International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Methodology of Securely Backing up Cloud Information by MFA and De-duplicating Encrypted Information

Naga Chandrika H.[1], **Dr. Pramod Pandurang Jadhav**[2]

Submitted: 09/01/2024 **Revised**: 15/02/2024 **Accepted**: 23/02/2024

Abstract: In the context of mobile devices or laptops, the potential loss, theft, misplacement, or corruption of sensitive information is a concern. To mitigate these risks, a technique involving remote backup memories is employed; nonetheless, complete trustworthiness cannot be attributed to the backup server, necessitating the prior encryption of the information. While the key benefits from enhanced protection owing to its reduced size in comparison to the backup information, its memorization remains an insurmountable task for the average individual. In this endeavour, the problem of formulating a secure deduplication scheme based on portions is explored within the context of enterprise backup memories. A user-centric design for an information backup scheme is put forth, incorporating multifactor authentication. Initially, an encipherment is chosen by the user, divided into three shares, and subsequently rendered irretrievable. Specifically, it is discerned that the susceptibility of the information within a diminutive portion to brute-force attacks surpasses that of the file-based deduplicating procedure. A randomized oblivious encipherment generating mechanism is suggested, rooted in the internal operations of the memories service. The reconstruction of the encipherment is effortlessly achievable through the amalgamation of shares preserved insideinto utilizer's smartcards and desktop. Stipulated scheme not only attains the necessary security objectives but also demonstrates heightened resilience and practicality.

Keywords: *Cloud Information, Securedly Backing, Deduplicating Encrypted Information, Trustworthiness, Randomized.*

1. Introduction

In recent times, there has been a swift evolution in distant memories preservation, aiming to provide society with a service that is adaptable, economical, and user-friendly [1]. The adoption of the deduplicating procedure, known as dedupe, is on the rise among cloud memories services, exemplified by entities like Amazon S3, to effectively curtail the expenses associated with enciphered information storage. Nevertheless, complete trust in the distant servers is withheld, as evidenced by reported incidents of information leakage, which adequately substantiate this assertion [2]. Upon detecting redundancy in the system, the deduplicating process retains solely a single instance of identical information and establishes a reference directing to the archived duplicate for other replicates. To ensure confidentiality, the user must perform encryption on the information before initiating the upload. Even in the event of disclosure of the code, the assailant remains incapable of restoring the plaintext without access to the decryption key. The achievement of information confidentiality is attained through the utilization of deterministic encipherment, such as convergent encrypting processes (CE) [3], wherein the encipherment is derived from the information itself, and identical plaintext consistently results in the same encipherment keys and ciphertexts. An instance of frequent usage in the encrypting process is the symmetric encipherment AES [4], employed, for example, to encipher information. As a consequence, the application of the deduplicating process to the enciphered content can be executed without divulging the concealed stored information. Nonetheless, the prevailing secure designs for deduplication [3], [5], [6], [7], [8], [9] exhibit, to a certain degree, incongruities with the real-world requirements of the deduplicating process in terms of both security and operational efficiency. Nevertheless, the encipherment remains challenging for ordinary individuals to commit to memory and is typically stored on a USB device [10] or a laptop. In the event of theft or tampering with the storage device, the user is compelled to undergo a re-registration process, characterized by complexity. Meanwhile, there exists a potential risk of information exposure.

[1,2]*Department of Computer Science and Engineering*
[1] *Research Scholar, Dr. A. P. J. Abdul Kalam University, Indore*
[2] *Research Supervisor, Dr. A. P. J. Abdul Kalam University, Indore*
E-mail Id: [1]nagachandrika87@gmail.com, [2]ppjadhav21@gmail.com
** Corresponding Author: Naga Chandrika H*
Email: nagachandrika87@gmail.com

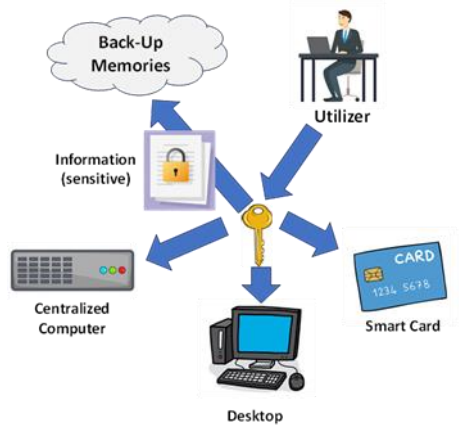

Fig 1. Potential risk of information exposure.

