Encryption and Deduplication of Data In The Web

N . Archana Devi[1], M . Jayasree[2], C. Mohana Priya[3], Dr.P.Mathiyalagan[4]
[1, 2, 3, 4] Department of Computer Science and Engineering
[1, 2, 3, 4]Sri Ramakrishna Engineering College ,Coimbatore

Abstract-*Data De-duplication is one of important data techniques for eliminating duplicate copies of repeated data by comparing the data in server storage to reduce the amount of storage space. By proposing this advanced duplication system supporting authorised duplicate check and compares the storage system with file content. In this new duplication system, The keys will not be issued by users directly, which will be kept and managed by the private server instead. In this way, the users cannot upload the same hash value data because it compares the whole data base storage system, which means that it can prevent the duplication process with same content. To get a file value, the user needs to send a request to the private server. To perform the duplicate check for some file, the user needs to get the file content from the server. The authorised duplicate check for this file content can be performed by the nave Bayes classifier in the server storage before uploading this file. Based on the results of duplicate check, the user uploads this file.*

I. INTRODUCTION

File systems often contain redundant copies of information that are identical , possibly stored on the single host,on a shared storage cluster or backup to the secondary storage. This fine de-duplication creates more opportunities for space savings ,but reduces the sequential layout of some files, which may have significant performance impact for storage when hard disks are used. Alternatively whole file de-duplication is simpler and eliminates file-fragmentation .Because the disk technology trend is towards improved sequential bandwidth and reduced per-byte cost, with little or no improvement in random access speed, it's not clear that trading away sequentiality for space savings makes sense, at least in primary storage. Eliminating redundant data can significantly shrink storage requirements and have efficient bandwidth. Because primary storage is usual, it typically store many versions of the same information so that new workers can reuse previously done work.operations like backup,that store extremely redundant information.De-duplication lowers storage costs as fewer disks that are needed for backup.Backup and archive data usually includes a lot of duplicate data.The same data is stored over and over again, consuming unnecessary storage space on disk.This creates a chain of cost and resource inefficiencies within the organisation.

II.RELATED WORK

1) On the implementation Of Pairing-Based Crypto systems

In this paper we discuss pairings and algorithms to find pairing-friendly curves. We will not discuss curve-finding and point-counting algorithms geared towards standard elliptic curve cryptography which necessarily requires curves that are not pairing-friendly, but instead direct the interested reader to Blake and Smart.There is only one known mathematical setting where desirable pairings exist, that is hyper elliptic curves. Because, this elliptic curves, which are the simplest case, and also the only curves that are still used in practice. All existing implementations of pairing-based crypto systems are built with elliptic curves. Accordingly, we provide a brief overview of elliptic curves, and functions known as the Tate and Weil pairings from which the cryptographic pairings are derived. We describe several methods for obtaining curves that yield Tate and Weil pairings that are efficiently computable yet that are still cryptographically securable.We discuss many optimisations that greatly reduce the running time of a implementation of any pairing-based crypto system. These techniques are used to reduce the cost of the pairing from several minutes to several milliseconds on a modern consumer-level machine.

2) Reclaiming Space from Duplicate Files in a Server less Distributed File System

This paper addresses the problems of identifying and coalescing identical files that are stored in the Far site distributed file system, for the purpose of reclaiming storage, that spaces are consumed by incidentally redundant content. Far site is a secure, scalable, server less file system that functions as a centralised file server but that are distributed among a network based collection of desktop workstations.

Far site is a distributed file system that provides security and feasibility by storing encrypted replicas of each file on multiple desktop machines. To free space for storing these replicas, the system coalesces incidentally duplicated files, such as shared documents among workgroups or multiple users' copies of common application programs.

3) Understanding data deduplication ratios

A data deduplication ratio over a particular time period is the number of bytes input to a data deduplication process divided by the number of bytes output. The length of time that data is retained, impacts data deduplication ratios in two ways. First, if more data is examined when deduplicating new data, the likelihood of finding duplicate data is understanding data deduplication ratios 9 of 13,storage networking industry association Increased and the space savings may increase. Secondly, if a data deduplication ratio is calculated over longer periods of time it may increase because the numerator tends to increase more rapidly than the denominator.

