CLOUD STORAGE DE-DUPLICATION AND ENCRYPTION

Joshi Vinay Kumarmr[1], V Ravi Shankar[2]

[1]CSE, M.TECH, GITAM University, Hyderabad
[2]Assistant Professor, GITAM University, Hyderabad

Abstract

De-duplication is one of the latest trend technologies in the current market because of its ability to reduce costs. But it comes in many different indication and organizations need to understand each one of them if they are to choose the one that is best for them. De-duplication can be applied to data on primary storage, backup storage, cloud storage, LAN and WAN transfers.Organizations frequently use de-duplication in backup and calamity recovery applications as well to avoid duplicate of similar data for savage of space in cloud.In this paper we attempt Data de-duplicationcombined with methods of implementing de-duplication, HASH BASED DE-duplication,Erasure Code, Threshold Proxy Re-encryption technique.

Keywords*—*de-deuplication; erasure code;threshold re-encryption technique; homomorphism; hash based DE duplication.

--***--

1. INTRODUCTION

Data de-duplication refers to the elimination of redundant data. De-duplication algorithms identify and delete duplicate, leaving only one copy (or 'single instance') of the data to be stored. However, indexing of all data is still retained should that data ever be needed. De-duplication[7] is able to reduce the required bandwidth and storage capacity, since only the unique data is stored. For example, a typical email system might contain 100 instances of the same 1 MB file attachment. If the email platform is backed up or archived, all 100 instances are saved, requiring 100MB storage space. With data De-duplication, only one instance of the attachment is actually stored; each subsequent instance is just referenced back to the one saved copy. In this example, a 100 MB storage and bandwidth demand could be reduced to only 1 MB. The practical benefits of this technology depend upon various factors, such as point of application, algorithm used, data type and data retention/protection policies. Let's take a look at some of the ways de-duplication technologies by where the de-duplication happens (server or client side), by granularity of the de-duplication (file or sub-file based), and finally by the logic of discovering duplicate data.

The most important problem in cloud computing is that large amount of storage space and security issues. One critical challenge of cloud storage is management of ever-increasing volume of data. To improve scalability, storage problem data de-duplication is most important technique and has attracted more attention recently. It is an important technique for data compression, it simply avoid the duplicate copies of data and store single copy of data. Data de-duplication take place in either block level or file level[12]. In file level approach duplicate files are eliminate, and in block level approach duplicate blocks of data that occur in non-identical files. De-duplication reduce the storage needs by up to 90-95% for backup application,68% in standard file system. Important issues in data de-duplication that security and privacy to protect the data from insider or outsider attack. For data confidentiality, encryption is used by different user for encrypt there files or data, usinga secrete key user perform encryption and decryption operation. For uploading file to cloud user first generate convergent key, encryption of file then load file to the cloud. To prevent UN authorize access proof of ownership protocol is used to provide proof that the user indeed owns the same file when de-duplication found. After the proof, server provides a pointer to subsequent user for accessing same file without needing to upload same file. When user want to download file he simply download encrypted file from cloud and decrypt this file using convergent key.

2. PRILIMINARIES

In this section we first define,what is data de-duplication, existing Convergent encryptiontechnique.

- **Data de-duplication :**This concept is a familiar one which we see daily, a URL is a type of pointer; when someone shares a video on YouTube, they send the URL for the video instead of the video itself. There's only one copy of the video, but it's available to everyone. De-duplication uses this concept in a more sophisticated, automated way.
- **Convergent encryption:**Convergent encryption is used to encrypt and decrypt file. User can derive the convergent key from each original data copy, then using that key encrypt data file. Also user derives tag for data copy to check duplicate data. Iftag is same then both files are same. Both convergent key and tag are independently derives. Convergent encryption [9],[10] also known as content hash keying, is used to produces identical cipher text from identical plaintext files. The simplest implementation of convergent encryption can be defined as: Alice derives the encryption key from her file F such that K = H(F), where H is a cryptographic hash function. Convergent

encryption scheme can be defined with four primitive functions:

- KeyGenCE(M) -> K is the key generation algorithm that maps a data copy M to a convergent key K;
- EncCE(K, M) ->C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C;
- DecCE (K, C) ->M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M ; and
- TagGen(M) -> T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).
- **Proof of ownership:** proof of ownership (PoW)[11][14] is a protocol enables users to prove their ownership of data copies to the storage server.PoW is implemented as an interactive algorithm run by user and storage server act as prove and verifier. The verifier derives a short value ϕ(M)from a data copy M . To prove the ownership of the data copy M , the prover needs to send ϕ to the verifier such that ϕ= ϕ(M).