Identified Research Gaps: Through the utilization of diverse portioning techniques, the deduplicating process can be executed either at the comprehensive file level or the detailed portion level. In 2015, a information securing scheme centred on the server (SCD) was brought into being by Chang et al. [11]. In this scheme, the encipherment is chosen and divided in 3 parts using sharing of secret by Shamir [12] by server. Subsequently, these parts were individually saved inside USB devices, a desktop, and the centralized-computer itself. Proposals by Chang et al. appears to be operational because aspects of information-based safety inherent in sharing of secret by Shamir [13, 14]. In summary, the process of deduplicating based on files (FBD) proves suitable for small or static file formats, such as pdb, lib, dll, etc. Even minor alterations introduced to the file yield an entirely distinct copy, consequently resulting in diminished efficiency of the deduplicating process. A set of traits that characterize a trustworthy and efficient identity authentication scheme includes: (i) the provision of substantial security to endure various predictable intrusions, (ii) allowing utilizer option of choosing with updating of passwords, (iii) the accomplishment of simultaneous authorizations with the utilizer along with centralized computers, (iv) the generation of protected sessions-based encipherment towards following communications, and (v) safeguarding of utilizer's confidentiality. Minor alterations within the file result in an entirely distinct copy, leading to diminished efficiency in the deduplicating process. On the contrary, portion-based deduplication (PBD) demonstrates the capability to partition a given information streams to tinier portions of variable or fixed sizes, generally ranging from 16 Kilobytes to 4KiloBytes. Hence, a significant reduction in memory usage can be achieved through the elimination of repetitiveness at the portion level. In actuality, the primary consumer of memories is increasingly constituted by sizable files [15]. An increasing number of practical memory systems are adopting portion-based redundancy elimination (PBD) as their fundamental technique for eliminating repetitiveness [16]. Fundamentally, extensive research has been conducted on various authentication methods to fulfil the aforementioned criteria [17-19], generally classified into either traditional methods or biometric-based multi-factor authentication techniques. The assertion that the latter represents a pragmatic approach to ensuring authentication in security frameworks is challenging to evade [20-26].

2. Related Works

The initial authorization method, introduced by Lamport, employed a username and password combination alongside identifying process – a one-factor rule vulnerable to attack due to utilizer's tendency to choose short, non-random passwords susceptible to both dictionary attacks and casual observation. As Wang et al. [19] later noted, the prevalence of password reuse and note-taking further compromised security, rendering password-based authorization alone insufficient. Vulnerability to compromised information spurred the development of a reinforced validation protocol – two-factor authentication. This method leverages a two-pronged approach, where one prong hinges on knowledge, in the form of memorized encipherments known as passwords, while the other relies on a tangible possession, such as a smart card or token. As a result, even if one prong's information were exposed, the second remains a formidable barrier, effectively bolstering the integrity of the validation process.

2.1 Information deduplicating procedure

Just like an information compression engine identifies redundancies within a single file, a deduplicating process seeks to eliminate duplicated information, both within and among files. Generally, a "portion-based" deduplicating approach can capture finer-grained redundancies within individual files, often leading to a higher rate of duplicating procedure removal. The paper [27] recommended the implementation of novel hybrid data portioning techniques to enhance the effectiveness of the deduplicating procedure within digital repositories. A content-aware hybrid data portioning procedure known as Two Threshold Two Divisor (TTTD) was proposed in conjunction with a Dynamic Prime Coding (DPC) procedure to minimize the data volume requiring the deduplicating procedure. A groundbreaking cloud auditing scheme [28] emerged, facilitating secure information redundancy elimination across users and simplifying ownership verification. This scheme leverages a delayed modification procedure to minimize modification frequency and delegate modification tasks to the cloud, significantly alleviating the computational overhead of verifying information ownership within digital repositories. A groundbreaking system [29] proposed by Song et al., dubbed LSDedup, was meticulously crafted to optimize digital repositories by minimizing data

