ISSN 2395-1621

Improving Cloud Storage Security Using Data Partitioning Technique with Recovery

[#1]Prof.A.P.Kadam, [#2]Ketaki Mane, [#3]Santoshi Garibe, [#4]Shital Gadikar

[1]ketakimane11@gmail.com
[2]santoshigaribe75@gmail.com
[3]gshital21@gmail.com
[4]mianjalikadam@gmail.com

ABSTRACT

Cloud is used for storing data on large amount on a network. It is similar to internet or network which benefits user by allowing use of resources on pay as you use basis. Hence expenditure cost is decreased on large amount. It is used for accessing, configuring and manipulating user resources online. Anyone can take benefits of higher cloud security with only internet availability. Remote data integrity is maintained with the help of cloud. Different partitioning mechanisms are used divide and store encrypted file on different servers on cloud. Servers are highly secured and error localization techniques are also used. Computational cost hence decreases. Cloud storage deals with storing data securely. In case data is crashed or hacked by hackers; hackers are able to view data only in the encrypted format. Hence data is secured. Lost data can be recovered from recovery server with the help of algorithm. Thus, it maintains security and prevents data or information from getting lost.

ARTICLE INFO

Article History

Received :24th April 2016

Received in revised form :

26th April 2016

Accepted : 28th May 2016

Published online :

30th May 2016

Keywords— Remote Data Integrity Checking, Partitioning, Error Localization, Cloud Storage,SBA,AES.

I. INTRODUCTION

Cloud computing is something with the help of which user can use computer at the remote place through the network and work on it. Different companies like amazon, allows use of their public clouds as a virtual private cloud which is useful for reducing the cost of purchasing high cost public cloud. Situation arises when company is in need of some resource for short duration of time. This resource might be not useful for that particular company in future. Such times, using cloud services user can take resources on rent with pay as you use basis further decreasing the cost of purchase. Maintenance cost is reduced as everything is maintained by third party. It is favoured with platform, location and device independency. Security is tremendously is improved. User can access the shared resources anywhere and anytime with on-demand self - service.

Clouds can be classified as public, private, community and hybrid.

- Public cloud: It can be accessed by anyone at any time and hence less security is provided. It is publically open.
- Private cloud: It is owned by private organizations and gives higher security than that of public cloud. Only authorized users can access these types of cloud.
- Community cloud: These are owned by group of different organizations. It is similar to that of private cloud in terms of security.

Hybrid cloud: It is combination of two or more types of cloud. Critical resources are handled by private cloud.

Types of Service Models in Cloud:-

Different types of service models provided by cloud are Saas,Pass,Iaas. .

Software as a Service (SaaS):- It includes different web services which performs functions which is done with software installed on individual computers.

1. Platform as a service (PaaS):- It includes different platforms as a service like operating system and file system. It protects user data privacy by authentication and secret sharing.

2. Infrastructure as a Service (IaaS):- It includes business services which are invisible to customers. It provides infrastructure as a service to user.

II. LITERATURE SURVEY

[1]. Improving cloud security using data partitioning technique
Publication Year: January February 2015
Author: Mr.Akash Kanade, Ms.Rohini Mule
This paper describes that the partition technique used for security purpose to store data on cloud. It improves data integrity checking, data storage mechanisms and encryption mechanism. Use of AES algorithm is done for encryption and decryption process.

[2]. Improving Cloud Data Storage Using Data Partition and Recovery
Publication Year: January 2015
Author: A.R.Zade, Shaikh Umar, Potghan Rahul, Rale Sagar and Borade Sagar
This paper describes partition techniques and use of digital signatures to verify the integrity of data

[3]. Improving Cloud Security Using Data Partitioning And Encryption Technique
Publication Year: 2015
Author: Mr. Akash Kanade, Ms. Rohini Mule, Ms. Namrata Nagvekar
This paper describes the partition technique used to divide file and store in on cloud server. Files are stored after encrypting them. When user needs access to particular file, it decrypts file and get access to the data.

[4]. PDDS
Publication Year: 2013
Author: C. Selvakumar,
In this paper the partitioning method is used for dividing file into different parts. It checks correctness of data by ensuring the pre-computation which is done before storing the data. Remote integrity checking and error localization is carried out by using different algorithms.

III. MOTIVATION

- To provide enormous security to the data stored on cloud servers.
- To provide recovery mechanism in situation of data loss.

IV. EXISTING SYSTEM

The existing system of cloud data security is not much secured due to some following reasons. System can be enhanced by using various algorithms and modules. Recovery of the data can be also made in in the situation of server breakdown or crash. There are some drawbacks in existing system.

• It is not much secured.
• The algorithms used has different levels of complexity depending on the type of cloud service.
• Scalability was not provided.
•Elasticity and availability of data was not provided.
•Third party auditor (TPA) is used for privacy and security issues.

