International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Deduplication in cloud storage by managing ownership dynamically

Bhanupriya BN[1], Dr.Siddaraju[2]

[1] *M. Tech Student , Department of Computer Science & Engineering, Dr. Ambedkar institute of Technology,Bangalore-560056,bhanumanvitha@gmail.com*

[2] *Professor and Head, Department of Computer Science & Engineering, Dr. Ambedkar institute of Technology,Bangalore-560056,*

Abstract – *In cloud storage services, by eliminating the redundant data and stores only a single copy of them by deduplication technology commonly used to reduce the space and bandwidth requirements. When multiple users outsource the same data in cloud storage the deduplication is most effective but it raises issues relating to security and ownership.In this paper we propose a novel server side deduplication scheme for encrypted data.It allows the cloud server to control access to outsourced data when ownership changes dynamically by secure ownership group key distribution and exploiting randomized convergent encryption.It would block data leakage not only to cancelled users even though they previously owned data but also to an honest but curios cloud srorage server.The proposed scheme confirms integrity of data against any tag inconsistency attack.*

***Key Words*: Datadeduplication, Cloud computing, Integrity of data, Dynamic ownership management.**

1. INTRODUCTION

Distributed computing gives versatile, low-cost, and area free online administrations ranging from basic reinforcement administrations to distributed storage infrastructures. The quick development of information volumes put away in the distributed storage has prompted an expanded demand for systems for sparing plate space and network bandwidth. To lessen asset utilization, many cloud capacity administrations, for example, Dropbox [1], Wuala[2], Mozy [3], and Google Drive [4], utilize a deduplication strategy, where the cloud server stores only a single duplicate of excess information and gives links to the duplicate as opposed to putting away other real copies of that information, paying little mind to what number of customers ask to store the information.

The investment funds are note worthy [5], and reportedly, business applications can accomplish plate and bandwidth reserve funds of over 90% [6]. However, from a security viewpoint, the common use of users data raises another challenge. As clients are worried about their private data, they may encode their information before outsourcing in request to shield information protection from unauthorized outside, and additionally from the cloud service provider [7],[8],[9]. This is supported by current security rends and various industry directions, for example, PCIDSS [10]. Be that as it may, traditional encryption makes deduplication unthinkable for the accompanying reason.

Deduplication strategies exploit information similarity to recognize similar information and decrease the storage space. Conversely, encryption calculations randomize the scrambled documents with a specific end goal to make ciphertext indistinguishable from hypothetically arbitrary information. Encryptions of similar information by various clients with different encryption keys brings about various ciphertexts, which makes it troublesome for the cloud server to determine whether the plain information are the same and deduplicate them. Say a client Alice scrambles a record M under her secretkeys Ka and stores its comparing ciphertext CA. Sway would store CB, which is the encryption of M under his mystery keys kB. At that point, two issues arise: (1) in what manner can the cloud server distinguish that the underlying file M is the same, and (2) regardless of the possibility that it can detect this, how might it enable both sides to recover the put away information, in view of their different mystery keys? Straightforward customer side encryption that is secure against a picked plaintext assault with haphazardly chosen encryption keys anticipates deduplication [11],[12].

One innocent arrangement is to enable every customer to encrypt the information with people in general key of the cloud storage server. At that point, the server can deduplicate the identified information by unscrambling it with its private key pair. In any case, this arrangement permits the cloud storage server to get the outsourced plain information, which may violate the protection of the information if the cloud server cannot be completely trusted.

2. CONTRIBUTION

We propose a deduplication scheme over encrypted data. The proposed scheme ensures that only authorized access to the shared data is possible, which is considered to be the most important challenge for efficient and secure cloud storage services in the environment where ownership changes dynamically. It is achieved by exploiting a group key management mechanism in each ownership group.

