ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 11, November 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Enabling Cloud Storage Auditing and Eliminating De-Duplicated Data

Yandrapu Gayatri[1]
M.Tech Scholar
Department of Computer Science & Engineering,
Pydah College of Engineering and Technology, Gambheeram,
Visakhapatnam, AP – India

A.V.D.N.Murthy[2]
Assistant Professor
Department of Computer Science & Engineering,
Pydah College of Engineering and Technology, Gambheeram,
Visakhapatnam, AP – India

Dr. Ramesh Challagundla[3]
Professor& Principal
Department of Computer Science & Engineering,
Pydah College of Engineering and Technology, Gambheeram,
Visakhapatnam, AP – India

Abstract: ***Cloud storage is a strategy of storage beyond an interface where the storage space is manipulated and maintained on demand. This technology gives benefits by many features like data back-up and archival, you do not have of retaining hardware resources, greater data accessibility, and so data de-duplication is an important feature in cloud hosting storage. As a lot of data is dynamically updated and trapped in today's scenario, the previous methods used for checking static data honesty can no longer be used on analyze the integrity of the stored dynamic data in the cloud. Inside the existing system, focused on key management in a built in key exposure resilient system. In this paper, De-duplication process identifies and gets rid of the repeated data in the backup storage, ultimately increasing the network band width. Providing security along with the de-duplication process is a challenging task. We have introduced the Blowfish algorithm, it is a symmetric block cipher it is used for encryption and decryption purpose and blowfish is ideal for both domestic and exportable use to existing encryption algorithms. And strategy of de duplication of data wherein the built-in key exposure system will examine the duplicity of data and get rid of the obsolete one using whirlpool algorithm.***

Keywords: ***Data storage, cloud storage auditing, de-duplication, cloud computation, Blowfish algorithm and Whirlpool algorithm.***

I. INTRODUCTION

The Cloud computing is an model to provide access to processing resources and applications on the Internet Cloud processing platform offers the network resources and storage space to the remote users. Information can be utilized by an individual at whenever and from everywhere via Internet. So the end user and his data need not to be on same physical location. In addition, user even will not require to manage the genuine resources. Cloud processing permits the users to reach distributed resources by providing services as per user necessity within the network to perform operations. Managing and implementing certain applications developed by the users can be done through cloud processing services. As a result of computing high scalability and availability it improve response time, which results in top rated and provides services to its users on a huge scale. Cloud hosting computing era have plenty of research issues. De-duplication is one of them. It is a compression technique which identifies and locates the duplicate data. It then eliminates repeat copies of repeating data and saves the space for data that needs to be physically store. Hence, the two main features of data de-duplication are Reduction in Storage Portion and Efficient Volume of Replication.

In storage servers, De-duplication criteria detect redundant data by creating cryptographic hash of the data to be stored. Hash is a fixed length representation of arbitrary size message. Hashing reduces the complexity of comparing two data portions or data record because the size of the hash is much smaller as compared with the size of the data. For each newly arriving record, server first calculates its hash unsecured personal and searches this hash signature in already preserved hash index in the system. If the storage space finds that the access is available for this signature in the hash index (i. e. data has already been stored), then, in host to storing again, server only creates a reference for this, which points to the location of block on the disk. Otherwise server stores this record on the disk and adds an entry for its hash signature in the hash index.

Data de-duplication techniques are broadly employed by storage servers to get rid of the possibilities of holding multiple copies of the data. De-duplication identifies replicate data portions going to be trapped in storage systems also removes duplication in existing stored data in storage systems. Hence deliver a significant cost conserving.

