International Journal of Research in Advent Technology (IJRAT) Special Issue
E-ISSN: 2321-9637
Sharadchandra Pawar college of Engineering, Dumbarwadi, Pune 410504, Organizes
National Conference "MOMENTUM-17", 04th & 05th February 2017
Available online at www.ijrat.org

Data Sharing and Deduplication on Encrypted Data For Cloud Environment

Sukanya Gunjal Sonali Thorat
Student, SPCOE, Department Of Computer Engineering, Dumbarwadi, Otur
Email: sukanyagunjal5@gmail.com

Abstract-Distributed computing offers another method for administration arrangement by re-orchestrating different assets over the Internet. The most critical and mostly known cloud administration is data storage. To safeguard the protection of data holders, data are frequently put away in cloud in an encrypted structure. In any case, encrypted data present new difficulties for cloud data deduplication, which gets to be significant for big data storage and preparing in cloud. Customary deduplication plans can't take a shot at encrypted data. Existing arrangements of encrypted data deduplication experience the ill effects of security shortcoming. They can't adaptably bolster data access control and revocation. Hence, few of them can be promptly conveyed practically speaking. In this paper, propose a plan for sharing data and deduplication on encrypted data put away for cloud in view of possession test and intermediary re-encryption. It incorporates cloud data deduplication with access control. We assess its execution in light of broad investigation and PC recreations. The outcomes demonstrate the unrivaled proficiency and adequacy of the plan for potential down to earth arrangement, particularly for big data deduplication in distributed storage.

Index Terms-Distributed computing;Distributed Storage; Security;Data deduplication;Big Data.

1. INTRODUCTION

Cloud computing provides unlimited virtualized resources through internet. By providing this service it hides all the implementation details as well as the entire platform. Cloud services have continuously management of these services. It gives more attention towards the utilization of storage as well as to store space on cloud. In this paper proposed data de-duplication technique, with more reliability. In data de-duplication process are removing unnecessary copies of data and save memory space. Previously, many de-duplication systems are implemented based on the policies such as, file level, block level deduplications and client-server side de-duplication. Proposed system provide a smart solution to manage data redundancy on cloud that arises due to massive data transaction by user of cloud. Proposed technique helps to save the space of cloud as well as it save bandwidth and make it more responsive. Many times there was same data stored by different users on the cloud and also this data have different encryption keys with respect to their owner hence it result into data redundancy. Different level of data protection is provided for cloud data, which required more bandwidth [7]. It is a bottleneck problem in current situation. Proposed Ramp secret sharing algorithm helps to solve this bottleneck problem. It preserve data secrecy during data encryption. Previously, PoW i.e. proofs of Ownerships are used to overcome the problem of hash signature of file. System will neglects uploading of same files. In system, metadata file is generated for each file to check whether user wants to upload the file is already present or not by comparing metadata file that is generated. System has two servers that help for check data sharing and privacy. These servers known as P –CSP and S-CSP, they allow user to deal with data by verifying from trustee.

System introduced P-CSP for locating relative file block address and SCSP maintains logical mapping. Each time trustee will verify user's identity while uploading data and data de-duplication is also checked [8]. While checking de-duplication our system avoids multiple transaction of file tags over network. System work better for hiding user's identity and neglecting data duplication. System implementing this system for file and block level de-duplication. Our system perfectly hides user identity and avoid data de-duplication. System, provides better data security as data is in encrypted format.

The rest of the paper has been organized as: section 2 shows the related work along with their limitations,

section 3 defined the proposed work of system. section 4 followed by conclusion and references.

2. RELATED WORK

To our knowledge, none of existing studies formally address the problem of convergent key management. J. Stanek, A. Sorniotti, E. Androulaki, [1], determined the popularity of the data; in different level of protection that is provided for the data in cloud. This system provides the guarantee of semantic security for unpolar data but less security and better security with appropriate bandwidth benefits for popular data. D. Harnik Pinkas, and Shulman-P [2], extracting de-duplication that is used as side channel. In this paper, authors studied about cross-user deduplication that provides guarantees of higher privacy with slightly reducing bandwidth savings in cloud storage.

J. Xu, E.-C. Chang, and J. Zhou [3], represents the deduplication for cross different users in which identical duplicated files from multiple users are detected and removed safely. W. K. Ng, Y. Wen, and H. Zhu [4], described private data deduplication protocols & also formalized the context of two-party computations. Private data deduplication protocol is secure in simulation-based framework. Yan, and D. Wang [5], were studied about Cauchy Reed Solomon coding for the construction of distribution matrix. Distribution matrix is constructed for encryption and decryption of data. High reliability provision mechanism i.e. R-ADMAD is proposed by D. Wang. RADMAD is dynamic and distributed recovery process in the cloud storage. The conflict between de-duplication and encryption was first discovered by Mihir Bellare, S. Keelveedhi distributed file system [6]. To overcome the conflict, they used the convergent encryption, in which the hash of the data was used as the encryption key, to make identical plaintexts be encrypted to the same cipher text regardless of which user they belonged to. However, Farsite only worked in the granularity of file-level, so that it could only save space with identical files. Storer etc. coalesced data at chunk-level, thus achieved de-dup with files that were merely similar as opposed to identical. Furthermore, they presented an on-line de-duplication scheme which saved not only storage space, but also network bandwidth. Storer etc. also used convergent encryption to guarantee the data confidentiality [9]. However, convergent encryption does leak information that a particular cipher text, and thus plaintext, already exists. So an adversary can get more potential knowledge by constructing a particular data chunk and checking whether it has already existed on the storage. What's worse, the encryption key is determined if a typical plaintext is given, i.e. the mapping from plaintext to cipher text is determinate but not random, which is less secure according to semantic security principal.