As compared to the previous deduplication schemes over encrypted data, the proposed scheme has the following advantages in terms of security and efficiency. First, dynamic ownership management guarantees the backward and forward secrecy of deduplicated data upon any ownership change. As opposed to the previous schemes, the data encryption key is updated and selectively distributed to valid owners upon any ownership change of the data through a stateless group key distribution mechanism using a binary tree.

The ownership and key management for each user can be conducted by the semi-trusted cloud server deployed in the system. Thus, the proposed scheme delegates the most laborious tasks of ownership management to the cloud server without leaking any confidential information to it, rather than to the users.

Second, the proposed scheme ensures security in the setting of PoW by introducing a re-encryption mechanism that uses an additional group key for dynamic ownership group. Thus, although the encryption key (that is the hash value of the file) is revealed in the setting of PoW, the privacy of the outsourced data is still preserved against outside adversaries, while deduplication over encrypted data is still enabled and data integrity against poison attacks is guaranteed. 2. A multimedia streaming service based on a pay-as-you-go policy in the cloud is a good example of services that have these backward and forward security requirements.

3. BACKGROUND

Gather correspondence can profit by IP multicast to accomplish versatile trade of messages. Be that as it may, there is a test of successfully controlling access to the transmitted information. IP multicast without anyone else's input does not give any instruments to preventing non group individuals to approach the gathering correspondence. In spite of the fact that encryption can be utilized to secure messages traded among gathering individuals, circulating the cryptographic keys turns into an issue. Specialists have proposed a few different approaches to gathering key administration. These methodologies can be isolated into three primary classes: concentrated gathering key administration conventions, decentralized structures and distributed key administration conventions. The three classes are depicted here and an insight given to their components and objectives. The territory of gathering key administration is then surveyed and proposed arrangements are ordered by those qualities. Gather correspondence applications can utilize IP multicast [Deering 1989] to transmit information to all n assemble individuals utilizing least assets. Effectiveness is accomplished in light of the fact that information parcels should be transmit-ted once and they navigate any connection between two hubs just once, subsequently sparing transmission capacity. This diverges from unicast based gathering correspondence where the sender needs to transmit n duplicates of a similar bundle. However adaptable, IP multicast does not give instruments to restrict the get to the information being transmitted to authorised assemble individuals just [Ballardie and Crowcroft 1995]. Any multicast-empowered host can send IGMP [Fenner 1997] messages to its neighbor switch and demand to join a multicast gathering. There is no validation or get to control implemented in this operation [Hardjono and Tsudik 2000]. The security challenge for multicast is in giving a compelling strategy to con-trolling access to the gathering and its information that is as productive as the underlying multicast.

An essential strategy for restricting access to data is through encryption and particular circulation of the keys used to scramble assemble data. An encryption calculation takes input information (e.g., a gathering message) and plays out some transformations on it utilizing a cryptographic key . This procedure produces a figured content. There is no simple approach to recoup the first message from the figured content other than by knowing the correct key. Applying such a strategy, one can run se-cure multicast sessions. The messages are ensured by encryption utilizing the picked key, which with regards to bunch communication is known as the gathering key . Just the individuals who know the gathering key can recoup the first message. Besides, the gathering may require that enrollment changes cause the gathering key to be invigorated. Changing the gathering key keeps another part from decoding messages traded before it joined the gathering. On the off chance that another key is dispersed to the gathering when another part joins, the new part can't disentangle past messages regardless of the possibility that it has recorded before messages encoded with the old key. Additionally, changing the gathering key keeps a leaving or ousted aggregate part from getting to the gathering correspondence. In the event that the key is changed when a part leaves, that part won't have the capacity to unravel

aggregate messages encoded with the new key. In any case, dispersing the gathering key to substantial individuals is a perplexing issue. Al-however rekeying a gathering before the join of another part is minor (send the new gathering key to the old gathering individuals encrypted with the old gathering key), rekeying the gathering after a part leaves is much more muddled. The old key can't be utilized to appropriate another one, on the grounds that the leaving part knows the old key. There-fore, a gathering key wholesaler must give another versatile component to rekey the gathering. A basic plan for rekeying a gathering with n individuals has the key dispersion focus (KDC) doling out a mystery key to every individual from the gathering.

