Uploading data to public cloud storage using proxy and monitoring its integrity using MD5 Algorithm

Tanmay Borkar[1], Mangesh Dudhal[2], Snehal Ghadage[3], Sumit Kawale[4]

[1] *Student, Department of Computer Engineering, NBNSSOE, Maharashtra, India*
[2] *Student, Department of Computer Engineering, NBNSSOE, Maharashtra, India*
[3] *Student, Department of Computer Engineering, NBNSSOE, Maharashtra, India*
[4] *Student, Department of Computer Engineering, NBNSSOE, Maharashtra, India*

ABSTRACT

In today's technical world, use of internet has widely spread in all the fields of science. With that multiple techniques are invented to extract usefulness of internet. Cloud computing is one of these fields. It is the most popular concept nowadays which is widely used. The public cloud servers are used for data storage which makes rapid development in cloud. This is the cause for storing the data onto the cloud infrastructure. But for security purpose user needs to encrypt their important data while storing it in the cloud. Client data checking should be done no matter whether it is outsourced or done without downloading the data. As there are several threats and issues with data, we have proposed "Proxy-Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud". Cryptographic Proxy Oriented Data uploading is used for avoiding data from fake source and algorithm for protecting user data from external as well as internal environment. Cloud services are very popular but because of security problem unauthorized user will be restricted from storing or uploading data on cloud. To solve this issue, the unauthorized user will then modify the proxy and get data uploaded to cloud. The existing systems do not provide the security mechanism for storing data and hence the proposed system will fulfill all issues caused by existing system. Data uploading with proxy alteration gives the advantage of data stealing and uploading malwares on cloud environment as well. Integrity of remote data is another issue in public cloud storage (PCS). The proposed system provides the provision of checking data integrity without actually downloading it and manually checking it. Proposed system is also efficient and flexible in all aspects.

Keyword: - *: Cloud Computing, Data Integrity, Key Generation, Encryption and Decryption*

1. INTRODUCTION

Nowadays as daily routine, technical sound gadgets are mostly preferred than the manual gadgets for different operations. Large number of use of internet enables the user to use data on large scale. Hence to store this big type of the data, cloud computing is popular way. Many of the users are given real preference to store data on public cloud servers. The main issue is security issue in the cloud storage. Newly emerged security problems have to be solved to help clients to compute their data in public cloud. Security issue solved by Proxy-Oriented Data uploading and Remote Data Integrity Checking in Public Cloud can ensure the data will be protected without downloading. We proposed system for remote data integrity checking in public cloud and proxy –oriented data uploading to cloud. For designing this system, ID-PUIC protocol which is efficient and flexible both, is used.

Cloud storage is good for huge data storage and maintaining the data, which is helpful for Business data or any other important data. In public cloud environment, most of the users can upload their data to public cloud server and check their remote data's integrity through Internet. Many of users transfer their data to public cloud server and through internet which does not check the data integrity. The clients store their huge data over remote public cloud servers. As the data which is stored is out of control of the users, it is more prone to security risks. In this stage the risk involved are confidentiality compromised, accessible services and data integrity lose.

Cryptographic Proxy Oriented Data uploading avoids data from fake sources and algorithms for protecting user data from external as well as internal environment. Integrity checking is one of the basic requirements on many systems. Integrity checking is used for integrity preservation of cloud user data. Access to the public cloud server is limited in some cases. The user uploads data to the third party that acts as proxy and lets further execution easily. Proposed protocol can realize public remote data integrity checking and private remote data integrity checking, based on the original client's authorization.

Based on identity-based public cryptography and proxy validations, Remote data integrity checking protocol which suits best for devices with limited access capacity is used. While uploading file to the cloud storage, proxy separately saves copy of same file because when file is corrupted or hacked by hacker or integrity of file not ensured then same files are regenerated by proxy. We introduced secure system or security model for the same reason. ID-PUIC protocol is efficient and protected. It can also provide public and private remote data integrity checking and delegated remote data integrity checking based on the authorization of actual users. Proxy is used as security proof technique for the cloud storage.

