An Unified Coding-Based Framework For Defending Against Pollution Attacks In Cloud Storage

S.Rajeswari[1], D.Subashri[2], M.Ann Bency M.E.,[3], P.Veeralakshmi M.E., Ph.D[4]
[1, 2]Dept of Information Technology
[3]Assistant Professor, Dept of Computer Science and Engineering
[4]Associate Professor, Dept of Information Technology
[1, 2, 3, 4] Prince Shri Venkateshwara Padmavathy Engineering College

Abstract- Cloud storage *is a model of data storage that the digital data is stored in logical environment, and the physical storage spans multiple servers and that is typically owned and also managed by a hosting server. These cloud storage providers are responsible for keeping the data available in the cloud. It is unified object storage for developers and enterprises, from live applications data to cloud archival. This is a simple technique and scalable way to store, access, and share data over the Internet. One of the providers such as Amazon Web Services own and maintain the network-connected hardware and software, while you provision and use what you need via a web application.*

This system enables business users to store their files in the Cloud. Files can be stored in Private or shared Company disk space. You can easily store, browse, edit, share and leave comments on documents with your co-workers or with your business partners.

Cloud computing has made several opportunities for businesses, those opportunities are numerous security challenges that need to be considered and addressed prior to committing to a cloud computing strategy.

Thus the implementation of a cloud computing strategy means that the critical data in the hands of a third party, so ensuring the data remains secure is very important. So the data needs to be encrypted at all times, that can be managed with different form of encryption techniques.

Keywords- cloud storage, pollution attack, security, performance, coding.

I. INTRODUCTION

The security of user's data to cloud is the major impact of all stakeholder. The user, System designer and cloud storage provider has to manage their data[1]. ***Pollution attacks***, whereby a set of malicious entities attempt to corrupt stored data, are one of the many risks that affect cloud data security. Those cloud storage providers should not be hacked and it should be safe to maintains their data [2]. There is a malicious entities has corrupted the stored file or data. This is the major risk factor that affect the data. Cloud storage is divided into public cloud storage, private cloud storage and hybrid cloud storage. Public cloud storage is designed specifically for large scale, multi user cloud storage.

II. LITERATURE SURVEY

In [1] represent a typical Cloud Storage system architecture, a reference Cloud Storage model and Multi-Tenancy Cloud Storage model, survey the past and the state-of-the-art of Cloud Storage, and discuss the advantage and challenges that must be addressed to implement the cloud Storage.

In [2] results indicate that it is possible to simultaneously achieve all the objectives set forth for BLCS systems by using ENIGMA, and that a careful choice of the various system parameters is crucial to achieve a good compromise among them. Moreover, they also show that LT coding-based BLCS systems outperform traditional BLCS systems in all the aspects mentioned before they implemented.

In [3] address the problem of pollution attacks in coding-based distributed storage systems. In a pollution attack, the adversary maliciously alters some of the stored encoded packets, which results in the incorrect decoding of a large part of the original data upon retrieval. We propose algorithms to detect and recover from such attacks. In contrast to existing approaches to solve this problem, our approach is not based on

adding cryptographic checksums or signatures to the encoded packets, and it does not introduce any additional redundancy to the system.

III. PROPOSED SYSTEM

In the proposed system, the implementing explains how to prevent pollution attack and how to store our file to be safe?. The following techniques and algorithm are implemented in our projects. The first technique is to the user send a file to admin with encrypted file by encryption we are using RC5 encryption algorithm. The user only able to upload the file format (.jpg, .doc). Another technique is Checksum Method, a user send an encrypted file to admin by using this method it generate a key for the particular file. the encrypted file has to verified by the admin. By this verification the checksum method generates a key to the file if the file has same key means the file not hacked or not attached. After verification the admin has to upload the file to cloud. The file should be stored safely[4]. If the user need a file means send a request to the admin after getting the response key from admin the user able to download the file successfully. By this way of file transfer the third party cannot hack and doesn't interrupt users source file. By our application large size file should be stored safely without any third party interactions.