V. PROPOSED SYSTEM

To overcome the drawbacks of the existing system, a new system is proposed with various advantages. This system consist of trusted third party (TTP) to help to ensure the security of the system.

A. Trusted Third Party(TTP):
It facilities the communication between two parties which trust on third party in a network.

B. Architecture of the System:

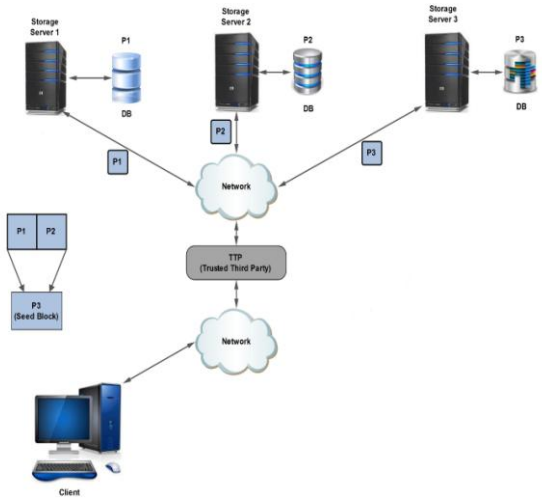

Fig.1: Architecture of the system

Process is as follows:
1. Log in to system
2. Upload the input file on TTP Server.
3. In next step calculate the size of file.
4. File partition: If size<=minimum size or size>=maximum size then show error message.
Else Split file with respect to number of servers with extension and index value.
5. Then extract the digital signature of each partition.
6. Generate secret key for each partition.
7. Encrypt respective partition using respective secret keys.
8. Store partition sequence, digital signature, keys and file attribute at TTP.
9. Send each partition at respective server.
10. Merging file: TTP request for file partitions from servers.

11. Extract new digital signature of each partition and compare it with stored digital signature at TTP.
If new digital signature equals to store digital signature at TTP Merge file otherwise data is corrupted.
12. Decrypt the merged file with key.
13. In case server is crashed or hacker tries to hack or damage the file; crashed part of the file can be recovered using recovery server with the help of seed algorithm.

C. Algorithms:

1. Partition Algorithm

Partitioning function plays an important role in this work because it splits larger files into smaller parts to
store the data effectively in quick manner enhancing easy access to data also when there is needed or demanded by end user. The original data is complex and there is difficulty in storing it in cloud, so partition function is used for make the storage easy in cloud.
Partitioning and Merging files
1. Load the Input file and size.
2. Partitioning files: Count size <= s then split file in n blocks with extension and index value.
Return files, otherwise declare as Invalid size.
3. Encrypt all partition files and store in cloud.
4. Merging files: check (SHA-1)
If (file!)
{
call recovery server and recover file and calculate the index value and merge file
}.
Else
count the index value and merge files.
Return file.
5. Decrypt the merge file for access.

1. Secure Hash Algorithm (SHA-1)

SHA1 is a message digest algorithm which takes as a input a message and produces as output 160-bit hash value. .SHA1 algorithm is 6-step process of padding of '1000…',appending message length,preparing 80 process functions,80 constant, preparing 5 word buffers, processing input in 512 blocks.
Both the transmitter and intended receiver of a message in computing and verifying a Digital signature uses the SHA1. The SHA1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. SHA-1 forms part of several widely used security applications and protocols.

Pseudo Code For SHA1 Algorithm:

Initialize Variables:

h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0

Preprocessing:

```
append the bit '1' to the message
append 0 ≤ k < 512 bits '0', so that the
resulting message length (in bits)
```

is congruent to 448 ≡ −64 (mod 512)
append length of message (before pre-processing), in bits, as 64-bit big-endian integer

Process the message in successive 512-bit chunks:

```
break message into 512-bit chunks
for each chunk
     break chunk into sixteen 32-bit big-
endian words w[i], 0 ≤ i ≤ 15
```

Extend the sixteen 32-bit words into eighty 32-bit words:

```
for i from 16 to 79
      w[i] = (w[i-3] xor w[i-8] xor w[i-
14] xor w[i-16]) leftrotate 1
Initialize hash value for this chunk:
a = h0
b = h1
c = h2
d = h3
e = h4
```

Main loop:

```
for i from 0 to 79
   if 0 ≤ i ≤ 19 then
        f = (b and c) or ((not B)/> and d)
        k = 0x5A827999
  else if 20 ≤ i ≤ 39
        f = b xor c xor d
    k = 0x6ED9EBA1
  else if 40 ≤ i ≤ 59
        f = (b and c) or (b and d) or (c
and d)
        k = 0x8F1BBCDC


   else if 60 ≤ i ≤ 79
        f = b xor c xor d
    k = 0xCA62C1D6


  temp = (a leftrotate 5) + f + e + k +w[i]
  e = d
  d = c
  c = b leftrotate 30
  b = a
  a = temp
Add this chunk's hash to result:
  h0 = h0 + a
  h1 = h1 + b
  h2 = h2 + c
```



```
h3 = h3 + d
h4 = h4 + e
```