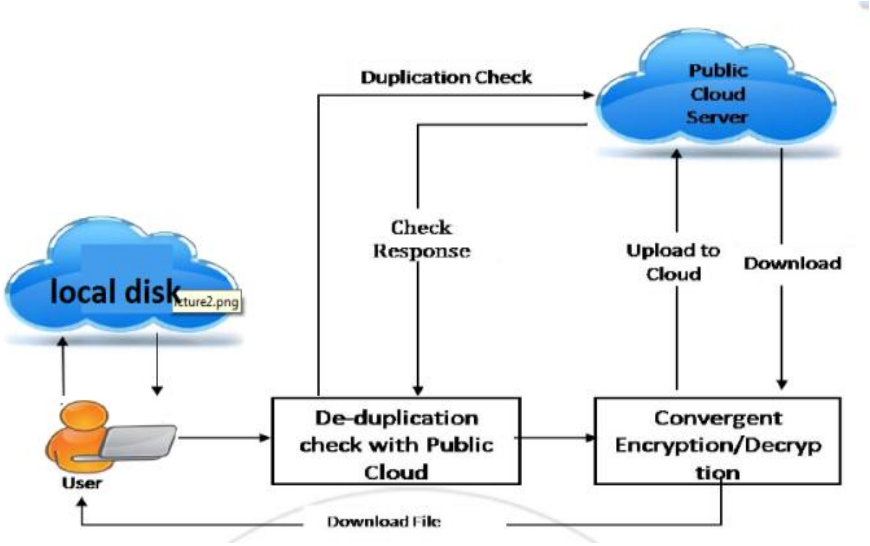


Fig1: De-duplication Process

The data transformation process of the Blowfish algorithm for Encryption and Decryption, respectively. The details and working of the blowfish algorithm given below.

Blowfish is a symmetric block cipher which can be effectively used for encryption and Protecting of data. It requires a variable-length key, from 32 bits to 448 portions, making it ideal for securing data. Blowfish was designed in 1993 by Bruce Schneider as a fast, free substitute for existing encryption algorithms. It is ideal for applications where key does not change frequently, like a communications website link or an automated data file encrypted.

Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block dimensions are 64 portions, and the main element can be any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very effective on large microprocessors.

By using whirlpool algorithm eliminating de-duplication. Whirlpool, which is a block-cipher-based secure hash function. Whirlpool produces a hash code of 512 bits for an input message of maximum length below 2256 bits. The underlying block out cipher, based on the Advanced Encryption Standard (AES), takes a 512-bit key and operates on 512-bit blocks of plaintext.

II. Related Work

World is adapting digital technologies, Switching from legacy way of Digital approach. Data is the primary thing which exists in digital form all over the place. To store this significant data, the storage strategy should be efficient as well as intelligent enough to obtain the redundant data to save. Data de-duplication techniques are extensively employed by storage space servers to get rid of the options of storing multiple replications of the data. De-duplication identifies duplicate data

servings going to be kept in storage systems also eliminates duplication in existing stored data in storage systems. Hence yield a significant cost saving.

Based on the data granularity, de-duplication strategies can be categorized into two main categories: file-level de-duplication and block-level de-duplication, which is nowadays the most frequent strategy. In block-based de-duplication, the block size can be fixed or adjustable. The location at which de-duplication is performed: if data are de-duplicated at the client, it is called source-based de-duplication, otherwise target-based. In source-based de-duplication, the client first hashes each data portion the wishes to publish and sends these results to the storage service provider to check whether such data are already stored: thus only "unde-duplicated" data segments will be actually uploaded by the end user. While de-duplication at the client area can achieve bandwidth savings, it however can make the device susceptible to side-channel attacks whereby opponents can immediately discover if the certain data is stored or not.

Types of de-duplication strategies

According to the operational area, data de-duplication strategies can be classified into two approaches.

File-level De-duplication

File level de-duplication, as the name suggests, is always performed over an one file. Value of two or more files can determine that the files are very similar, as shown fig2.

Fig2: File Level De-Duplication

Block-level De-duplication

Block level de-duplication is conducted over blocks. Firstly it splits the files into obstructions and stores simply an one copy of each block out. Fixed-sized blocks or variable-sized chunks can be used with block-level de-duplication as shown in fig3.

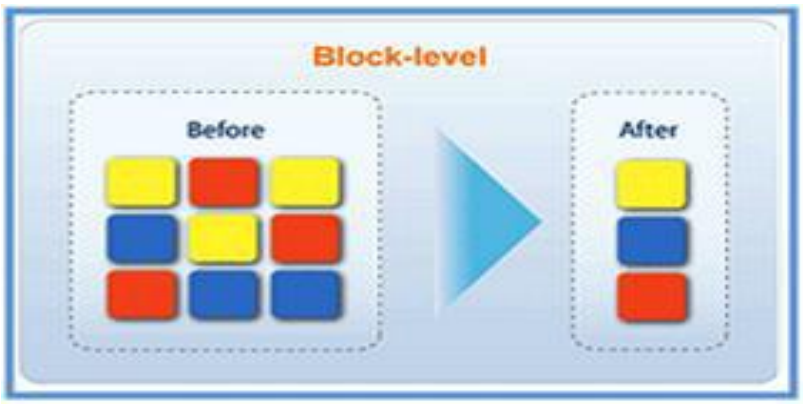


Fig3: Block Level De-Duplication

A. Blowfish Algorithm:

A great encryption algorithm plays an important role in protecting the information in storing or transferring it. The security algorithms are categorized into Symmetric (secret) and asymmetric (public) keys encryption.