From the above discussion of previous de-duplication it is analyzed that they have certain overheads such as extra unnecessary key management is required which makes system more costly. There is need of such system which works on efficient management of encryption keys, efficient data sharing and data encryption over cloud. Hence proposed a system that provide all these beneficial features that will discussed in following sections.

3. PROPOSED WORK

A proposed a system called Data deduplication on Encrypted Data for Cloud. System propose a plan to rollback encrypted data put away in cloud in view of possession test and intermediary re-encryption. It incorporates cloud data deduplication with access control.

System access its execution in light of broad investigation and PC recreations [10]. The outcomes demonstrate the unrivaled proficiency and adequacy of the plan for potential down to earth arrangement, particularly for big data rollback in distributed storage.

The major contributions of this system are as follows:

1. System motivate to save cloud storage and preserve the privacy of data holders by proposing a scheme to manage encrypted data storage with rollback. Our scheme can flexibly support data sharing with rollback even when the data holder is offline, and it does not intrude the privacy of data holders [11].

2. System propose an effective approach to verify data ownership and check duplicate storage with secure challenge and big data support.

3. System integrate cloud data rollback with data access control in a simple way, thus reconciling data rollback and encryption.

4. System prove the security and assess the performance of the proposed scheme through analysis and simulation. The results show its efficiency, effectiveness and applicability.

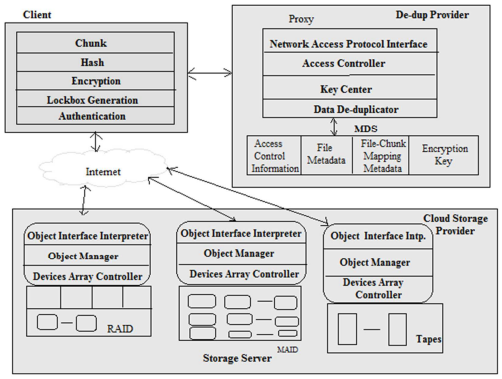

Fig. 1. System Architecture

Proposed system scheme can resist the attacks (Brute Force Attack) and achieve good performance without keeping data holders online all the time. It also ensures the confidentiality of stored data and supports digital rights management [12]. System provide more flexibility to achieve good performance. Proposed system having low storage cost of cloud. System support large amount of data that is big data.

4. PROPOSED ALGORITHM

4.1 Algorithm 1: AES Algorithm:

Algorithm finds the Proxy Re-Encryption : A PRE scheme is represented as a tuple of (possibly probabilistic) polynomial time algorithms (KG; RG; E; R; D). (KG; E; D) are the standard key generation, encryption, and decryption algorithms.

AES perform encryption of messages by using the following algorithm, which is then divided into 3 steps:

1. Key Generation:

Step 1: Take two distinct prime numbers p and q.

Step 2: Find n such that n = pq.
n represent the modulus for both the public and private keys.
Step 3: Find the totient of n, $\phi(n)$
$\phi(n)=(p-1)(q-1)$.
Step 4: Choose an e such that $1 < e < \phi(n)$, and such that e and $\phi(n)$ share no divisors other than 1 (e and $\phi(n)$ are relatively prime) [13].
Step 5: Determine d (using modular arithmetic) which satisfies the congruence relation
$de = 1 \pmod{\phi(n)}$.

In other words, take d such that de - 1 can be evenly divided by (p-1)(q-1), the totient, or $\phi(n)$.
This computation is perform using the Extended Euclidean Algorithm, since e and $\phi(n)$ are relatively prime and d is to be the modular multiplicative inverse of e [14].
d is remain as the private key exponent.

The public key has modulus n and the public exponent e. The private key has modulus n and the private exponent d, which is remain secret.

2. Encryption:

Step 1: Person A transmits his/her public key (modulus n and exponent e) to Person B, keeping his/her private key secret.
Step 2: When Person B wishes to send the message "M" to Person A, he first converts M to an integer such that $0 < m < n$ by using agreed upon reversible protocol known as a padding scheme.
Step 3: Person B computes, with Person A's public key information, the ciphertext c corresponding to
$c = me \pmod{n}$.
Step 4:Person B now sends message "M" in ciphertext, or c, to Person A.

3. Decryption:

Step 1: Person A recovers m from c by using his/her private key exponent, d, by the computation
$m = cd \pmod{n}$.
Step 2: Given m, Person A can recover the original message "M" by reversing the padding scheme.

This procedure works since,
$c = me \pmod{n}$,
$cd = (me)d \pmod{n}$,
$cd = mde \pmod{n}$.

