MAINTAINING DATA INTEGRITY FOR SHARED DATA IN CLOUD

Mrunal M.Ruke, Supriya Gore, Supriya Chavan and prof : Bhawana Dakhare

Bharati Vidyapeeth College of Engineering,
Sector-7,C.B.D,Belpada,Navi Mumbai-400614, India

ABSTRACT

Cloud computing is defined as a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle the applications. User can easily modify the shared and stored data in the cloud. To overcome this data modification in cloud the signature is provided to each individual user who accesses the data in cloud.

Once the data is modified by the user on a block, the user must ensure that the signature is provided on that specific block. When user misbehaves or misuses the system the admin has authority to revoke that particular user from the group. After revoking that user, the existing user must re-sign the data signed by the revoked user. In addition to this, the security of the data is also enhanced with the help of public Auditor who is always able to audit the integrity of shared data without retrieving the entire data from the cloud.

KEYWORDS

Cloud Computing, PublicAuditor, Revoke User, Signature, Data Integrity

1. INTRODUCTION

People can easily work together in a group by Sharing and storing the services in the cloud. More specifically, once a user creates shared data in the cloud, every user in the group is able to not only access and modify shared data, but also share the latest version of the shared data with the rest of the group. To protect the integrity of data in the cloud, number of mechanisms have been proposed. In these mechanisms, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signatures. One of the most significant and common features of these mechanisms is to allow a Public Auditor to efficiently check data integrity in the cloud without downloading the entire data, referred to as public auditing. This Public Auditor could be a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or Third Party Auditor(TPA) who is able to provide verification services on data integrity to users. With shared data, once a user modifies a block, that user also needs to compute a new signature for the modified block. Due to the modifications from different users, different blocks are signed by different users.

For security reasons, when a user leaves the group or misbehaves, this user must be revoked from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. As a result, the integrity of the entire data can still be verified with the public keys of existing users only.

2. PROBLEM STATEMENT

Many public auditing mechanisms were introduced for efficient integrity checking. During public auditing method it fails to preserve the identity privacy on shared data and results in reviling significant confidential information to Public Auditor. In existing system once the user is revoked from the system, the blocks which were previously signed by this revoked user used to be re-signed with the help of straightforward method. In which the public auditor asked to existing user to first download the blocks previously signed by the revoked user, then it verifies the correctness of the blocks, then re-sign these blocks and finally upload the re-signature to the cloud. This method cause huge amount of communication and computation resources by downloading and verifying the blocks, but the content of the block remains same. This method is insecure because the private data of revoked user is misused by an existing user.

The proposed mechanism allows a public Auditor to efficiently check the data integrity in the cloud without downloading the entire data. This mechanism preserves the confidentiality of the shared data by using the proxy re-signature mechanism. In this mechanism the blocks which were previously assign to revoked user will be re-signed by the existing user. For the security purpose secret key will be provided while login.

3. LITRATURE SURVEY

Qian Wang et.al. [1] has proposed a model to solve the problem of integrity of data stored in the cloud. The TPA has allowed verifying data in cloud storage through auditing process and motivating public auditing system in the cloud. TPA check the outsourced data integrity. The advantages of auditing are to detect, prevent errors and maintain the database regularly. Auditing should not bring any new vulnerability towards privacy of data. Based on the proxy re-signature method designs a public auditing scheme for data storage with proficient user revocation in cloud. The original user can acts as a group manager and able to retract the users if necessary. For each block of data to be stored in cloud server, data owner is assigned with a signature and the integrity of data relies in the correctness of all the signatures. In a cloud if a user modifies a single block including insertion or deletion, the index of the modified block is changed and the user needs to compute a new signature for the modified block. User access the modified data with the new signature generated to perform. For security reasons, when a user leaves the group or misbehaves, user is revoked from the group.

As a result, this revoked user should no longer able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group users. Therefore the content of shared data is not changed during user revocation, the blocks which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. during user revocation, an existing user need to first download the blocks previously signed by the revoked user, verify the correctness of these blocks, then re-sign these blocks, and finally upload the new signatures to the cloud.[4]

The authors Buying Wang et.al [4] has envisioned that data can be easily shared by group. Proxy re-signature technique was utilized with help of this method and the user can re-sign the revoked user block and need not to download data from server to verify the shared data integrity and also maintain the whole data integrity.