- In Symmetric key encryption or secret key encryption, just one key is employed for both encryption and decryption of information.
- In asymmetric key encryption or public key encryption uses two secrets, one for encryption and other for decryption.

Blowfish is a symmetric block encryption criteria designed in consideration with,

- Fast: It encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles every byte.
- Compact: It can run within just 5K of memory.
- Simple: It uses addition, XOR, lookup stand with 32-bit operands.
- Secure: The key length is variable, it can be in the product range of 32~448 bits: default 128 portions key length.
- It is suited for applications where the key does not change often, like communication hyperlink or an computerized record encrypted.
- Unpatented and royalty-free.

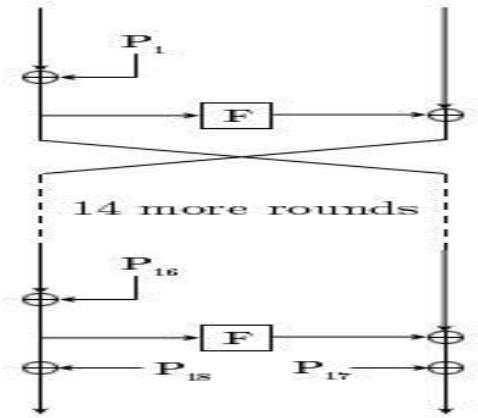


Fig4: The Feistel structure of Blowfish

Blowfish algorithm is a 16-round Feistel cipher and uses large key-dependent S-boxes. Here is the visual representation of this encryption algorithm.

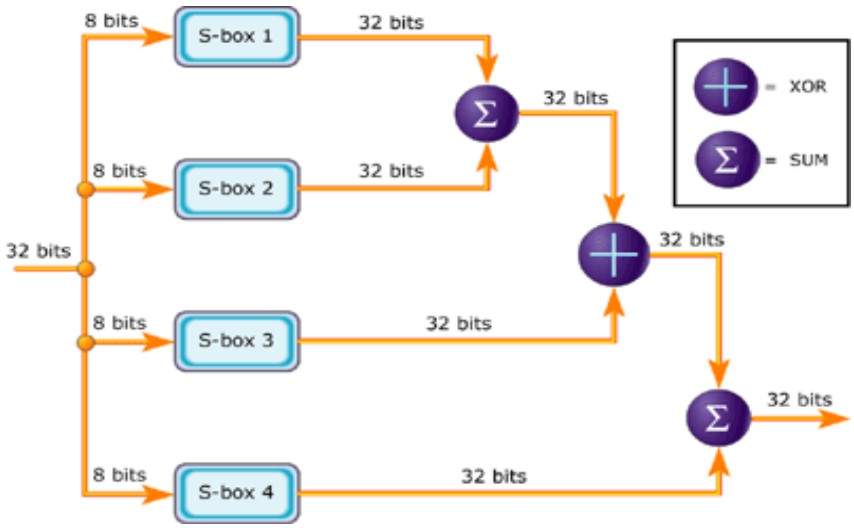


Fig5: S-boxes

The diagram to the left side shows the action of Blowfish. Each line signifies 32 bits. The criteria keep two subkey arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit suggestions and produce 32-bit outcome. One entry of the P-array can be used every circular, and after the last round, each half of the data block is XORed with one of the two remaining empty P-entries. The diagram to the right shows Blowfish's F-function. The function divides the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 232 and XORed to produce the last 32-bit output.

Since Blowfish is a Feistel network, it can be upside down simply by XORing P17 and P18 to the cipher text block, then using the P-entries in opposite order.