By the symmetry property of mods we have that

mde = mde (mod n).

Since de = 1 + kϕ(n), we can write
mde = m1 + kϕ(n) (mod n),
mde = m(mk)ϕ(n) (mod n),
mde = m (mod n).

From Euler's Theorem and the Chinese Remainder Theorem, we can show that this is true for all m and the original message [16],[17].
cd = m (mod n), is obtained.

4.2 *Algorithm 2: Grant Access to Duplicated Data Algorithm:*

Algorithmic steps for Resource Allocation are as follows :

Input: pkj, Policy (ui), Policy (AP)

Output: SOUVA, SOPVA

Step 1: CSP requests AP to challenge ownership and grant access to duplicated data for uj by providing pkj.
Step 2:After ensuring data ownership through challenge, AP checks Policy (AP) and issues CSP rkAP→ui = RG (pkAP; skAP; pkj) if the check is positive [17],[18].
Step 3: CSP transforms (EpkAP; DEK) into E(pkj; DEK) if Policy ui authorizes uj to share the same data M encrypted by DEKi: (RrkAP→ui); E(pkAP; DEK)) = E(pkj; DEK).
Step 4: Data holder uj obtains DEKi by decrypting E(pkj; DEKi) with skj: DEKi = D(skj; E(pkj; DEKi)), and then it can access data M at CSP.

5. CONCLUSION

Propose technique provides data security using data encryption on Data for cloud environment. Proposed scheme can flexibly support for both that is for data updation and data sharing even when the data holders are offline. Encrypted data can be securely accessed because only authorized data holders can obtain the symmetric keys used for data decryption. In future work accuracy is calculatedand performance analysis and test showed that our scheme is secure and efficient under the described security model and very suitable for encrypted data for cloud environment.

Acknowledgments

I express my sincere thanks to my project guide Prof. G. S. Deokate who always being with presence & constant, constructive criticism to made this paper. I would also like to thank all the staff of Computer Department for their valuable guidance, suggestion and support through the paper work. Above all I express our deepest gratitude to all of them for their kind-hearted support which helped us a lot during paper work. At the last I thankful to my friends, colleagues for the inspirational help provided to me through a paper work.

REFERENCES

[1] J. Stanek, Sorniotti, E. Androulaki, and Kencl, "A secure data deduplication scheme for cloud storage," in Technical Report, vol. 9, no. 7, Feb 2013.
[2] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage." IEEE Security & Privacy, vol. 8, no. 6, pp. 40–47, 2010.
[3] J. Xu, E.-C. Chang, and J. Zhou, "Weak leakage-resilient clientside deduplication of encrypted data in cloud storage," in ASIACCS, 2013, pp. 195–206.
[4] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proc 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441–446.
[5] Liu, Y. Gu, Sun, B. Yan, and D. Wang, "R-admad: High reliability provision for large-scale de-duplication archival storage systems".
[6] Mihir Bellare, S. Keelveedhi, T. Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage", USENIX 2013.
[7] Zhe Sun, Jun Shen,"DeDu: Building a Deduplication Storage System over Cloud Computing." IEEE Transaction, May 2011.
[8] G. Wallace, et al., "Characteristics of backup workloads in production systems," in Proc. USENIX Conf. File Storage Technol., 2012, pp. 1–16.
[9] M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: Server aided encryption for deduplicated storage," in Proc. 22nd USENIX Conf. Secur., 2013, pp. 179–194.
[10] D. T. Meyer and W. J Bolosky, "A study of practical deduplication," ACM Transaction Storage, vol. 7, no. 4, pp. 1–20, 2012.

[11] Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication", Cryptology—EUROCRYPT, 2013, pp. 296–312.
[12] Zhaocong Wen, J. Luo, Huajun Chen†, Jiaxiao Meng, "A Verifiable Data Deduplication Scheme in Cloud Computing," International Conference on Intelligent Networking and Collaborative Systems, 2014.
[13] J. Li, Y. K. Li, X. F. Chen, P. P. C. Lee, and W. J. Lou, "A hybrid cloud approach for secure authorized deduplication," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216, May 2015.
[14] C. W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," J. Big Data, vol. 2, no. 1, pp. 1–32, 2015.
[15] N. X. Xiong, et al., "Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems," IEEE J. Select. Areas Commun., vol. 27, no. 4, pp. 495–509, 2009.
[16] Y. Z. Zhou, Y. X. Zhang, H. Liu, N. X. Xiong, and A. V. Vasilakos, "A bare-metal and asymmetric partitioning approach to client virtualization," IEEE Trans. Serv. Comput., vol. 7, no. 1, pp. 40–53, Jan.-Mar. 2014.
[17] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proc. 18th ACM Conf. Comput. Commun. Secur., 2011.
[18] R. D. Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proc. 7th ACM Symp. Inf. Comput. Commun. Secur., 2012.
[19] C. W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," J. Big Data, vol. 2, no. 1, pp. 1–32, 2015.
[20] L. F. Wei, et al., "Security and privacy for storage and computation in cloud computing," Inf. Sci., vol. 258, pp. 371–386, 2014,