redundancy while simultaneously safeguarding highly sensitive information. This innovative approach empowers remote users to engage securely with cloud servers, enabling them to verify the confidentiality level of their entrusted files. To expedite information deduplicating procedure, a specialized procedure [30] proposed by Begum et al. leveraging Merkle Hash Trees within a MapReduce framework was executed. This innovative approach enhanced the efficiency of entity indexing, leading to a notable performance boost in both storage and retrieval operations within digital repositories. Furthermore, a data encryption scheme was employed, utilizing the Niederreiter Public Key Cryptosystem based on the assigned level of sensitivities (high/low) of each entity, thereby ensuring robust protection for sensitive information. A groundbreaking framework [31] proposed by Qi et al. dubbed AC-Dedup emerged, granting secure and dynamic access control within mobile cloud memories. This revolutionary approach harnesses two pioneering encrypting procedures, known as mixed message locked encryption and random stub re-encryption, to effectively repel diverse security threats. Each procedure stands guard against specific attack types, meticulously safeguarding the integrity and confidentiality of information entrusted to the cloud. A groundbreaking cloud auditing framework [32] by Qi et al. emerges, enabling streamlined verification of data integrity and optimized redundancy elimination within digital repositories. This framework leverages a revolutionary signature conversion mechanism, empowering seamless transformation of verification tokens from an acquired entity to those of the acquiring entity, facilitated by the cloud's assistance. This innovative approach ensures efficient data transfer while upholding robust data security safeguards. An initial mathematical approach [33] modelled the information migrating procedure problem as an integer linearized programming (ILP) puzzle, tackled by a commercial solver when facing a single empty target centralized computers. This framework then underwent expansion to address situations with multiple pre-occupied target centralized computers, where efficient heuristic procedures based on Bloom Filter-derived data summaries emerged as the preferred solution strategy.

3. Proposed Schemes

In the safe buyer-side across-utilizer deduplicating procedure based system, three entities are involved: the Encipherment Centralized Computer (KS), the Buyers (C's), and the Public Cloud Memories Centralized Computer (SS), as illustrated in Figure 1. The system is designed to provide a periodic file backup service to clients within a business-based networks. The Encipherment Centralized Computers (KS) assumes the responsibility of client authentication and the generation of encryption keys for data portions. In this process, a client (Cj) employs a portioning algorithm on their backup data. When Cj requests authentication, the Key Server (KS) validates the client and, in an oblivious manner, generates a unique Portion Encryption (CE) key, denoted as 'k,' for every portion ('ch') within information of backups named as Cj's. Subsequently, Cj encrypts their data portions using the corresponding CE keys. The resulting ciphertexts are then uploaded to the Public Cloud Storage Server (SS2), such as Microsoft Azure Backup. The Public Cloud Storage Server (SS) is responsible for storing the deduplicated incoming data stream within the appropriate containers before committing them to persistent storage. It is crucial to note that the entire data protection phase, encompassing key generation and data encryption, remains opaque to the Public Cloud Storage Server (SS). The server merely provides a basic and straightforward interface to its clients, akin to the scenario of plaintext data backup.

4. Model of Securing the Cloud Back-Up Infrastructure

The primary emphasis in this research lies in safeguarding the confidentiality of foreseeable data. This focus stems from the recognition that achieving semantic security for unforeseeable data is attainable through Portion Encryption (PE). Throughout the protocol execution, the Key Server (KS) remains oblivious to any details regarding the input portion of the client, denoted as Cj. It is imperative to note that, in the event of a compromised Public Cloud Storage Server (SS), a potential risk arises in the form of an offline brute-force attack.