4. SYSTEM MODEL

In this proposed system the sharing of data between users in a group with highly secure manner in the cloud. An authorized member in a group must access the shared data using HMAC algorithm. During user revocation the block which were previously signed by revoke user will be re-sign by an existing user in the group.

Also the public verifier is able to verify the integrity of shared data without retrieving the entire data from the cloud. The Integrity of the data can be verified by using SHA-1 algorithm. Identity of the signer on each block in shared data is kept private from the public verifier. This method also supports a novel public auditing mechanism for the integrity of shared data with efficient user revocation in cloud.

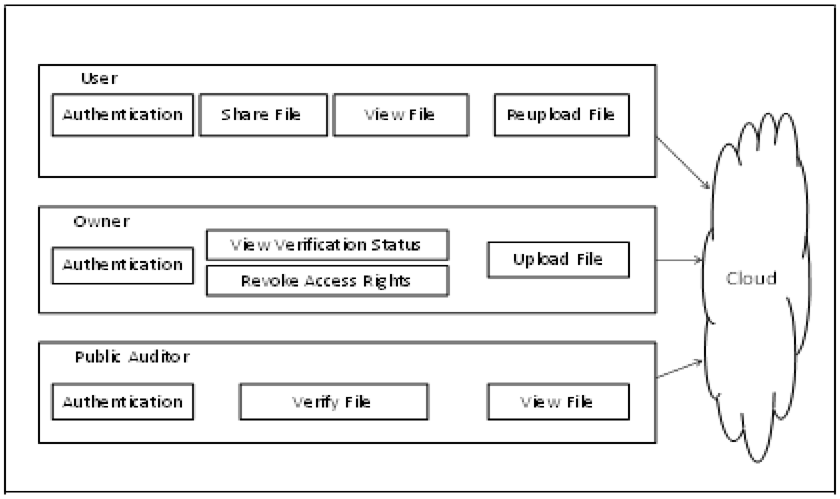

Figure1: Architecture Diagram

4.1 SYSTEM MODULE

The System consists of the following module.

4.1.1 User Module

User module can be divided into the following sub modules:

1. Registration
2. File upload
3. Download

4. Re-upload

4.1.2 Auditor Module

Auditor Module can be divided into the following sub modules:

1. File Verification
2. View Verification Status

4.1.3 Owner Module

Owner Module can be divided into the following sub modules:

1. View Files
2. Revoke User

5. SECURITY MODEL

5.1 For signature generation

For generating signature on the blocks we have used Hmac algorithm.Hash-based message authentication code (HMAC) provides the server and the client each with a public and private key. The public key is known, but the private key is known only to that specific server and that specific client. The client creates a unique HMAC, or hash, per request to the server by combing the request data and hashing that data, along with a private key and sending it as part of a request. The server receives the request and regenerates its own unique HMAC. The server compares the two HMACs, and, if they're equal, the client is trusted and the request is executed.

HMAC can then be expressed:

1. Append zeros to the left end of K to create a b-bit string K+ (for example, if K is of length 160 bits and b = 512, then K will be appended with 44 zero bytes 0x00).

2. XOR (bitwise exclusive OR) K+ with ipad to produce the b-bit block Si.

3. Append M to Si.

4. Apply H to the stream generated in Step 3.

5. XOR K+ with opad to produce the b-bit block So.

6. Append the hash result from Step 4 to So.

7. Apply H to the stream generated in Step 6 and output the result.

5.2 Integrity Verification

For verifying the data integrity SHA-1 algorithm is used.SHA-1 stands for Secure Hash algorithm.

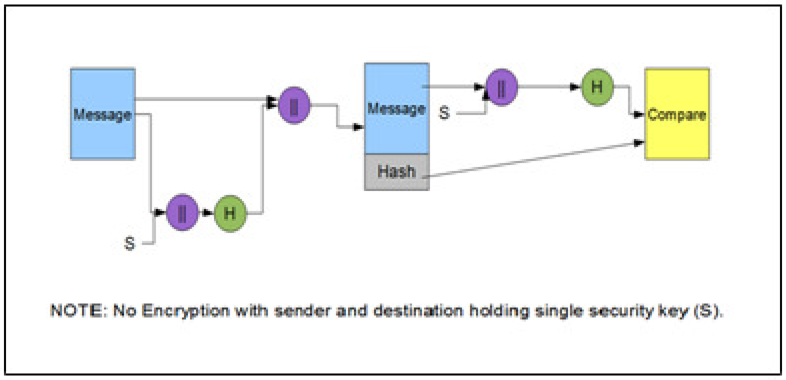
Message
S
||
H
Message
Hash
Compare
NOTE: No Encryption with sender and destination holding single security key (S).