3.1. System Architecture

As shown in Fig:1, the proposed system consists of the following algorithms:

3.1.1 Pollution detection algorithm:

A pollution detection algorithm that detects, with high probability if a set of untrusted storage resources provides at least one polluted coded fragment. The algorithm is based on a modified version of the LT decoding algorithm exploiting Gaussian Elimination; since an analytical model for decoding (and detection) performance[5] is unavailable in the literature we resort to simulations to estimate the detection probability.

3.1.2 Pollution Identification algorithm:

We design an identification algorithm that identifies the storage resources that are polluters with high probability. The algorithm we propose is not based on cryptographic checksums or digital signatures (hence it does not rely on the existence of a PKI or preestablished secure channels) and it only exploits coding redundancy and efficient decoding algorithms that require the solution of systems of linear equations.

3.1.3 BP core algorithm

Belief propagation is commonly used in artificial intelligence and information theory and has demonstrated empirical success in numerous applications including low-density parity-check codes, turbo codes, free energy approximation, and satisfiability.

The algorithm was first proposed by Judea Pearl in 1982, who formulated this algorithm on trees, and was later extended to poly trees.

It has since been shown to be a useful approximate algorithm on general graphs.

If $X=\{X_i\}$ is a set of discrete random variables with a joint mass function p, the marginal distribution of a single X_i is simply the summation of p over all other variables: However, this quickly becomes computationally prohibitive: if there are 100 binary variables, then one needs to sum over $2^{99} \approx 6.338 \times 10^{29}$ possible values. By exploiting the polytree structure, belief propagation allows the marginals to be computed much more efficiently.

Polluter identification can be cast as a statistical inference problem as follows. The main idea is to characterize each SNs i 2 AS by an unknown (hidden) binary state _i, where _i = 1 is used to identify a polluter and _i = 0 is used to identify an honest SN.
The goal is then to infer 8i 2 as p(_i = 1).

Source code:

```
Belief propagation(W,H)
1: for all a 2 W do
2: P(_a) = 0
3: if a 2 H then
4: p(_a = 1) = 0
5: else
6: p(_a = 1) = 0.5
7: end if
8: end for
9: for i = 1 to BPt do
10: G  Build random factor graph(W)
11: {p(_a)}  BP inference(G)
12: for all a 2 W do
13: P(_a) = P(_a) + p(_a = 1)
14: end for
15: end for
16: for all a 2 W do
17: P(_a) = P(_a)/BPt
18: end for
19: return {P(_a)}
```

3.1.4 RC5 Encryption:

The RC5 encryption algorithm is a fast, symmetric block cipher suitable for hardware or software implementations. [6]A novel feature of RC5 is the heavy use of data-dependent rotations. RC5 has a variable-length secret key, providing flexibility in its security level. The algorithm can be broken into two stages: initialization, and operation.

- In the initialization stage the 256-bit state table, S is populated, using the key, K as a seed. Once the state table is setup, it continues to be modified in a regular pattern[7] as data is encrypted. The initialization process can be summarized by the pseudo-code;

```
j = 0;
for i = 0 to 255:
S[i] = i;
for i = 0 to 255:
j = (j + S[i] + K[i]) mod 256;
swap S[i] and S[j];
```

It is important to notice here the swapping of the locations of the numbers 0 to 255 (each of which occurs only once) in the state table. The values of the state table are provided. Once the initialization process is completed, the operation process may be summarized as shown by the pseudo code below;

```
i = j = 0;
for (k = 0 to N-1)
{
i = (i + 1) mod 256;
j = (j + S[i]) mod 256;
swap S[i] and S[j];
pr = S[ (S[i] + S[j]) mod 256]
output M[k] XOR pr
}
```