This attack vector involves the compromised SS launching an offline brute-force attack by systematically enumerating ciphertexts associated with predictable file candidates. The attacker subsequently compares these ciphertexts with the target ciphertext in an offline capacity. Despite the usual implementation of stringent security policies within enterprise networks, the research framework acknowledges the plausible scenario of an external adversary compromising a limited subset of internal clients.

In light of this, the adversary gains the capability to execute an online brute-force attack by further infiltrating the Key Server (KS). This implies that, even within the confines of a well-protected enterprise network, the assumption is made that a restricted number of internal clients could be compromised, providing a potential avenue for adversarial activities.

Definition of ideal functionality F_{dedupe} for the proposed schemes of prevention of duplications. The inputs of the function $dedupe$may be defined as

Inputs to F_{dedupe} areas follows:

• *C*lients *Cj* has a input portions *ch*;

• The keys servers *KS*'s inputs is a chosen secret *dt*;

• The cloud storages servers *SS* has no inputs.

Outputs of F_{dedupe} are:

- *Cj* obtains the portion key *k*;

• Outputs of *KS* is $\theta\_$;

• *SS* get the ciphertext $c = Enc(k, ch)$

The validation of the security of the proposed scheme within the malicious model is contingent upon the incapacity of a probabilistic polynomial-time (PPT) adversary to discern between the real-world execution of the devised protocol and an ideal-world protocol that embodies the functionality denoted as F_{dedupe}. This evaluation is conducted under the supervision of a PPT simulator. It is noteworthy that, within the scope of this study, considerations exclude side-channel attacks, proof of ownership, and key management.

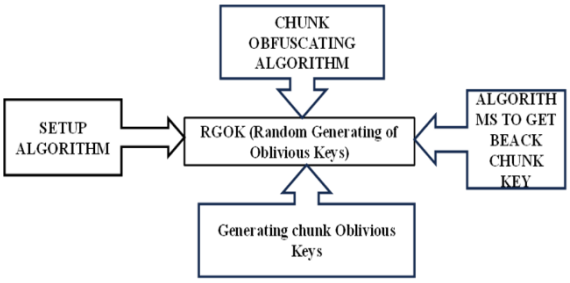

Fig 2. Random Generating of Oblivious Keys

The architectural blueprint of our system is intended to complement ongoing research endeavors. As part of the foundational assumptions, it is posited that all communication channels linking the Key Server (KS), denoted as C_j, and the Public Cloud Storage Server (SS) are secure. These channels are deemed impervious to eavesdropping or tampering attempts by potential adversaries.

4.1 Steps of the Random Generating of Oblivious Keys Algorithm

Step 1: Key Server is Setup

Step 2: Initialization() sub-algorithm is started.

Step 3: Clinet Keys (PK_i) and Master Secret Keys (MK_i) are generated

Step 4: Client Keys (PK_i) are distributed to the client.

Step 5: Master Secret Keys (MK_i) are kept hidden in the Key Server for future authentication purposes.

Step 6: PortionObfuscation() algorithm is started

Step 7: Obfuscation of chunks of data is done by a randomly generated number generation.

Step 8: In PortionObfuscation() algorithm hash function is also used along with the generated public keys PK_i

Step 9: ObliviousKeyGen() function is initiated

Step 10: KeyRecovery() algorithms is initiated for generating the keys for deciphering the data.

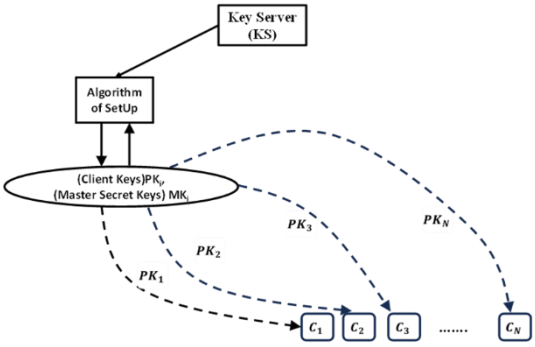

Fig 3. Flowchart of Key Distribution in Deduplication mechanism of Backup Data.