Problems

- **Confirmation attack:** A more fundamental problem with convergent encryption is the confirmation attack. Here an attacker can check if a given key H is in the associative array. If the attacker can do this, he can also check if a given plaintext X is in the associative array by checking the presence of

  $$H=HB (E (HA(X),X)))$$

  If no preventative measures are taken, this could allow an attacker to confirm if the user is in possession of a certain file, for example a banned book or a pirated movie.
- **Offline brute-force attack[9]:** In convergent encryption it is easy to recognize the correct key. The correct key K will satisfy the equation

  $$K=HA (D (K,X'))$$

  While theoretically interesting, offline brute-force attacks on conventional symmetric cyphers are already possible in practice. Plaintexts often contain easily recognizable structures such as file headers. This can then be used as an effective heuristic to check if the correct key is found. Such an attack will work on any cipher where keys are significantly shorter than the messages, which in practice means anything but the one-time pad.
- **Learn the remaining attack:**Perhaps the most important of the possible attacks is the learn the remaining attack. Suppose an attacker knows most of the file, for example if the file is a PDF form where the user needs to fill in sensitive information, says a PIN code. The attacker can now create all possible versions of the file X and check if it matches a cipher text X′ by encrypting and comparing or using the identity

  $$X=D (HA(X),X'))$$

In fact, the attacker does not even need to have the value X′, it is sufficient to check if a given key H is in the associative array using the equation given above.

This attack is possible when X is known to be a member of a small set. The set of possible X's can then be exhaustively tried at the small cost of three hashing and one encryption operation per try. Or in information theoretical terms: when the relative entropy of the plaintext relative to the attacker is low.

3. RELEATED WORK

Priliminaries: In this section we try to present, methods implementing de-duplications, Hash based de-duplication.

- **Hash Based Deduplication:** De-duplication engineering ordinarily partitions information into littler lumps/squares and uses calculations to appoint every information lump an one of a kind hash identifier called a finger impression to each one pieces/pieces. To make the unique finger impression, it utilizes a calculation that figures cryptographic hash esteem from the information pieces/squares, paying little heed to the information sort. These fingerprints are put away in a list. The de-duplication calculation thinks about the fingerprints of information piece/square to those effectively in the record. In the event that the unique mark exists in the record, the information piece/square is supplanted with a pointer to information lump/square.On the off chance that the unique mark does not exist, the information is composed to the plate as another novel information piece .Now we examine about the methodology or strategy used to do the de-duplication. Diverse information de-duplication items use distinctive routines for separating the information into components or pieces or squares, however every item utilizes some system to make a mark or identifier or finger impression for every information component. As indicated in the underneath figure, the information store contains the three remarkable information components A, B, and C with an unique mark. These information component mark qualities are contrasted with recognize copy information. After the copy information is recognized, one duplicate of every component is held, pointers are made for the copy things, and the copy things are not put away. The fundamental ideas of information de-duplication are outlined below