B. Whirlpool Algorithm:

The Whirlpool hash function needs a| message of any span less than 2256 octet and returns an absorb of 512 bytes. Initially, the message is cracked up into blocks of 512 bytes. If the message cannot be break up evenly, then the last block is appended with a '1, ' then a range of '0's, and finally the span of the initial message as a 256bit integer, to make that last block outcome to 512 octet. These blocks are then compressed with a MiyaguchiPreneel schema, then fed into 10 rounds of Whirlpool's block cipher 'W. ' W takes the outcome of the previous procedure and the current block as keys, then XORs the output of W and the two inputs to generate the consequence of the block.

Whirlpool Logic

The encryption key input for each and every iteration i is the intermediate hash Hi-1value from the previous version, and the plaintext is the existing message block mi. The outcome for this iteration? Involves the bitwise XOR of the current message block, the intermediate hash value from the previous iteration, and the outcome from W.

The algorithm takes as type a message with a maximum length of less than 2256 bits and produces as output a 512-bit message digest. The input is highly processed in 512-bit blocks. Figure depicts the overall digesting of a message to produce a digest.

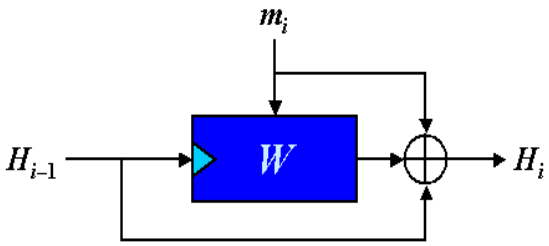


Fig6: Whirlpool logic

III. Existing System and Proposed System

Existing System:

Existing system destroys the client's key exposure in cloud storage auditing. Though many research works about cloud storage auditing have recently been required for recent years, a critical security problem exposure problem for cloud storage auditing, has continued to be unexplored in previous studies. While all existing protocols give attention to the faults or dishonesty of the cloud, they may have avoided the possible weak sense of security and/or low security settings at the client.

Disadvantages:

- Though many research works about cloud storage auditing have been done in recent years, a critical security problem—the key exposure problem for cloud storage auditing, has remained unexplored. While existing protocols focus on the faults or dishonesty of the cloud, they have overlooked the possible weak sense of security and/or low security settings at the client.

- Auditing protocol did not consider this critical issue of how to deal with the client's secret key exposure for cloud storage auditing, and any exposure of the client's secret auditing key would make most of the existing auditing protocols unable to work correctly.
- The client still uses the traditional key revocation method. Once the client knows his secret key for cloud storage auditing is exposed, he will revoke this secret key and the corresponding public key. Meanwhile, he generates one new pair of secret key and public key, and publishes the new public key by the certificate update. The authenticators of the data previously stored in cloud, however, all need to be updated because the old secret key is no longer secure. Thus, the client needs to download all his previously stored data from the cloud, produce new authenticators for them using the new secret key, and then upload these new authenticators to the cloud. Obviously, it is a complex procedure.

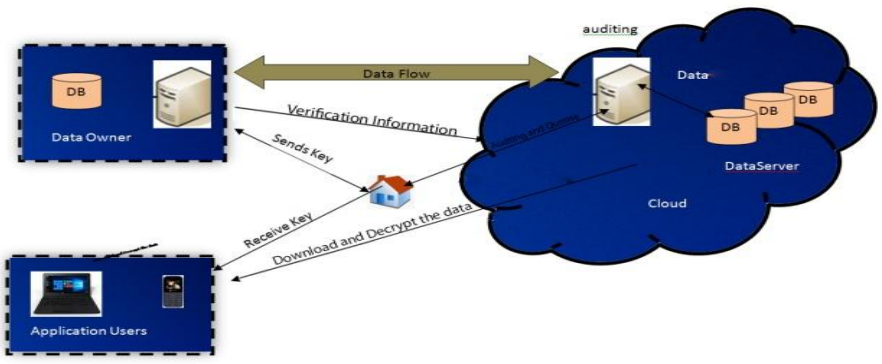
Proposed System:

In this paper we propose new key generation approach, in this key is generated automatically on upload and download and the key will be automatically updated to the data owner mail (TPA) And we implement blowfish algorithm for encryption and decryption As this complete paper narrates the different methodologies on enabling cloud storage auditing with key exposure resilience, but none of the methodologies seems to be perfect. So, in this paper as a bit proposes a method of an effective key exposure resistance where we adopt the de-duplication strategy of data. Moreover, it will check the duplicity of data and eliminate the redundant one using Whirlpool algorithm.