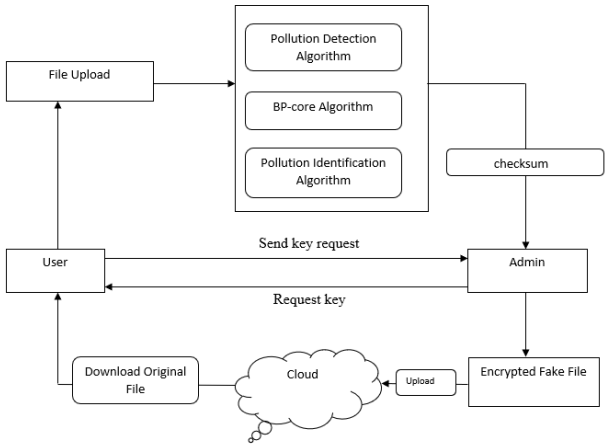

Fig:1 System Architecture

IV. IMPLEMENTATION

In this paper, we implemented how we can able securely store the files in cloud area, in which the user is registering initially with his details then login into the storage to upload files. The algorithms used in this paper are used to identify the polluter files which has been uploaded by the user. If the file has been identified as polluter file then an alert trigger will be send to the user's mail and also the file will not be uploaded into the storage area. If once the file is uploaded the admin only give permission to load the file into the cloud. The file which has been uploaded in the cloud only in an encrypted format. Admin only maintains the accessibility of the number of users. If the user wants to download the file he should send a request to admin after admin's response the original file has been downloaded. Admin can have the access of blocking the user if the user wants to again enter into the system then he has to contact admin then the unblock key will be send to the user's mail, then the user can able to access the storage.

V. CONCLUSION and FUTURE WORKS

Through this project we can have shown that rateless codes allow one to design a simple pollution detection mechanism that can be used to check data integrity during the normal read operations of a cloud-based storage system. Nonetheless, the detection mechanism alone is not enough to solve the most issues. We are also blocking the user from entering into the application. Only the admin has the entire access to the server.

The study of other rateless code families, e.g. band codes proposed, in order to investigate possible improves both in the terms of identification performance and computational cost. Hence in our project, we can implement the blocking the user who sends a polluter file.

REFERENCES

[1] Cosimo Anglano, Rossano Gaeta, and Marco Grangetto,"Exploiting Rateless Codes in Cloud Storage Systems",
IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,VOL. 26, NO. 5, MAY 2015.

[2] J.Shyamala1, A.Annie Jesus Suganthi Rani2, M.Antony Vijaya3," A Secure Storage based on Rateless Erasure Code and Order-Preserving Encryption in Cloud Computing", International Journal of Computer Science and Mobile Computing, Issue. 4, April 2015.

[3] Jian Liu,Kun Huang, Hong Rong, Huimei Wang, and Ming Xian. Privacy-Preserving Public Auditing for

Regenerating-Code-Based Cloud Storage. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 7, JULY 2015.

[4] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure regeneration with efficient and reliable convergent key management," in IEEE Transactions on Parallel and Distributed Systems, 2014, pp. vol. 25(6), pp. 1615–1625.

[5] K. Bailey, "Addressing the Growth and Complexity of Information Security Concerns," International Data Corporation (IDC), Tech. Rep., Feb. 2013.

[6] Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, and E. Magli, "Band codes for energy-efficient network coding with application to p2p mobile streaming," IEEE Transactions on Multimedia, vol. 16, no. 2, pp. 521–532, 2014.

[7] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou ,"Toward Secure and Dependable Storage Services in Cloud Computing", IEEE Transactions On Services Computing, Vol. 5, No. 2, April2012.

[8] S. T. Shen, H. Y. Lin, and W. G. Tzeng, "An effective integrity check scheme for secure erasure code-based storage systems," IEEE Transactions on Reliability, vol. 64, no. 3, pp. 840–851, 2015.

[9] Hsiao-Ying Lin, and Wen-Guey Tzeng,, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans. Parallel andDistributed Systems, vol. 23, no. 06, June. 2012.

[10] V. Bioglio, M. Grangetto, R. Gaeta, and M. Sereno, "An optimal partial decoding algorithm for rateless codes," in IEEE ISIT, 2011, pp. 2731–2735.