Data deduplication lowers business risks, increases revenue opportunities, and reduces storage tier costs, resulting in a perfect storm for companies deploying an adaptive storage infrastructure. Storage resiliency technologies, such as RAID or RAIN, safeguard the deduplicated data to ensure high availability of applications accessing the data.

4) Proofs of Ownership in Remote Storage Systems

In this paper we identify attacks that have exploited client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on a very small hash signature of these files. More specifically, that an attacker who knows the hash signature of a several file can convince the storage service that it owns that file; hence the server lets the attacker download the entire file.

we introduce the part that explains the proofs-ofownership (PoWs), which lets a client efficiently prove to a server that that the client holds a file, rather than just some short information about it.

III. PROPOSED SYSTEM

The use of the efficient encryption is proposed, i.e., the keys are derived from the plaintext and stores al1 of it. Some security problems are considered, and producing a secure de-duplication of data. But, these protocols focus on server storage side de-duplication and they do not involve in data leakage settings, against harmful users.

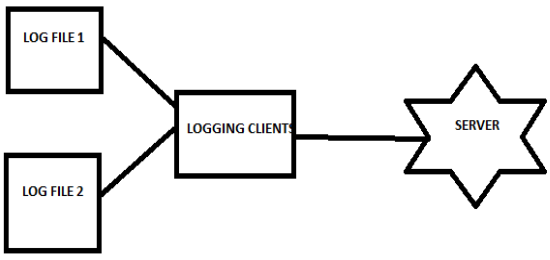
In this paper, proposing a considerate solution for storing data and maintaining log records in a server that operates in a cloud based environment. Addressing the security and integrity issues not just during the log generation phase, but also during other stages in the log records management process,even includes log collection, transmission, storage, and retrieval of data. The major contributions of this paper includes proposing architecture for the various components of the system and developing cryptographic protocols to reduce integrity and confidentiality issues with storing, maintaining, and querying log records at the storage provider and in transit.

Log Generators

These computing devices generate log data. Each organisation has its cloud-based log management and has a number of log generators. Each those generators work up with certain logging capability. The log files are generated by these hosts they are stored temporarily till the time they are pushed to the logging client.

Logging Client or Logging Relay

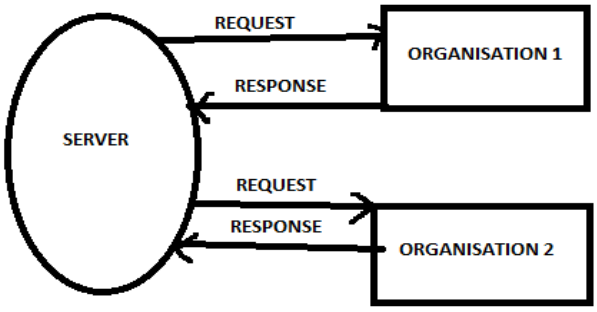
The logging client is a sort of collector receiving groups of log records generated by the log generators, and prepares the log data with it ,that it can be pushed to the server for long term storage. The log data gets transferred from generators to the client batch wise, either on a schedule, or as a need depending on the amount of log data waiting to be transferred. The logging client ensures security protection on batches of accumulated log data and pushes each batch to the logging server. When the logging client pushes log data to the server it acts as a logging relay. The terms logging client and logging relay are used interchangeably. The logging client or relay can be implemented as a group of host collaborations. Assuming as a single logging client.



Logging server

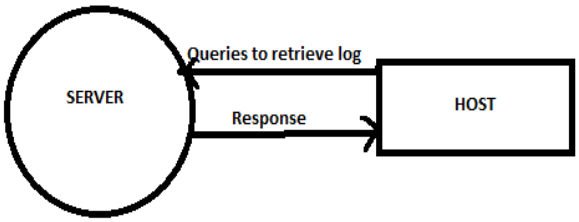
The logging server provides long term storage and maintenance to log data that are received from different logging clients of different organisations. The logging server is maintained by a storage service provider.Those organisations that have subscribed to the logging server services can upload data to the server. The server, on request from an organisation can also delete log data and perform log

rotation. Before the logging server delete or rotate log data it needs a proof from the requester, that the latter is authorised to make such an request. The logging client generates such proof. Hence,the proof can be given by the logging client to any entity that it wants to authorise.