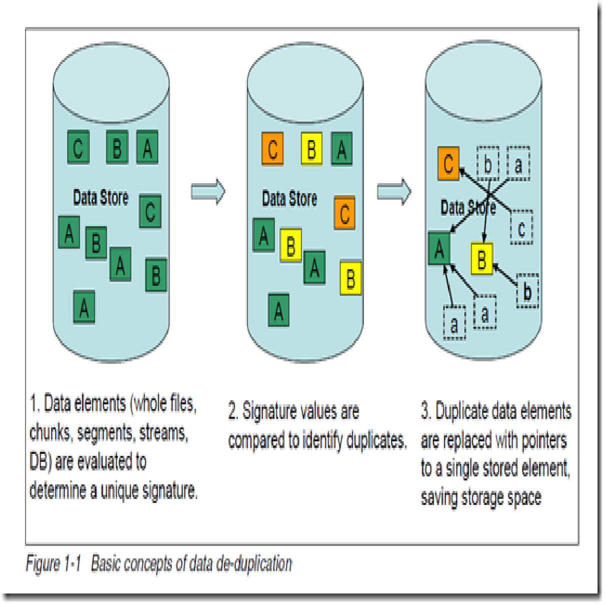

C B A
Data Store
C
A B A B
A
C B A
Data Store
C
A B A B
A
C b a
Data Store
c
A B b
a a
1. Data elements (whole files, chunks, segments, streams, DB) are evaluated to determine a unique signature.
2. Signature values are compared to identify duplicates.
3. Duplicate data elements are replaced with pointers to a single stored element, saving storage space
Figure 1-1 Basic concepts of data de-duplication

Fig. 1 Basic concepts of data de-duplication

1 – Hash based De-duplication

Hash based data de-duplication methods use a hashing algorithm to identify "chunks" of data. Commonly used algorithms are Secure Hash Algorithm 1 (SHA-1)[7] and Message-Digest. Algorithm (MD5). When data is processed by a hashing algorithm, a hash is created that represents the data. A hash is a bit string (128 bits for MD5 and 160 bits for SHA-1) that represents the data processed. If you processed the same data through the hashing algorithm multiple times, the same hash is created each time.

Here are some examples of hash codes: **MD5 – 16 byte long hash** – # echo "The Quick Brown Fox Jumps Over the Lazy Dog" | md5sum9d56076597de1aeb532727f7f681bcb0– # echo "The Quick Brown Fox Dumps Over the Lazy Dog" | md5sum5800fccb352352308b02d442170b039d**SHA-1 – 20 byte long hash**– # echo "The Quick Brown Fox Jumps Over the Lazy Dog" | sha1sumF68f38ee07e310fd263c9c491273d81963fbff35– # echo "The Quick Brown Fox Dumps Over the Lazy Dog" | sha1sum d4e6aa9ab83076e8b8a21930cc1fb8b5e5ba2335

Hash based de-duplication breaks information into "pieces", either settled or variable length, and methods the "piece" with the hashing calculation to make a hash. On the off chance that the hash as of now exists, the information is considered to be a copy and is not put away. In the event that the hash does not exist, then the information is put away and the hash list is overhauled with the new hash. In Figure 1-2, information "lumps" A, B, C, D, and E are handled by the hash calculation and makes hashes Ah, Bh, Ch, Dh, and Eh; for purposes of this sample, we accept this is all new information. Later, "pieces" A, B, C, D, and F are handled. F creates another hash Fh. Since A, B, C, and D produced the same hash, the information is attempted to be the same information, so it is not put away once more. Since F produces another hash, the new hash and new information are put away.