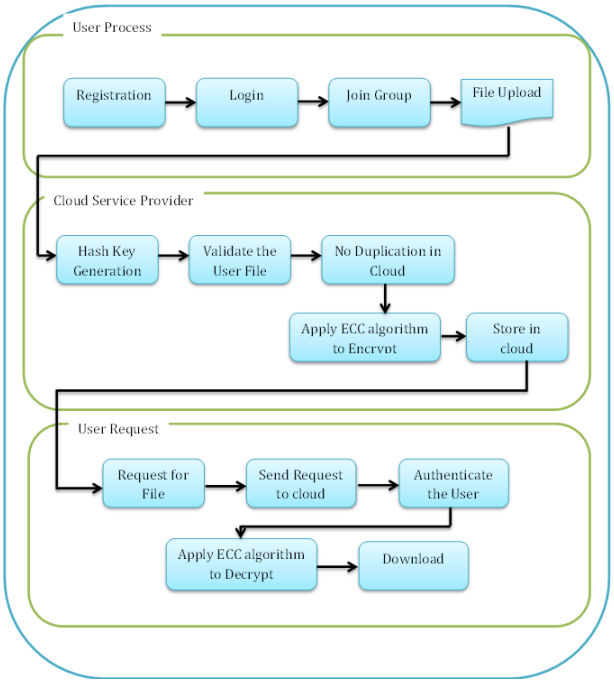
Assemble key administration is a troublesome undertaking in executing vast and dynamic secure multicast. In this paper, another plan is proposed in the premise of top to bottom examination of the necessities of the protected multicast and gathering key administration. The plan depends on the multicast assemble security design and multicast security gather key administration engineering proposed by IETF. This plan develops amass key in light of pairings and appropriates the gathering key utilizing HSAH work polynomial, and oversees aggregate key making utilization of the dynamic layering GCKS. The plan is better in security, bring down in calculation cost and correspondence cost. The investigation correlation demonstrates that the plan has solid versatility and effectiveness.

4. PROPOSED WORK

We propose a productive gathering key administration convention in conveyed assemble correspondence. This convention depends on Elliptic Curve Cryptography and the key length while giving securities at an indistinguishable level from that of different cryptosystems provides. We give the abnormal state security and stay away from the replication of record in the cloud benefit provider. In proposed framework, we utilizing hash capacity to produce key for the document .By utilizing hash capacity to maintain a strategic distance from the duplication in cloud. After that we applying cryptographic procedure for security purpose. We utilizing ECC calculation for encryption and decoding process.

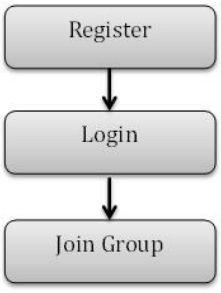
4.1 System architecture:

A framework design is a reasonable model that characterizes the structure, conduct, and more perspectives of a framework. An engineering portrayal is a formal depiction and portrayal of a framework, sorted out in a way that backings thinking about the structures and practices of the framework.



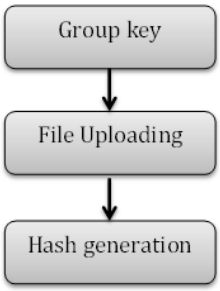
Registration and Login:

In our process , new user register the details and get the username and password for further process.Using Username and Password , user login into Group.Group generate key for the valid user and process inside the group under the valid key .



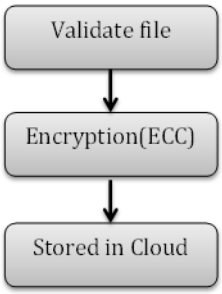
Join Group and File Upload:

In file upload process , user choose the file from the system and generate hash key for each file.Hash key generation is provided to avoid duplication of file to the cloud.If the file is already in cloud ,user should upload another file to cloud.