2. PROPOSED SYSTEM

This paper primarily focuses on uploading data using proxy and checking data integrity remotely. By having identity-based public key scientific discipline, our planned system is economical with respect to cost. This is because the certificate management in system is eliminated. ID-PUIC may be a novel proxy-oriented knowledge uploading and remote knowledge integrity checking model publicly cloud. We tend to offer the formal system model and security model by designing the primary concrete ID-PUIC protocol. Therefore, our designed system is incontrovertibly secure.

2.1 CONCRETE ID-PUIC PROTOCOL

Concrete ID-PUIC protocol contains four Servers:

- Middle server
- Encryption - Decryption server
- Checksum server
- Proxy Server

So as to point out the intuition of our construction, the concrete protocol's design is represented in figure 2. First, setup is performed on every servers involved in the system and then client is allowed to upload data.

Before uploading data through PC, user will receive an OTP generated by Key Generation Algorithm for user authentication. The data will be then sent to the Middleware Server. It will then send data to Encryption Server for encryption, which will then send back encrypted data to Middleware Server. Now the checksum of encrypted file will be calculated through Checksum server and the encrypted file will be send to Proxy server. Proxy server will upload data to DRIVEHQ cloud. Now when user wishes to check data, user will request for data integrity check (without actually downloading the file).Also user can download (the uploaded) file from the cloud in a secure way.

2.2 PRIVATE CHECKING and DELEGATED CHECKING

Our planned system satisfies the non-public checking (Checksum check), delegated checking (Proxy check) for data. Proxy oriented data uploading represents that every user should upload data to cloud through particular proxy only. Based on the system architecture, our system also supports checking the integrity and intactness of data uploaded on cloud without downloading the actual data. Furthermore, by using identity-based public key cryptology, our proposed system is very efficient since the certificate management is eliminated.

2.3 AES ALGORITHM

The AES stands for Advance Encryption standard. AES is substitution-permutation based design which is a combination of both substitution and permutation. AES has a 128 bit fixed size block with a 128, 192, 256 bits size key. To obtain the different size of keys the different number of transformation need to be done repeatedly. The AES

cipher converts the input i.e. plaintext into the cipher text. The cycles of rounds are 10 cycles of repetition to obtain the 128-bit key, 12 cycles for 192-bit key and 14 cycles for the 256-bit keys. The different rounds and steps in the AES algorithm are given below –