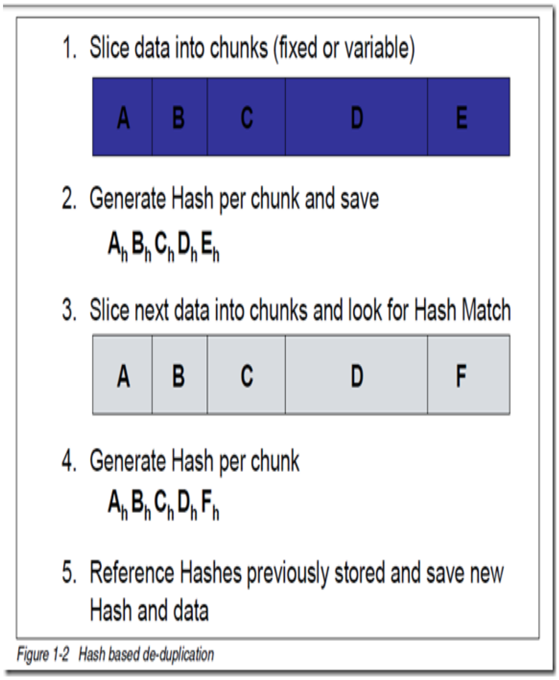

1. Slice data into chunks (fixed or variable)
A B C D E
2. Generate Hash per chunk and save
A_h B_h C_h D_h E_h
3. Slice next data into chunks and look for Hash Match
A B C D F
4. Generate Hash per chunk
A_h B_h C_h D_h F_h
5. Reference Hashes previously stored and save new Hash and data
Figure 1-2 Hash based de-duplication

Fig 2 Hash based de-duplication

3.1 Fixed-Length or Fixed Block

In this information de-duplication calculation, it softens the Information up to lumps or square, and the piece size or piece limits is Altered like 4kb, or 8kb and so forth. What's more the square size never shows signs of change. While distinctive gadgets/arrangements may utilize diverse square sizes, the piece size for a given gadget/arrangement utilizing this strategy stays consistent. The gadget/arrangement dependably figures a finger impression or signature on a settled piece and checks whether there is a match. After a square is handled, it progresses by precisely the same size and take an alternate piece and the methodology rehashes.

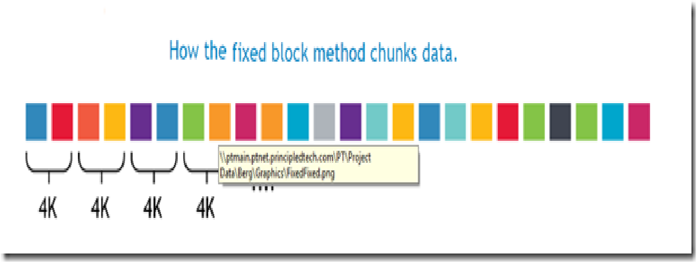

How the fixed block method chunks data.
\\ptmain.ptnet.principledtech.com\PT\Project Data\Berg\Graphics\FixedFixed.png
4K
4K
4K
4K

Fig 3 Fixed block method chunks data

- **Advantages**
  Requires the base CPU overhead, and quick and straightforward

- **Disadvantages**
  Since the piece size or square limits is Altered, the principle restriction of this methodology is that when the information inside a document is moved, for instance, when adding a slide to a Microsoft PowerPoint deck, all ensuing squares in the record will be revised and are liable to be considered as unique in relation to those in the first document. Littler piece size

give preferable de-duplication over extensive ones, however it takes all the more preparing to de-duplicate. Bigger square size give low de-duplication, however it takes less preparing to de-duplicate. So what really matters is Less stockpiling reserve funds and not productive.

3.2 Variable-Length or Variable Piece

In this information de-duplication calculation, it softens the Information up to pieces or square, and the piece size or square limits are variable like 4kb, or 8kb or 16kb and so on. What's more the square size changes rapidly amid the whole process. The gadget/arrangement dependably computes a finger impression or signature on a variable piece size and checks whether there is a match. After a piece is prepared, it propels by taking an alternate square size and take an alternate squares and the methodology rehashes

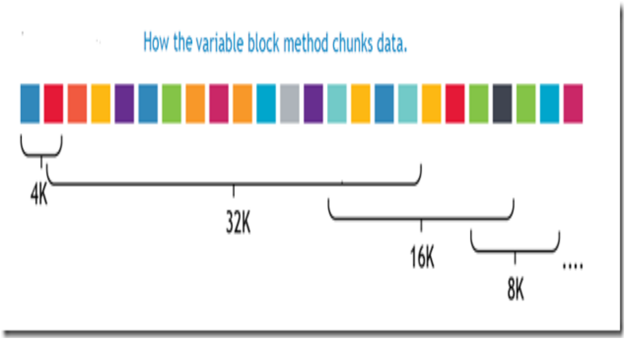


Fig 4 Variable block method

- **Advantages**
  Higher de-duplication ratio, high storage space savings.