Figure 2: Basic hash function

SHA-1 (160 bit message) Algorithm Framework

- Step 1: Append Padding Bits

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

- Step 2: Append Length

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating 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)

- Step 4: Prepare Processing Constants

SHA1 requires 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 requires 160 bits or 5 buffers of words (32 bits):

H0 = 0x67452301

H1 = 0xEFCDAB89

H2 = 0x98BADCFE

H3 = 0x10325476

H4 = 0xC3D2E1F0

- Step 6: Pseudo Code

```
For loop on k = 1 to L
        (W(0),W(1),...,W(15)) = M[k] /* Divide M[k] into 16 words */
        For t = 16 to 79 do:
        W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1
        A = H0, B = H1, C = H2, D = H3, E = H4
        For t = 0 to 79 do:
            TEMP = A<<<5 + f(t;B,C,D) + E + W(t) + K(t) E = D, D = C,
                        C = B<<<30, B = A, A = TEMP
        End of for loop
    H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E
End of for loop
```

Output:

H0, H1, H2, H3, H4, H5: Word buffers with final message digest

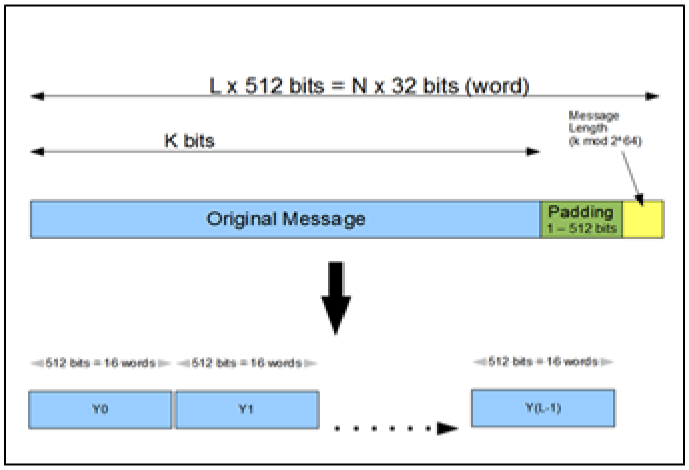

Figure 3: Message Diagram

6. CONCLUSION

“Maintaining Data Integrity for Shared Data in cloud” is required where, the cloud offers data storage and sharing services to the group. By using the service user can easily modify and share the confidential data in the system. To restrict the data modification the signature is assigned on each block of the data. Due to which user will able to modify or access the data by the permission of the owner of the data. If user tries to misuses the data in system, the data owner has authority to revoke that user from the group and by using proxy re-signature mechanism the signature on the data blocks of revoked user is overridden by the new signature. This helps in preserving the security of the system. To extent the security the secret key is also provided to the user at the time of login.

In addition to this, the security of the data is also enhanced with the help of public verifier, who is always able to audit the integrity of shared data without retrieving the entire data from the cloud.

REFERENCES

[1] Q.Wang, C.Wang,J. Li,K. Ren, W. Lou Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing,”4epp. 355-370

[2] Boyang Wang, Baochun Li, Hui Li, ” Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud” IEEE Trans.

[3] International Research Journal of Engineering and Technology (IRJET) ,”Public Auditing For Shared Data with efficient manner in cloud”, In the preceding of irjet,- ISSN: e-ISSN: 2395 -0056 Volume: 02 Issue: 09 | Dec-2015

[4] B. Wang, B. Li, and H. Li, “Public Auditing for Shared Data with Efficient User Revocation in the Cloud,” in the Proceedings of IEEE INFOCOM 2013, 2013, pp. 2904–2912

[5] International Journal of Scientific Engineering and Applied Science (IJSEAS), “Effective Privacy-Preserving Public Auditing for Data Sharing in Cloud” , In the preceding of ijseas, - ISSN: 2395-3470 Volume-1, Issue-4| July-2015

AUTHORS

Mrunal Ruke is a Final year engineering student from Bharati vidyapeeth college of Engineering Kharghar,Navi mumbai,India

Supriya Gore is a Final year engineering student from Bharati vidyapeeth college of Engineering Kharghar, Navi mumbai,India

Supriya Chavan is a Final year engineering student from Bharati vidyapeeth college of Engineering Kharghar, Navi mumbai,India