Advantages of Proposed System:

- We initiate the first study on how to achieve the key-exposure resilience in the storage auditing protocol and propose a new concept called auditing protocol with key-exposure resilience. In such a protocol, any dishonest behaviors, such as deleting or modifying some client's data stored in cloud in previous time periods, can all be detected, even if the cloud gets the client's current secret key for cloud storage auditing.
- This very important issue is not addressed before by previous auditing protocol designs. We further formalize the definition and the security model of auditing protocol with key-exposure resilience for secure cloud storage.
- There is a greater possibility of duplication of the data and due to this; the huge storage space is used unnecessarily. So, in order to enhance the storage space available in the cloud for better network efficiency and protect and maintain the integrity of data, the proposed system put forwards an idea of maintaining the same in a key exposure resilient system.
- In proposed system we introduced two algorithms, one is blowfish algorithm and other one is whirlpool algorithm.
- Blowfish algorithm is used for the encryption and decryption purpose and whirlpool algorithm for eliminating de-duplicated data on cloud storage. We also show that our proposed design supports the TPA.

System Architecture



Data Owner:

Data owner register in the cloud. After login into the cloud and upload a file which should be audit by the third party auditor. Before uploading the file data owner checks if there is any de-duplication is exists or not. If de-duplication is found then the file cannot be uploaded otherwise file will be uploaded in cloud. The file which is uploading by the owner is in encrypted format. The secret key will be send to third party auditor.

Data User:

Data user need to send a key request to third party auditor. After getting the key from third party, data user an access the file .The key is Time Stamp based Secret key. That key will be works for only one time access. If data user wants to access the file one more time again he needs to send a key request to third party auditor.

Third Party Auditor:

Third Party auditor accepts the data user key request and forward that request to Data owner. Data owner should accept the third party request and send a Time Stamp based secret key. Again the key send to the data user.

IV. ALGORITHM

A. Blowfish Algorithm:

Description of blowfish algorithm:

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

Subkeys

Blowfish uses a large number of subkeys. These keys must be precompiled before any data encryption or decryption.

The P-array consists of 18 32-bit subkeys:

P1, P2,..., P18.

There are four 32-bit S-boxes with 256 entries each:

S1, 0, S1, 1,..., S1,255;

S2, 0, S2, 1,..., S2,255;

S3, 0, S3, 1,..., S3,255;

S4, 0, S4, 1,..., S4,255.

Encryption

Blowfish has 16 rounds.

The input is a 64-bit data element, x.

Divide x into two 32-bit halves: xL, xR.

Then, for i = 1 to 16:

xL = xL XOR Pi

xR = F (xL) XOR xR

Swap xL and xR

After the sixteenth round, swap xL and xR again to undo the last swap.

Then, xR = xR XOR P17 and xL = xL XOR P18.

Finally, recombine xL and xR to get the cipher text.

Decryption is exactly the same as encryption, except that P1, P2…., P18 are used in the reverse order.

Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all subkeys are stored in cache.

Generating subkeys by using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed line. This string involves the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, and so forth
2. XOR P1 with the first 32 components of the key, XOR P2 with the second 32-bits of the main element, and so on for all portions of the main element (possibly up to P14). Repeatedly circuit through the main element bits until the complete P-array has recently been XORed with key portions. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, and many others, are equivalent keys.)
3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
4. Exchange P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.6.
6. Replace P3 and P4 with the outcome of step (5).
7. Continue the process, upgrading all entries of the P array, and then all four S-boxes in order, with the outcome of the consistently changing Blowfish algorithm.

In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times.