- **Disadvantages**
  While the variable piece de-duplication may yield somewhat preferable de-duplication over the altered square de-duplication approach, it does oblige you to pay a cost. The cost being the CPU cycles that must be used in attempting to focus the document limits. The variable square approach obliges more transforming than altered piece in light of the fact that the entire document must be filtered, one byte at once, to distinguish square limits. Look at the accompanying sample focused around the accompanying sentence will clarify in subtle element: "DE duplication innovations are getting to be more critical now."Recognize how the variable square de-duplication has some crazy piece sizes. While this does not look excessively productive contrasted with altered square, look at what happens when I make an amendment to the sentence. it would appear that I utilized "a" when it ought to have been 'and'. Time to change the record: "de-duplication advances are getting to be more essential now." Document –> Spare After the document was changed and de-duplicated, this is the thing that the stockpilingsubsystem.

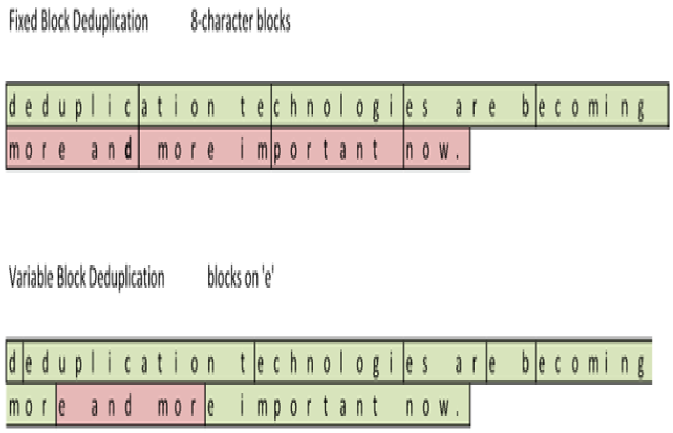

Fixed Block Deduplication
8-character blocks
deduplication technologies are becoming
more and more important now.
Variable Block Deduplication
blocks on 'e'
deduplication technologies are becoming
more and more important now.

Fig 5 File based de-duplication

The red areas speak to the changed obstructs that have changed. By including a solitary character in the sentence, a 'd', the sentence length moved and more squares abruptly changed. The Altered Close arrangement saw 4 out of 9 pieces changed. The Variable Shut arrangement saw 1 out of 9 pieces changed. Variable piece de-duplication winds up giving a higher stockpiling thickness and great storage room savings.so here the information de-duplication ratio degree can be given a

$$DR = \frac{\text{Total Data before Reduction}}{\text{Total Data after Reduction}}$$

4. METHODS OF DEDUPLICATION

PRILIMINARIES: In this section we just give a brief description of seven different methods of de-duplication implementation and small description about the each method by diagrammatic views.

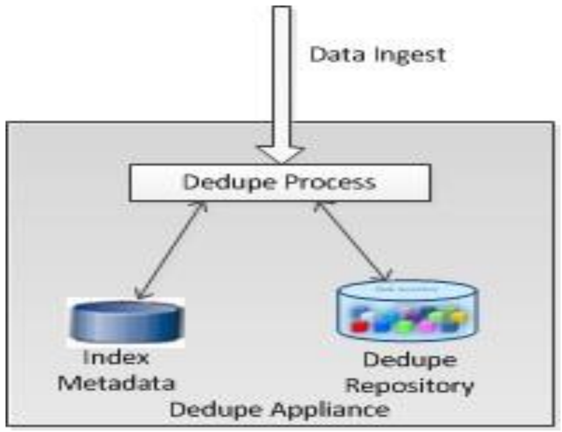
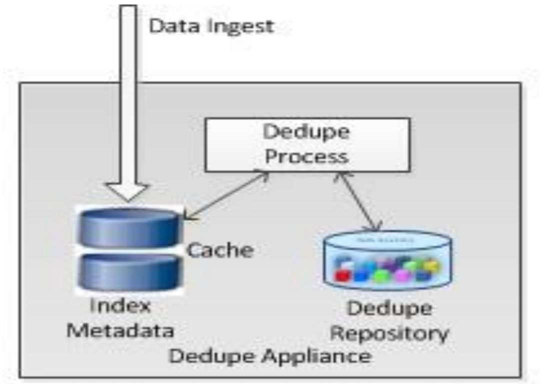
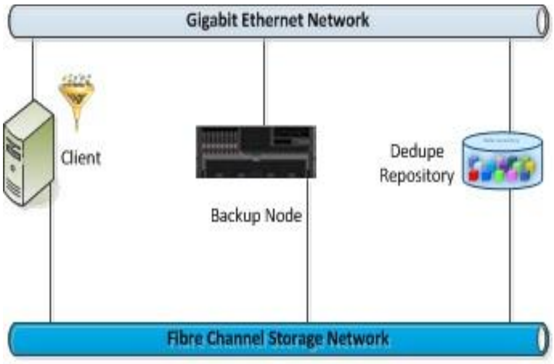