Produce the final hash value:

digest = hash = h0 append h1 append h2 append h3 append h4

1. Advanced Encryption Standard(AES)

The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unreadable form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext .Symmetric or secret-key ciphers use the same key for encrypting and decrypting, so both the sender and receiver must know and use of same secret key. AES as well as most encryption algorithms is reversible. This means that almost the same steps are performed to complete both encryption and decryption in reverse order. The AES algorithm operates on bytes, which makes it simpler to implement and explain. AES is an iterated block cipher means that the same operations are performed many times on a fixed number of bytes. These operations can easily be broken down to the following functions:

1. ADD ROUND KEY
2. BYTE SUB
3. SHIFT ROW
4. MIX

An iteration of the above steps is called a round. The amount of rounds of the algorithm depends on the key size.
At the time of decryption the last round the **Mix Column** step is not performed.
The number of cycles of repetition are as follows:

- 10 cycles of repetition for 128 bit keys.
- 12 cycles of repetition for 192 bit keys.
- 14 cycles of repetition for 256 bit keys.

Pseudo Code For AES Algorithm:

```
Cipher(byte in[16],
 byte out[16],
key_array round_key[Nr+1])
 Begin
   byte state[16];
   state = in;
   AddRoundKey(state, round_key[0]);
 for i = 1 to Nr-1 stepsize 1 do
   SubBytes(state);
   ShiftRows(state);
   MixColumns(state);
   AddRoundKey(state, round_key[i]);
end for
SubBytes(state);
   ShiftRows(state);
   AddRoundKey(state, round_key[Nr]);
 End
```

2. Seed Block Algorithm(SBA)

SBA is useful for collecting the information from any remote location and it also help for recover the data in case of deleting the data or cloud may be destroyed. The algorithm concerns about the simplicity of backup and recovery process. SBA uses exclusive-OR (XOR) operation for computation.
For e.g. We having two data files Partition A and B then A (XOR) B produces X. When 'A' file may be destroy or delete and we want that file so can be retrieve by using XOR of file A and X.

Pseudo Code For Seed Block Algorithm:

Initialization:
Main Cloud: Mc ; Remote Server: Rs ; Clients of Main Cloud:Ci ;Files:a1 and a1 ' ; Seed Block: Si; Random Number: Ri ; Client's Id: Client_Idi..

Input: a1 created by ci ;r is generated at Mc. Output: Recovered File a1 after deletion at Mc. Given: Authenticated clients allow uploading, downloading and do modification on its own files only.

Step 1: Generate a random number. int r=rand( );
Step 2: Create a Seed Block for each Ci and Store Si at Rs. Si=r XOR Client_Idi (Repeat Step2 for all clients).
Step 3: If Ci/Admin creates/modifies a1 and stores at Mc, then a1 ' creates as a1 ' =a1 XOR Si;
Step 4: Store a' at Rs ;
Step 5: If server crashes a1 deleted from Mc , then we do XOR to retrieve the original a1 as a1=a1 ' XOR Si ;
Step 6: Return a1 to Ci .
Step 7: End.

D. Advantages of the system

- Strong Security
- Reliability
- Scalability
- Simple and Easy to use
- Confidentiality
- Integrity

E. Limitations of the system

1. Due to cost limitation user are creating private cloud using web services rather than public cloud.

2. The security between client and TTP is not ensured when public connectivity like Wi-Fi is used. Instead of it private connectivity like hotspot is used.

VI. CONCLUSION

In this work, we propose an efficient data storage security in cloud service. The partitioning of data enables storing of the data in easy and effective manner. It also gives way for flexible access and there is less cost in data storage. The space and time is also effectively reduced during storage.

Future work is planned to provide higher level of security and searching mechanisms for outsourced computations in cloud services.

REFERENCES

[1] PDDS - Improving cloud data storage security using data partitioning technique Selvakumar, C. ; Rathanam, G.J. ; Sumalatha, M.R. Advance Computing Conference (IACC), 2013 IEEE 3rd International DOI: 10.1109/IAdCC.2013.6506806 Publication Year: 2013,

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), pp. 1-9, July 2009.

[3] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73,2012.

[4] Qian Wang; Cong Wang; Kui Ren; Wenjing Lou; Jin Li; , "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," Parallel and Distributed Systems, IEEE Transactions on , vol.22, no.5, pp.847-859, May 2011.

[5] Takabi. H, Joshi.J.B.D and Ahn.G, "Security and Privacy Challenges in Cloud Computing Environments," Security & Privacy, IEEE, vol.8, no.6, pp.24-31, Nov.-Dec. 2010.

[6] Website: Ms. Kruti Sharma, Prof. Kavita R Singh "Seed Block Algorithm: A remote Smart Data Backup Technique For Cloud Computing "CSNT, 978-1-4673-5603-9 in IEEE 2013.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), pp. 1-9, July 2009