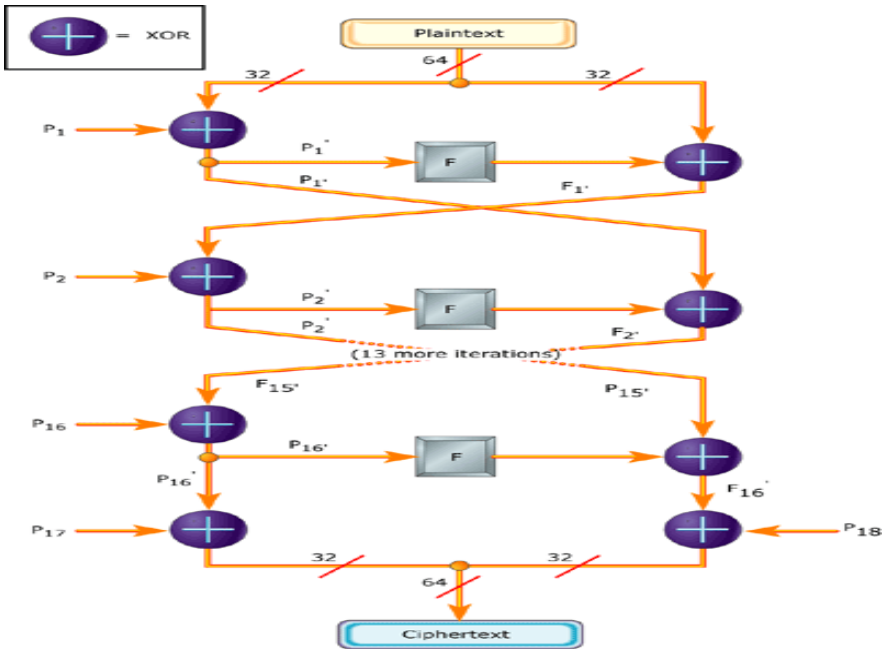


Fig7: Blowfish Algorithm

Whirlpool Algorithm:

The algorithm takes as input a message with a maximum length of less than 2256 bits and produces as output a 512-bit message digest. The input is processed in 512-bit blocks.

Consists of the following steps:

Step1: upload the file

Step2: Read the contents of the files from the cloud one by one.

Step3: Generate the hash code of the content by whirlpool and compare the hash code with the uploaded fileStep4: Read all the files from the cloud and generate the hash code.

Step5: compare the database file hash code with uploaded file hash code.

Step6: both are equal then duplicate if found, will not be uploaded both are not equal upload the file.

Consider

h	a	i

For encrypting the text we using “Blowfish algorithm”. Blowfish algorithm is a 64 bit block cipher. In this text “hai” it contains 3 bits remaining 61 bits are added. 64 bit is divided into two 32 bits.

Letter	Ascii	Hexa
h	104	68
a	97	61
i	105	69

Binary value

h	01101000
a	01100001
i	01101001

Then we have to perform XOR operation by using table;

0	0	0
0	1	1
1	0	1
1	1	0

h=0x68h, a=0x61h, i=0x69h

XOR P1 with the first 32 components of the key, XOR P2 with the second 32-bits.

Repeatedly circuit through the main element bits until the complete P-array has recently been XORed with key portions. The result value is a hexadecimal value.

