IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMPROVING ACCESSING EFFICIENCY OF CLOUD STORAGE USING DEDUPLICATION SCHEMES

Sonia.P.[1] Sowmiya.J[2] Swetha.G[3] Valluru Deekshitha[4],P.Neelaveni[5]

Student[1,2,3,4], Professor[5]

Department of Computer Science and Engineering

PERI INSTITUTE OF TECHNOLOGY

Abstract

Data de duplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality of sensitive data while supporting de duplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To better protect data security, this proposed system makes the first attempt to formally address the problem of authorized data de duplication. Different from traditional de duplication systems, the differential privileges of users are further considered induplicate check besides the data itself. We also present several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture. Security analysis demonstrates that our scheme is secure in terms of the definitions specified in the proposed security model. As a proof of concept, the proposed work implements a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments using our prototype. The proposed work shows that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Introduction

Cloud computing provides seemingly unlimited “virtualized” resources to users as services across the whole Internet, while hiding platform and implementation details. Today’s cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified *privileges*, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data.

To make data management scalable in cloud computing, de duplication has been a well-known technique and has attracted more and more attention recently. Data de duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. De duplication can take place at either the file level or the block level. For filelevel de duplication, it eliminates duplicate copies of the same file. De duplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication.

Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different cipher texts, making de duplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making de duplication feasible. It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is Deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys.

Thus, convergent encryption allows the cloud to perform de duplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. However, previous de duplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized de duplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and his own privileges as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. For example, in a company, many different privileges will be assigned to employees.

In order to save cost and efficiently management, the data will be moved to the storage server provider (SSP) in the public cloud with specified privileges and the de duplication technique will be applied to store only one copy of the same file. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with specified privileges to realize the access control. Traditional de duplication systems

based on convergent encryption, although providing confidentiality to some extent; do not support the duplicate check with differential privileges. In other words, no differential privileges have been considered in the de duplication based on convergent encryption technique. It seems to be contradicted if we want to realize both de duplication and differential authorization duplicate check at the same time.

Literature survey

[1] In this paper, they proposed an architecture that provides secure deduplication storage resisting brute force attacks, and realize it in a system called dupLESS . It enables clients encrypted data with an existing service. The encryption for deduplicated storage can achieve performance and space saving close to that of using the storage service with plaintext data. [12] There is a mechanism to reclaim space from incidental duplication to make it available for controlled file replication. This mechanism convergent encryption, which enable duplicate files to be coalesced into the space file, even if the files are encrypted with different users keys. [15] It is a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master key.

[17] In this proposed system , they construct a private de duplication protocol based on the standard cryptographic assumptions is then presented and analyzed. They show that the private data de duplication protocol is probably secure assuming that the underlying hash function is collision-resilient, the discrete logarithm is hard and the erasure coding algorithm can erasure up to many fractions of the bits. [21] In this paper, they design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data. This way, data de duplication can be effective for popular data, whilst semantically secure encryption protects unpopular content. We show that our scheme is secure under the Symmetric External Decisional Diffie-Hellman Assumption.

System Design

The system architecture establishes the basic structure of the system, defining the essential core design features and elements that provide the framework for the system. The systems architecture provides the architects view of the users' vision for what the system needs to be and do, and the paths along which it must be able to evolve and strives to maintain the integrity of that vision as it evolves during detailed design and implementation.