Fig 6 Inline de-duplication



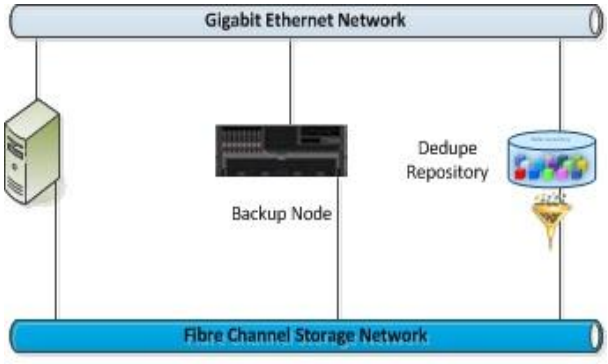
Post Process Dedupe: Data is deduped after it is stored in fast cache

Fig 7 post-process de-duplication



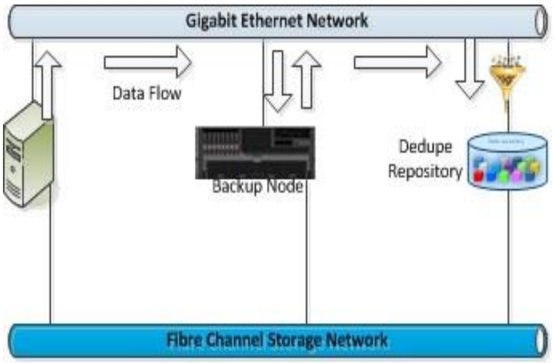
Client Based Dedupe: Client performs the dedupe processing

Fig 8 Client-side de-duplication



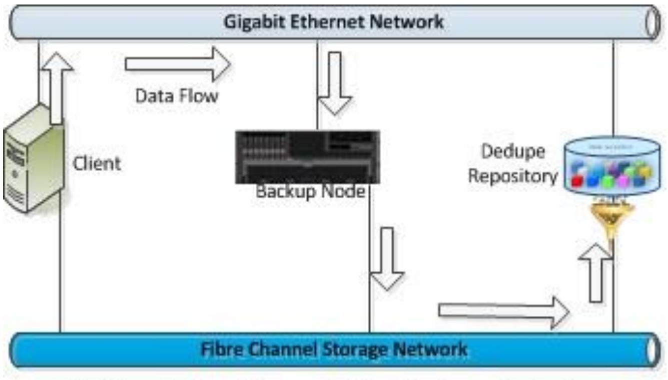
Target Based Dedupe: All dedupe processing is done at the target

Fig 9 Target-based de-duplication



NAS Based Dedupe: Client sends data to backup node. Backup node sends data to target over the Network and deduped

Fig 10 NAS based de-duplication



SAN Based Dedupe: Client sends data to backup node. Backup node sends data to target over the SAN and is deduped

Fig 11 SAN based de-duplication

- Inline de-duplication[1] -- Data is de-duplicated in real time as it is stored.
- Post-process de-duplication[2] -- Data is stored first, and de-duplicated later.
- Client-side de-duplication[3] -- Data is de-duplicated at the source.
- Target-based de-duplication -- Data is de-duplicated after sending it to a target.
- Network attached storage (NAS)[4]-based de-duplication -- Data is sent to the de-duplicated target over an IP network.
- Storage area network (SAN)[5]-based de-duplication -- Data is sent to the de-duplication target over Fiber Channel (FC).
- Global de-duplication [6] -- Data is de-duplicated across an infrastructure over all transport protocols.