Encrypted value for the given text is

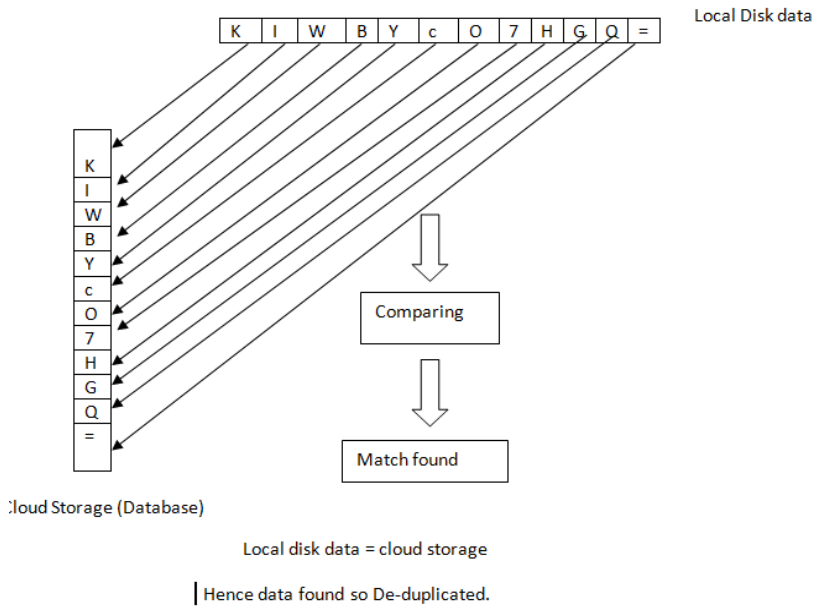


Fig8: De-Duplication is Found

V. CONCLUSION

We can conclude the blow fish algorithm is more secure to compare other symmetric key algorithms and produce best result for less processing time and rounds. To increase the key size of blowfish algorithm 128 to 448, it gives more privacy to the messages and provides high end data security when transmitting over any unsafe medium. By using whirlpool algorithm data de-duplication assures change in the fact that data is stored on disk at all tiers of storage (Primary, backup, Archival). However, it can only be beneficial if the De-dupe rate (or the disk space saving) justifies the extra expense of a data de-duplication application and the load it puts on the device resources. Alongside data de-duplication there are improvements in network and storage technology that provide the following: Extremely high data transfer rate, Reduce disk cost, Larger Disk IO rate. Data de-duplication that provides a higher de-dupe ratio at cost of lower throughput would detriment the thing benefit for advanced network and drive technologies mentioned above. Consequently the data de-duplication execution should effectively provide a high de-dupe ratio with a high throughput.

VI. FUTURE WORK

Future work, in cloud may be extended with more security features such as proofs of retrievability, data integrity checking out and search over encrypted data. In this paper we mainly concentrated on the how de-duplication is done in cloud storage space that is storage and retrieval. In future we plan to define other typical functions such as edit and delete. After implementing a prototype of the system, we aim to give a full performance analysis. Furthermore, we will work on finding possible optimizations in conditions of bandwidth, storage space and computation.

References

1. Enabling cloud storage auditing with key-exposure resistance authors by Jia yu, kui Ren, Senior Member IEEE, cong Wang, IEEE, Vijay vardharajan, Senior Member, IEEE.
2. G. Ateniese et al., “Provable data possession at untreated stores,” in Proc. 14th ACM Conf. Comput. Commun. Secur, 2007, pp. 598–609.
3. G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw, 2008, Art. ID 9.
4. K. Jin and E. Miller, "The effectiveness of de-duplication on virtual machine disk images" In Proc. SYSTOR 2009: The Israeli Experimental Systems Conference.
5. J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure de-duplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.Fröhlich, B. and Plate, J.

6. 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 and Communications Security, pages 491–500. ACM, 2011.
7. Atul Adya, William J Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R Douceur, Jon Howell, Jacob R Lorch, Marvin Theimer, and Roger P Wattenhofer. Farsite: Federated, available, and reliable storage for an incompletely trusted environment. ACM SIGOPS Operating Systems Review, 36(SI):1–14, 2002.
8. Mihir Bellare, Alexandra Boldyreva, and Adam ONeill. Deterministic and efficiently searchable encryption. In Advances in Cryptology-CRYPTO 2007, pages 535–552. Springer, 2007.
9. Ari Juels and Burton S. Kaliski, Jr. Pors: proofs of retrievability for large files. In Proceedings of the 14th ACM conference on Computer and communications security, CCS '07, pages 584–597, New York, NY, USA, 2007. ACM.
10. 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 Proc. 3rd Int'l Workshop Security Cloud Comput., 2011, pp. 160-167.

AUTHOR(S) PROFILE

Yandrapu Gayatri, is currently pursuing M.Tech in Department of Computer Science and Engineering at Pydah College of Engineering and Technology, affiliated to JNTUK University, AP, India. Her area of interest includes Cloud Computing.

Mr. A.V.D.N.Murthy, completed his MCA and M.Tech in Computer Science and Engineering. He is currently working as Assistant Professor of Department of Computer Science and Engineering at Pydah College of Engineering and Technology, affiliated to JNTUK University. He is having industrial experience of 1.2 years and teaching experience of 10 years. His areas of interest include Data mining, Image Processing, Cryptography & Network security, Cloud Computing, Computer Networks and Operating Systems.

Dr. Ramesh Challgundla, M.E,PH.D, MIEEE(USA), FIETE(IND), MIE(IND), MISTE is former faculty of Birla Institute of Technology, GITAM University, ANITS and current working as Professor and Principal of Pydah College of Engineering and Technology, Visakhapatnam.