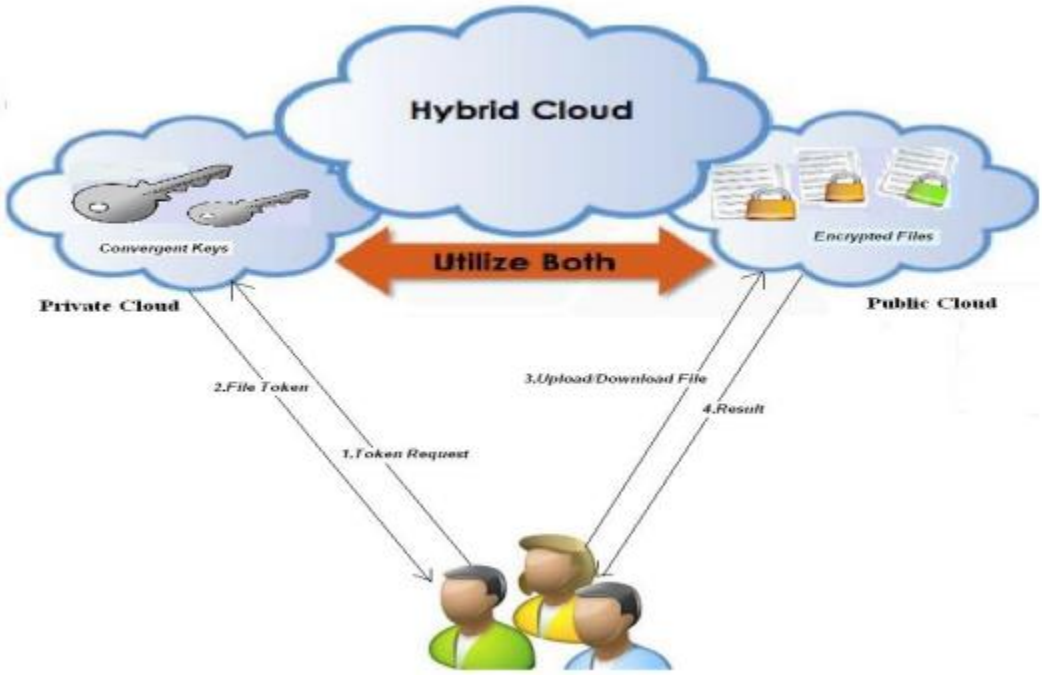

Fig 1: Architecture of the system

Module overview

User module

In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first. At the very least, you need to provide an email address, username, password, display name, and whatever profile fields you have set to required. The display name is what will be used when the system needs to display the proper name of the user.

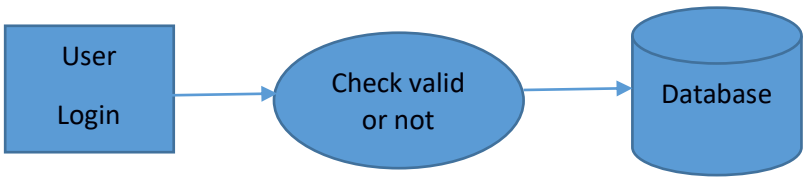

Fig 2: user module

Server start up and upload file

The user can start up the server after cloud environment is opened. Then the user can upload the file to the cloud.

Fig 3:server start up and upload file

Secure de duplication system

To support authorized de duplication the tag of a file F will be determined by the file F and the privilege. To show the difference with traditional notation of tag, we call is file token instead. To support authorized access a secret key KP will be bounded with a privilege p to generate a file Token. De duplication exploits identical content, while encryption attempts to make all content appear random; the same content encrypted with two different keys results in very different cipher text. Thus, combining the space efficiency of de duplication with the secrecy aspects of encryption is problematic.

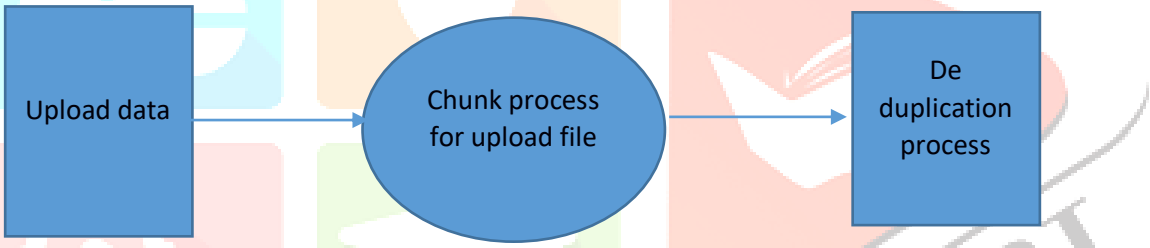

Fig 4: Secure de duplication system

Download file

After the cloud storage, the user can download the file based on key or token.Once the key request was received, the sender can send the key or he can decline it. With this key and request id which was generated at the time of sending key request the receiver can decrypt the message.