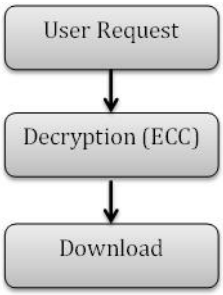
File encrypt and store into Cloud:

After the validation of file from the user with cloud , we apply cryptographic technique to improve the security level in cloud.For cryptographic technique , we using Elliptic Curve Cryptography(ECC) algorithm for encrypting the file.In Elliptic Curve Cryptography(ECC),it convert the file into binary format and store it in cloud.



User request and Download:

User send request to the cloud, cloud service provider decrypt the file .For cryptographic technique, we using Elliptic Curve Cryptography (ECC) algorithm for decrypting the file.Send the requested file to the user after validate the user.Then file will be downloaded in user location.



5. RESULTS

The project results show that deduplication is successfully achieved by generating hash keys to the files which we need to upload. By applying ECC algorithm can achieve data security.

5.1 Registration and login form

In this module Fig 5.1 we users register and if already registered will get login to the group.

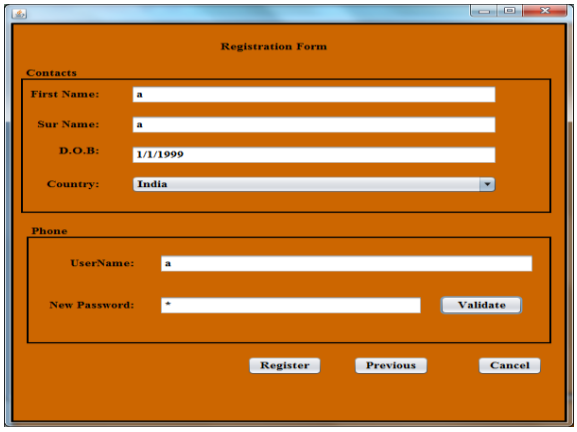


Fig -5.1: registration and login form

Group generate key for the valid user and process inside the group under the valid key .

5.2 Join group

This module Fig 5.2 gives the information about how to join the group

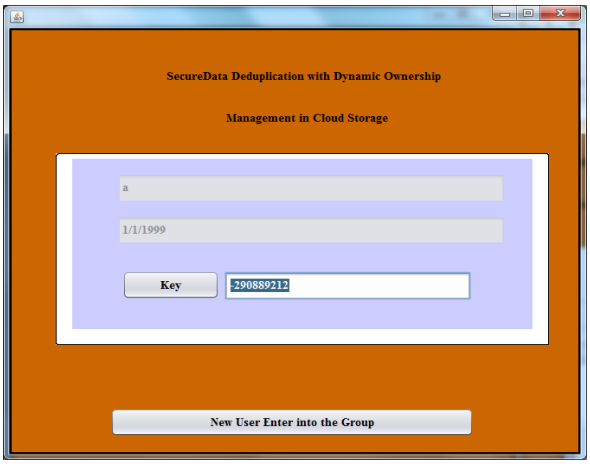


Fig -5.2: Join group

Would join to the group by entering valid key while it got generated when login .

5.3 Encryption of file in cloud

This module gives the information about how we are encrypting the file .

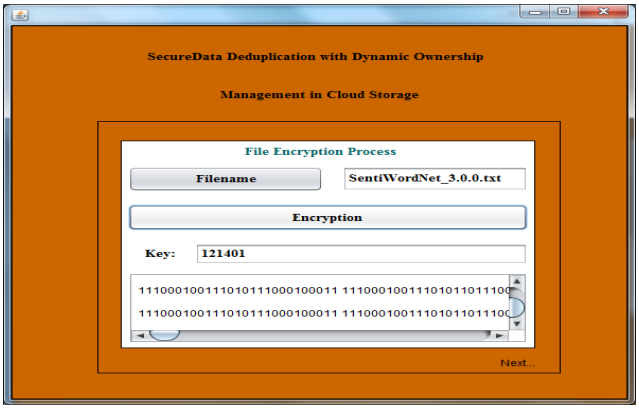


Fig -5.3: Encryption of file in cloud

5.4 Downloading file from cloud

This module gives the information about how to download the file corresponding path to download the file.