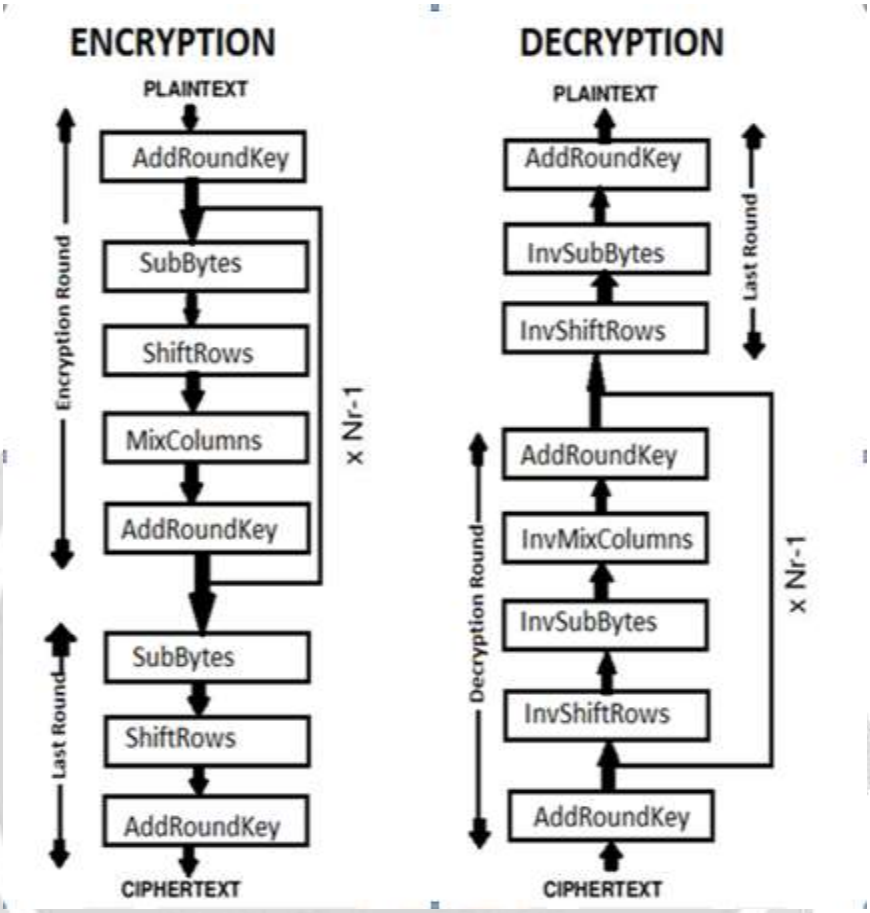

Fig -1: AES Algorithm

2.4 MD-5 ALGORITHM

The MD5 is a hashing function which produces 128 bit hash value. It is used to calculate the checksum which can be used to check the integrity of the files. The MD5 digest is largely used while transferring data over a network to check the intactness of the data. For this, the checksum of the data is calculated first and then transferred along with the data. So that when the data is arrived the checksum of the data is calculated and compared with the checksum provided with the transferred data. If the checksum calculated is same as that of the provided one, then the data is intact i.e. the data is not tampered or corrupted.

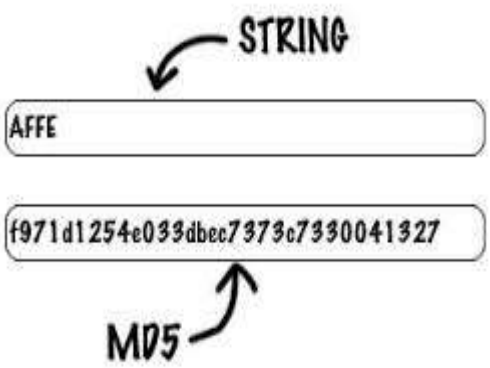

Fig -2: Input and output for MD 5 algorithm

2.5 CHECKSUM SERVER AND INTEGRITY FUNCTION

It is a server that needs to be activated to forward file to cloud. This is the server which acts like intermediate through which file need to be uploaded on cloud. Its role is to calculate the checksum of file before uploading file to cloud. Integrity Check function which resides in Checksum Server is activated for checking and processing the intactness of file. It is one of three servers which need to be initialized before sending request for Integrity Verification. Its role is to validate the string received from checksum Integrity server and Proxy Integrity Server and pass the result to user.

2.6 PROXY SERVER

In computer networking, the server which acts as an intermediary for requests to the other servers from the clients is known as proxy server. The server is set as a gateway for the file to be uploaded on a specific cloud service. At first the proxy server is activated and then the file to be uploaded on the cloud is transfer through the proxy server. This server is set as the only single entry point for the transfer of file to and from the cloud. This ensures that no unauthorized personal or machine access the resources from the cloud server. It is a server that needs to be activated to forward file to cloud. This is the server which acts like single proxy only through which file need to be uploaded on cloud and downloaded from cloud.

2.7 ENCRYPTION-DECRYPTION SERVER

The Encryption and Decryption servers are special dedicated servers for the security of file. The Encryption Server plays an import role in this system. After activating the encryption server, file which is send to this server gets encrypted using AES algorithm. After the encryption using AES algorithm, the file is send to next server to initiate the uploading process.

The Decryption server is needs to be activated before downloading file from cloud. After activating the decryption server, the file received from Proxy Server which is in encrypted format is decrypted and this decrypted file is sent as a response to user.

2.8 MIDDLEWARE SERVER

There are several transactions which occur in the network at a single time. Some transactions include sending file from one server to other server, sending encrypted data, checksum values, and other responses. To handle these transactions the Middleware Server acts as a middle agent. The Middleware server is needed to be activated to initiate the Uploading or Downloading Process. Basically this server is connected to all the components of the network and handles the transaction in between them.