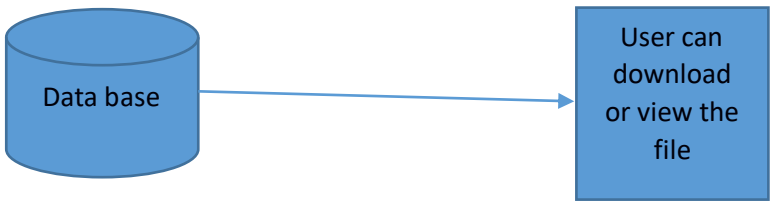

Fig 5: Download file

Proposed algorithm

Encryption

M3 encryption

- The algorithm itself is very complex and secure, but using it is as simple.
- The basic principle of this algorithm is character-remapping based on key self-mutation.
- The lifespan of a key is equal to the length of the key. This means that any state of the key will only be responsible for encrypting a part of the clear-text that is equal in length to the length of that version of the key before the key is self-mutated into a new version. This new version will then encrypt the next part of the clear-text, etc.
- In the overall process, you have a clear-key entered by the user which is diverted into 4 separate "threads" of different and constantly self-mutating keys. These 4 different keys are responsible for simultaneously converting the clear-text letters one letter at a time into cipher text by 2 different methods, these methods being: array remapping, and a sort of dynamic "substitution cipher". This whole process is repeated over and over again, re-encrypting everything a number of times before the cipher-text is finalized.
- An attempt of reversing the process by a potential attacker would require figuring out the end state of 4 different keys simultaneously going backwards one mutation-version at a time. As 2 of these keys are used for array remapping, it is necessary to get the whole of these keys per letter decoded in the clear-text.

Decryption algorithm

Data Encryption Standard (DES)

This stands for Data Encryption Standard and it was developed in 1977. It was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is 64 bits key size with 64 bits block size. Since that time, many attacks and methods have witnessed weaknesses of DES, which made it an insecure block cipher.

Algorithm:

function DES_Encrypt (M, K)

where M = (L, R)

M←–IP (M)

For round←–1 to 16 do

K←–SK (K, round)

L←L xor F(R,Ki)

swap(L, R)

end

swap (L, R)

M←IP-1 (M)

return M

End

Chunking Technique for Deduplication

Chunking is a process to split a file into smaller files called chunks. In some applications, such as remote data compression, data synchronization, and data de duplication, chunking is important because it determines the duplicate detection performance of the system. Content-defined chunking (CDC) is a method to split files into variable length chunks, where the cut points are defined by some internal features of the files. Unlike fixed-length chunks, variable-length chunks are more resistant to byte shifting.

Thus, it increases the probability of finding duplicate chunks within a file and between files. However, CDC algorithms require additional computation to find the cut points which might be computationally expensive for some applications. In our previous work (Widodo et al., 2016), the hash-based CDC algorithm used in the system took more process time than other processes in the de duplication system. This proposed work shows high throughput hash-less chunking . Instead of using hashes, RAM uses bytes value to declare the cut points. The algorithm utilizes a fix-sized window and a variable-sized window to find a maximum-valued byte which is the cut point. The maximum-valued byte is included in the chunk and located at the boundary of the chunk. This configuration allows RAM to do fewer comparisons while retaining the CDC property. We compared RAM with existing hash-based and hash-less deduplication systems. The experimental results show that our proposed algorithm has higher throughput and bytes saved per second compared to other chunking algorithms.

CONCLUSION

In this proposal, we are providing high security to the data which is stored in cloud storage and also secures the data from attackers by using encryption technique (blow fish algorithm) by using key sharing process between user and client .De duplication of the data can be done by chunk technique, here the data is be divided into fragments, comparison of data is done with the existing data. Here who owned the data of same copy is no need to store anymore and the data can be accessed by only authorized data of data copy, by this we achieve the data confidentiality, tag consistency, data reliability.

The main advantage of the proposed system was it supports both file level and block level. Here the main feature of the proposal is that data integrity, including tag consistency and also the de duplication is

possible by client and also server side .The data which is de duplicated is transferred to cloud storage library. It is more efficient, also reduces the space, cost efficient. Same copy of data is tried to transfer then de duplication takes place at the source. Less electricity, Fast recoveries, reduces the overall storage cost.

In this paper, the notion of authorized data de duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

Finally, we believe that cloud data storage security is still full of challenges and of paramount importance, and many research problems remain to be identified. In the proposed work de duplication is done for text and image it can be further extended for audio and video

References

1. M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
2. M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.
3. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.
4. M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
5. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.
6. P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.
7. M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
8. M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.
9. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

10. M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
11. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.
12. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In ICDCS, pages 617– 624, 2002.
13. D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 1992.
14. S. Halevi, D. Harnik, B. Pinkas, and A. ShulmanPeleg. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.
15. J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.
16. C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In Proc. of APSYS, Apr 2013.
17. W. K. Ng, Y. Wen, and H. Zhu. Private data deduplication protocols in cloud storage. In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.
18. R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and Communications Security, pages 81–82. ACM, 2012.
19. S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In Proc. USENIX FAST, Jan 2002. [18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In 3rd International Workshop on Security in Cloud Computing, 2011.
20. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29:38–47, Feb 1996.
21. J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secur data deduplication scheme for cloud storage. In Technical Report, 2013.