International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 12 | Dec-2017 www.irjet.net p-ISSN: 2395-0072

Block-Level Message-Locked Encryption for Secure Large File De-duplication

Bhagyashree Bhoyane[1], Snehal Kalbhor[2], Sneha Chamle[3], Sandhya Itkapalle[4] ,P. M. Gore[5]

[1234] Student, Computer Department, Padmabhushan Vasantdada Patil Institute of Technology, Pune, Maharashtra
[5] Professor, Computer Department, Padmabhushan Vasantdada Patil Institute of Technology, Pune, Maharashtra

---

Abstract: *In order to reduce the burden of maintaining big data, more and more enterprises and organizations have chosen to outsource data storage to cloud storage providers. This makes data management a critical challenge for the cloud storage providers. Cloud computing is the long dreamed vision of computing as a utility. Besides all the benefits of the cloud computing security of the stored data need to be considered while storing sensitive data on cloud. Cloud users cannot rely only on cloud service provider for security of their sensitive data stored on cloud.To achieve optimal usage of storage resources, many cloud storage providers perform de-duplication, which exploits data redundancy and avoids storing duplicated data from multiple users.MLE scheme can be extended to obtain secure de-duplication for large files, it requires a lot of metadata maintained by the end user and the cloud server. The technique of Third Party Auditor (TPA) checks integrity of data stored on cloud for data owner.*

KEYWORDS: **NLP (Natural language processing), Sentiment Analysis, synsets, Word Net**

1. INTRODUCTION

Reducing the burden of maintaining big data, more and more enterprises and organizations have chosen to outsource data storage to cloud storage providers. This makes data management a critical challenge for the cloud storage providers. To achieve optimal usage of storage resources, many cloud storage providers perform deduplication, which exploits data redundancy and avoids storing duplicated data from multiple users.

In terms of deduplication granularity, there are two main deduplication strategies. File-level deduplication: the data redundancy is exploited on the file level and thus only a single copy of each file is stored on the server. Block-level deduplication: each file is divided into blocks, and the sever exploits data redundancy at the block level and hence performs a more fine-grained deduplication.

In the traditional encryption providing data confidentiality, is contradictory deduplication occurs file level and block level. The duplicate copy of corresponding file eliminate by file level deduplication .For the block level duplication which eliminates duplicates blocks of data that occur in non-identical files.

2. RELATED WORK

[1] Deduplication is a popular technique widely used to save storage spaces in the cloud. To achieve secure deduplication of encrypted files, Bellare et al. formal a new cryptographic primitive named Message-Locked Encryption (MLE) in Eurocrypt 2013. Although an MLE scheme can be extended to obtain secure deduplication for large files, it requires a lot of metadata maintained by the end user and the cloud server. In this paper, we propose a new approach to achieve more efficient deduplication for (encrypted) large files. Our approach, named Block-Level Message-Locked Encryption (BL-MLE), can achieve file-level and block-level deduplication, block key management, and proof of ownership simultaneously using a small set of metadata. We also show that our BL-MLE scheme can be easily extended to support proof of storage, which makes it multi-purpose for secure cloud storage.

[2] With the continuous and exponential increase of the number of users and the size of their data, data deduplication becomes more and more a necessity for cloud storage providers. By storing a only one of its kind copy of duplicate data, cloud providers greatly shrink their storage and data transfer costs. The advantages of deduplication unfortunately come with a high cost in terms of new security and privacy challenges. We advise ClouDedup, a safe and well-organized storage space check which assures block-level deduplication and data confidentiality at the same time. Although based on convergent encryption, ClouDedup remains secure thanks to the definition of a component that implements an additional encryption operation and an access control mechanism.

[3] In this paper, we describe a COM between the interior enterprise application and public cloud storage platform which is closer to the client we called-- -Cloudkey, which is designed to take responsible for enterprise data backup business. Cloudkey stores data persistently in a cloud storage provider such as Amazon S3 or Windows Azure , allowing users to take advantages of the reliability and large storage capacity of cloud providers, also avoiding the need for dedicated server hardware. Clients access to the storage through Cloudkey running on-site, which provide lower-latency responses and additional opportunities for optimization through caches data.

[4] Data deduplication is one of the most important data compression techniques, used for removing identical copies of

repetitive data. For reduce duplication of data authorized duplication system is used. When a user uploads a file on the cloud, the file is split into a number of blocks, each block having a size of 4KB. Block is encrypted using a convergent key and subsequently a token is generated for it by using token generation algorithm. After encrypting the data using convergent key, users retain the key before sending the ciphertext to the cloud. Due to the deterministic nature of encryption, if identical data copies are uploaded the same convergent keys and the same cipher text will be produced thus preventing the deduplication of data. Each block is then compared with the database of cloud. After comparing, if a match is found in the cloud database then only metadata of the block is stored in DB profiler. This paper also prevents unauthorized access by using a secure proof of ownership protocol .The protocol uses authorize deduplicate check for hybrid cloud architecture.

3. PROPOSED SYSTEM

Data Owner uploads document, metadata, checksum on cloud after encryption using keys from Data Owner and Cloud Service Provider. Also, a copy of metadata and checksum is sent to Auditor.