3. SYSTEM ARCHITECTURE

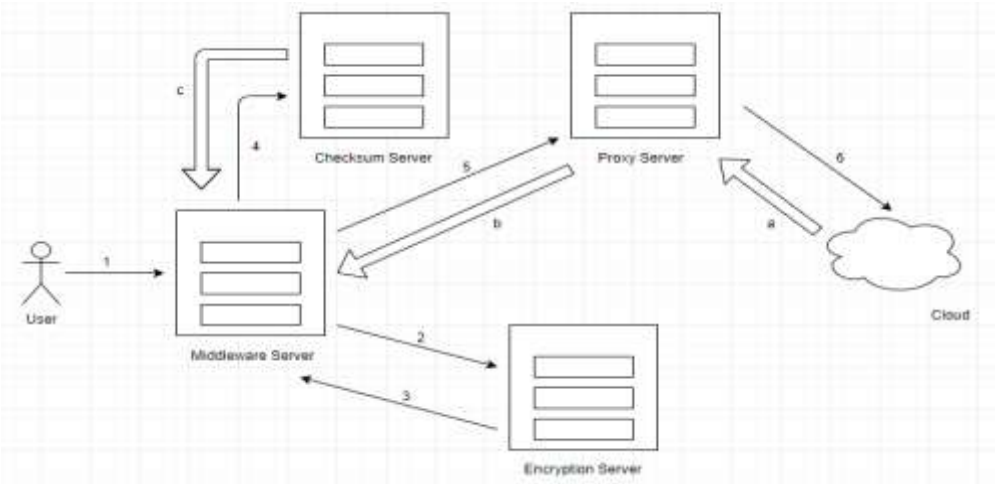

Fig-3: System Architecture

4. CONCLUSION AND FUTURE WORK

Motivated by the application benefits, this paper proposes the novel security concept of ID-PUIC in public cloud. The paper formalizes ID-PUIC's system model and security model. The concrete ID-PUIC protocol is provably secure and efficient by using the formal security proof and efficiency analysis. On the other hand, the proposed system can also recognize private remote data integrity checking which is based on the authorization of original client's. This provide infrastructure for any user to check the intactness of data on cloud without actually downloading it by integrity check. Thus our proposed system is beneficial to upload the files containing data in secure way.

5. REFERENCES

[1]. Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu,"Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," IEICE Trans. Communication, vol. E98-B, no. 1, pp. 190–200, 2015.

[2]. Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," J. Internet Technol., vol. 16, no. 2, pp. 317–323, 2015.

[3]. M.Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in Proc.CCS, 1996, pp. 48–57.

[4]. E.-J. Yoon,Y. Choi, and C. Kim, "New ID-based proxy signature scheme with message recovery," in Grid and Pervasive Computing (Lecture Notes in Computer Science), vol. 7861. Berlin, Germany: Springer- Verlag, 2013, pp. 945–951.

[5]. B.-C. Chen and H.-T.Yeh, "Secure proxy signature schemes from the weil pairing," J. Super computer, vol.65, no. 2, pp. 496–506, 2013.

[6]. X. Liu, J. Ma, J. Xiong, T. Zhang, and Q. Li, "Personal health records integrity verification using attribute based proxy signature in cloud computing," in Internet and Distributed Computing Systems (Lecture Notes in Computer Science), vol. 8223. Berlin, Germany: Springer-Verlag, 2013, pp. 238–251.

[7] H. Guo, Z. Zhang, and J. Zhang, "Proxy re-encryption with unforgeable re-encryption keys," in Cryptology and Network Security (Lecture Notes in Computer Science) vol. 8813. Berlin, Germany: Springer-Verlag, 2014, pp. 20–33.

[8] E. Kirshanova, "Proxy re-encryption from lattices" in Public-Key Cryptography (Lecture Notes in Computer Science), vol. 8383. Berlin, Germany: Springer-Verlag, 2014, pp. 77–94.

[9] P. Xu, H. Chen, D. Zou, and H. Jin, "Fine-grained and heterogeneous proxy re-encryption for secure cloud storage," Chin. Sci. Bull, vol. 59, no. 32, pp. 4201–4209, 2014.

[10] S. Ohata, Y. Kawai, T. Matsuda, G. Hanaoka, and K. Matsuura, "Re-encryption verifiability: How to detect malicious activities of a proxy in proxy re-encryption" in Proc. CT-RSA Conf., vol. 9048. 2015, pp. 410–428.