Log Monitor

The log monitor are the hosts that are used to monitor and review log data. They can retrieve log data from the server by generating queries. Based on the log data retrieved, monitors will perform further analysis as needed. They can even consult the log server to delete log data permanently, or rotate logs.



IV. OVERVIEW

The implementation of A Secure Client Side De-duplication Scheme in server Storage Environments with the creation of JAR files and verification of JAR files.

Overall Description:

To propose a comprehensive solution for storing and maintaining log records in a server that operates in a cloud-based environment.

Product Perspective:

Data can be uploaded into the server and providing security for our data which is available in storage and also creating log record for the data owner.

Product function:

The main purpose of this project is to provide data security and maintenance in storage.To create log record generation for the data owners.

Specific Requirement:

External Interface Requirement:

The user friendly simple and easy interface to understand and use with high interaction.The system prompt for the user to login to the application and for proper Input criteria.

User Interface:

The software provides good graphical interface for the user to perform the required task such as uploading and downloading the data from the storage.

Input & Output Details:

a. Input Specification: user request for storage services

b. Output Specification: Log generating, Storing and Monitoring with encryption /decryption schemes.

V. ALGORITHM DESCRIPTION

I.Advanced Encryption standard(AES)

KeyExpansion—Round keys are derived from the cipher key by using Rjindael's key scheduling.

Initial Round

AddRoundKey—Each byte of this state is combined with the round key using bitwise xor operation.

Rounds

SubBytes—a non-linear substitution step where each byte is replaced with the another according to a lookup table.

ShiftRows—a transposition steps, where each row of the state is shifted cyclically on certain number of steps.

MixColumns—a mixing operation which operates on the columns of certain state, combining the four bytes in each column.

Add-RoundKey

Final Round (no MixColumns)

Sub Bytes

Shift Rows

Add RoundKey

II. MD5 algorithm

Step1- Append padding bits

The input message is "padded" so that its length (in bits) equals to 448 mod 512. Padding is always performed, even if the length of certain message, that is already 448 mod 512.

Padding is performed as , a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent and equal to 448 mod 512. At least one bit and at most 512 bits are being appended.

Step2. Append length

A 64-bit representation of the message is appended to the result of step1. If the length of the message is greater than 2^64, only the low-order 64 bits will be used for this operation.

The resulting message has a length that is an exact multiple of 512 bits. The input message will have a length that is always exact multiple of 16 (32-bit) words.

Step3. Initialise MD buffer

A four-word buffer (A, B, C, D) is used to compute the message digest(MD5). Each of the message A, B, C, D is 32-bit register. These registers are initialised to the following values in hexadecimal.

Step4. Process the message in 16-word blocks

Four functions will be defined such that each of the function takes an input of three 32-bit words and produces a 32-bit word output.

VI. CONCLUSION

The proposed system is used for secure uploading of data into the server.This can be done with efficient Encryption and decryption Algorithm.The files can be downloaded securely by authorised users.

REFERENCES

[1] Jingwei Li, Jin Li, Dongqing Xie and Zhang Cai "secure auditing and deduplicating data in cloud", IEEE Transactions on Computers Volume: PP , Issue: 99, 26 January 2015.

[2] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in IEEE Conference on Communications and Network Security (CNS), 2013, pp. 145–153.

[3] "Message-locked encryption and secure deduplication," in EUROCRYPT, 2013, pp. 296–312.

[4] H. Wang, ―Proxy provable data possession in public clouds,‖ IEEE Transactions on services Computing, vol. 6, no. 4, pp. 551–559, 2013.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communication of the ACM, vol. 53, no. 4, pp. 50–58, 2010.