Registered users send access request and receive encrypted file if authorized. User calculates checksum to compare with original and reports to Data Owner if checksum mismatch occurs. Avoid De-duplication

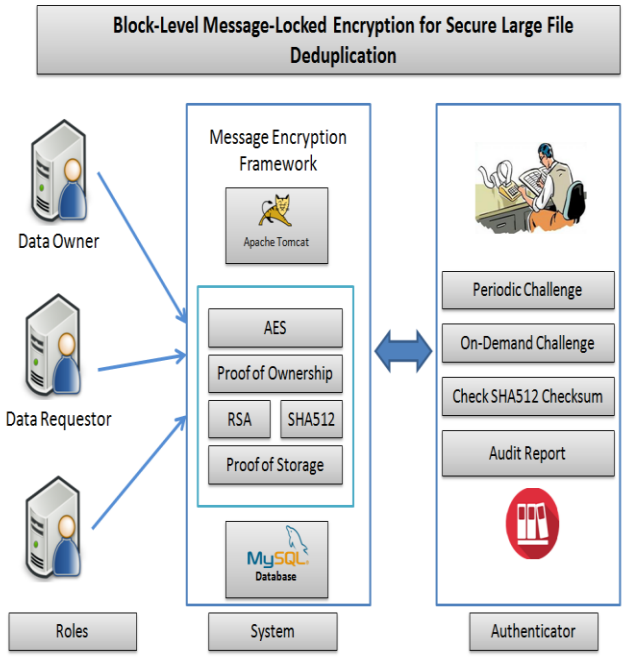

Fig-1: Archtecture of Block-Level Message-Locked Encryption for Secure Large File De-duplication

a. File Level
b. Block Level

Maintains the checksum of file data and block of file data and compare at the time of file upload to avoid De-duplication.

Auditor Receives metadata after upload. Performs periodic or on-Demand integrity checks by sending challenges to Cloud Service Provider. On response from Cloud Service Provider, Auditor confirms response and reports status to Data Owner.

4. PROPOSED METHOD

Algorithm Used:

1. AES Algorithm

a. Encryption

You acquire the subsequent AES steps of encryption for a 128-bit block:

1. Derive the set of round keys from the cipher key.
2. Initialize the state array with the block data (plaintext).
3. Add the initial round key to the starting state array.
4. Perform nine rounds of state manipulation.
5. Perform the tenth and final round of state manipulation.
6. Copy the final state array out as the encrypted data (cipher text).

Each round of the encryption process requires a series of steps to alter the state array.

These steps involve four types of operations called:

1. Sub-Bytes
2. Shift-Rows
3. Mix-Columns
4. Xor-Round Key

b. Decryption

As you might expect, decryption involves reversing all the steps taken in encryption using inverse functions:

1. InvSub-Bytes
2. InvShift-Rows
3. InvMix-Columns

Operation in decryption is:

1. Perform initial decryption round:

- Xor-Round Key
- InvShift-Rows
- InvSub-Bytes

2. Perform nine full decryption rounds:

Xor-Round Key
InvMix-Columns
InvShift-Rows
InvSub-Bytes

3. Perform final Xor-Round Key

2. RSA Algorithm

The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.

For protection purposes, the integers p and q should be favored at random, and should be comparable in magnitude but 'differ in length by a few digits to make factoring harder. Prime integers can be efficiently found using a primality test.

2. Compute n = pq.
   - n is used as the modulus for both the public and private keys. Its length, frequently articulated in bits, is the key length.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function. This value is kept private.
4. prefer an integer e such that $1 < e < \varphi(n)$ and gcd(e, $\varphi(n)$) = 1; i.e., e and $\varphi(n)$ are coprime.
5. Determine d as $d \equiv e-1 \pmod{\varphi(n)}$; i.e., d is the modular multiplicative inverse of e (modulo $\varphi(n)$)
   - This is more clearly stated as: solve for d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$
   - e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly 216 + 1 = 65,537. on the other hand, a large amount of smaller values of e (such as 3) have been shown to be fewer protected in some settings.
   - e is released as the public key exponent.
   - d is kept as the private key exponent.

The public key consists of the modulus n and the public (or encryption) exponent e. The confidential input consists of the modulus n and the private (or decryption) example d, which must be kept secret. p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d.

3. SHA 512 Algorithm

a. Append Padding Bits and Length Value:

This step makes the input message an exact multiple of 1024 bits:

b. Initialize Hash Buffer with Initialization Vector:

Before we can process the first message block, we need to initialize the hash buffer with IV, the Initialization Vector
c. Process Each 1024-bit (128 words) Message Block Mi:

Each message chunk is taken through 80 rounds of handing out.

d. Finally:

After all the N message blocks have been processed, the content of the hash buffer is the message digest.

5. CONCLUSION

In System Block-Level Message-Locked Encryption for Secure Large File De-duplication, is one of the most important data compression techniques, used for removing identical copies of repetitive data. For reduce duplication of data authorized duplication system is used.

REFERENCES

[1] Rongmao Chen*, Yi Mu*, Senior Member, IEEE, Guomin Yang, Member, IEEE, and Fuchun Guo "BL-MLE: Block-Level Message-Locked Encryption for Secure Large File Deduplication" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY - 2016

[2] Pasquale Puzio, Refik Molva, Melek O¨ nen, Sergio Loureiro "ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage" IEEE -2012

[3] Mr.Vinod B Jadhav ,Prof.Vinod S Wadne "Secured Authorized De-duplication Based Hybrid Cloud Approach" International Journal of Advanced Research in Computer Science and Software Engineering – 2014

[4] Aparna Ajit Patil, Asst. Prof. Dhanashree Kulkarni "Block Level Data Duplication on Hybrid Cloud Storage System" International Journal of Advanced Research in Computer Science and Software Engineering - 2015

[5] Chunlu Wang, Jun Ni, Tao Xu, Dapeng Ju "TH_Cloudkey: Fast, Secure and lowcost backup system for using public cloud storage" International Conference on Cloud and Service Computing - 2013