4. IMPLEMENTATION

Priliminaries: In this section we discuss about new encryption algorithm used that isErasure Code, Threshold Proxy Re-encryption technique on replication to provide sensitivity as well confidentiality on de-duplications.

- **Erasure - correcting code** is used in file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the holomorphic token with distributed verification of erasure coded data, this scheme achieves the integration of storage correctness insurance and data error localization. Challenge Response protocol is used to provide the localization of data error. In cloud data storage, a user stores the data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of erasure-correcting code to tolerate faults or server crash.
- **Threshold Proxy Re-encryption technique:** a fundamental approach of threshold PRE scheme is for secure computation. This scheme performs arbitrary computations on encrypted data without decrypting it. Threshold PRE technique has multiplicative holomorphic property. A multiplicative holomorphic encryption scheme supports the encoding operation over encrypted messages and forwarding operations

over encrypted and encoded messages. The three properties exhibited by Threshold PRE scheme are

- **Homomorphism:** Given two cipher texts c1 and c2 on plaintexts p1 and p2 respectively, one can obtain the cipher text on the plaintext p1+p2 and/or p1.p2 by evaluating c1 and c2 without decrypting cipher texts.
- **Proxy re-encryption:** Transforming encrypted data of one user to encrypted data of target user.
- **Threshold decryption:** By dividing the private key into several pieces of secret shares, all clients can work together to decrypt the cipher text – the output of the function.

5. CONCLUSION AND FUTURE WORK

In this paper we discussed the basic idea about the methods of de-duplications as well types for their implementations and also discussed about the HASH based de-duplication. Further taking security as note just gave an idea about Erasure Code[13], Threshold Proxy Re-encryption technique[13]. In near feature work is done to implement the de-duplication technology by using a new encryption algorithm in a cloud.

REFERENCES

[1]. Srinivasan, K., Bisson, T., Goodson, G. R., &Voruganti, K. (2012, February). iDe-dup: latency-aware, inline data de-duplication for primary storage. In *FAST*(Vol. 12, pp. 1-14).

[2]. Kathpal, A., John, M., &Makkar, G. (2011). Distributed Duplicate Detection in Post-Process Data De-duplication.HiPC.

[3]. Stringham, Russell R. "Client side data de-duplication." U.S. Patent 7,814,149, issued October 12, 2010.

[4]. Nguyen, T., Raymond, R. M., &Leonhardt, M. L. (2003). *U.S. Patent No. 6,658,526*. Washington, DC: U.S. Patent and Trademark Office.

[5]. Clements, A. T., Ahmad, I., Vilayannur, M., & Li, J. (2009, June). Decentralized De-duplication in SAN Cluster File Systems. In *USENIX Annual Technical Conference* (pp. 101-114).

[6]. Dutch, M. (2008, June). Understanding data de-duplication ratios. In *SNIA Data Management Forum*.

[7]. S. Quinlan and S. Dorward.Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan 2002.

[8]. 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.

[9]. M. Bellare, S. Keelveedhi, and T. Ristenpart.Message-locked encryption and secure de-duplication. In *EUROCRYPT*, pages 296–312, 2013

[10]. 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.

[11]. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg.Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis,and V. Shmatikov, editors, *ACM Conference on Computer andCommunications Security*, pages 491–500. ACM, 2011.

[12]. J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure de-duplication with efficient and reliable convergent key management. In *IEEETransactions on Parallel and Distributed Systems*, 2013.

[13]. Inbarani, W. S., Moorthy, G. S., & Paul, C. K. C. (2013). An Approach for Storage Security in Cloud Computing-A Survey. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *2*(1), pp-174.