. Snodgrass, "The Tiled Bitmap Forensic Analysis Algorithm" IEEE Trans on knowledge and data engineering, Volume 22, No 4, pp 590 – 601, May 2009.

[11] https://en.wikipedia.org/wiki/MD5.