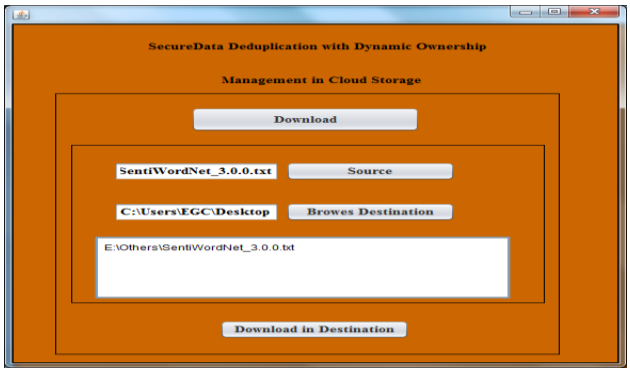


Fig -5.4: Downloading file from cloud

6. CONCLUSIONS

In asymmetric key distribution, elliptic curve cryptography key agreement is introduced. In this paper used elliptic curve cryptography and it provides much stronger security with smaller key size. By using hash key successfully can achieve deduplication in cloud storage. Eliptic curve cryptography is used for encryption and decryption process to decrease probability of attacking the file.

REFERENCES

[1] Dropbox, http://www.dropbox.com/.
[2] Wuala, http://www.wuala.com/.
[3] Mozy, http://www.mozy.com/.
[4] Google Drive, http://drive.google.com.
[5] D. T. Meyer, and W. J. Bolosky, "A study of practical deduplication,"Proc. USENIX Conference on File and Storage Technologies 2011, 2011.
[6] M. Dutch, "Understanding data deduplication ratios," SNIA Data Management Forum, 2008.
[7] W. K. Ng, W. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," Proc. ACM SAC'12, 2012.
[8] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication," Proc. StorageSS'08, 2008.
[9] N. Baracaldo, E. Androulaki, J. Glider, A. Sorniotti, "Reconciling end-to-end confidentiality and data reduction in cloud storage," Proc. ACM Workshop on Cloud Computing Security, pp. 21–32, 2014.
[10] P. S. S. Council, "PCI SSC data security standards overview," 2013.
[11] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services, the case of deduplication in cloud storage," IEEE Security & Privacy, vol. 8, no. 6, pp. 40–47, 2010.
[12] C. Wang, Z. Qin, J. Peng, and J. Wang, "A novel encryption scheme for data deduplication system," Proc. International Conference on Communications, Circuits and Ssytems (ICCCAS), pp. 265–269, 2010.
[13] Malicious insider attacks to rise, http://news.bbc.co.uk/2/hi/7875904.stm
[14] Data theft linked to ex-employees, http://www.theaustralian.com.au/australian it/datatheftlinked to-ex-employees/story-e6frgakx-1226572351953
[15] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in serverless distributed file system," Proc. International Conference on Distributed Computing Systems (ICDCS), pp. 617–624, 2002.
[16] P. Anderson, L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," Proc. USENIX LISA, 2010.
[17] Z. Wilcox-O'Hearn, B. Warner, "Tahoe: the least-authority filesystem," Proc. ACM StorageSS, 2008.
[18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, J. C. S. Lui, "A secure cloud backup system with assured deletion and version control," Proc. International Workshop on Security in Cloud Computing, 2011.
[19] J. Xu, E. Chang, and J. Zhou, "Leakage-resilient client-side deduplication of encrypted data in cloud storage," ePrint, IACR, http://eprint.iacr.org/2011/538.
[20] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," Proc. Eurocrypt 2013, LNCS 7881, pp. 296–312, 2013. Cryptology ePrint Archive, Report 2012/631, 2012.