4.2 Aims of this Design

The team has formulated a privacy-centric deduplication system at the chunk level, with a focus on attaining specific design objectives. In the domain of security, the goals encompass:

1) The realization of the ideal functionality F_{dedupe} within the models that are hazardous;

2) The prevention of off-line extreme-force attacks orchestrated by the Public Cloud Storage Server (SS); and

3) The mitigation of online brute-force attacks by impeding their pace while concurrently ensuring resilience to compromise from multiple clients. Turning attention to the performance dimension, the objectives are:

4) The achievement of efficient generation of chunk encryption keys; and

5) The establishment of a design that holds parity with plaintext Content-Based Deduplication (DCB) in terms of performance metrics, including deduplication ratio and data restoration speed.

4.3 Design of the Protocol

The process of generating a chunk encryption key involves the execution of a secure (oblivious) two-party computation protocol between the client, denoted as Cj, and the Key Server (KS). This design ensures that KS remains unaware of both the input and algorithmic output of Cj, while concurrently preventing Cj from deducing KS's confidential information. In a broader context, the

desired protocol can be instantiated through the implementation of any blind signature scheme. In this context, a widely-adopted approach involves leveraging a blind RSA signature, akin to methodologies explored in preceding studies [4]. An additional layer of complexity is introduced into the oblivious key generation by incorporating randomness into the process.

Algorithm Definition: Consider hash functions G mapping from the set of integers modulo n to binary strings of length l, and H mapping from binary strings to integers modulo n. The algorithm denoted as ROKG is formally defined in the following manner.

The devised a generating procedure of randomized obliviously encipherment (ROKG) algorithm for a collective of s clients within the framework comprises 4 foundational procedures.

Initialization(λ_n, λ_r) → ({MK, PK}): The initialization procedure receives safety variables λ_n and λ_r as input and generates n distinct groups of RSA variables denoted as $\{(N_i, e_i, d_i)|1 \leq i \leq n\}$. Consequently, the public variables encompass PK = $\{(N_i, e_i)\}$, and the secrets of master-type form $\{d_i\}$=MK.

ChunkObfuscation(ch, r, PK_i, H) → z: The algorithm for obfuscating chunks accepts the chunk data, denoted as *ch*, along with a randomly generated number, r. It also takes the associated public key $PK_i = \{N_i, e_i\}$ and utilizes the hash function H. The output of this process is the obfuscated chunk data, represented as z.

ObliviousKeyGen(MK_i, z) → θ': The algorithm for generating oblivious chunk keys is fed with the client's secrets of master-type, denoted as $d_i = MK_i$and portions that have gone through obfuscation, represented as z. As a result, it produces the corresponding obfuscated chunk key, denoted as θ'.

KeyRecovery(θ', H, G, PKi, r) → k or ⊥: The algorithm responsible for recovering chunk keys is provided with the input parameters θ', functions of hashing like H and G, pertinent public variables PK_i, and the randomly generated numeral r. Upon successful verification of θ', the algorithm results of portion-based encipherment that has gone through the encrypting process, denoted as k. Conversely, if the verification fails, the output is represented as ⊥.

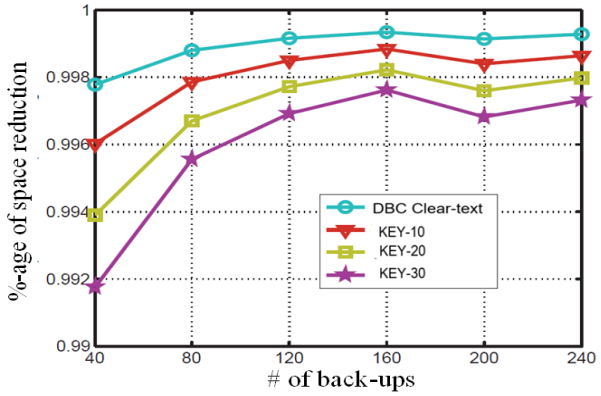

Fig 4. The space savings, denoted as sr, exhibit fluctuations with the expanding count of back-up memories instances inside clouds. Every back-up encompasses pictures derived from the machines of 54 distinct users.

4.4 Increasing the pace of information Restoration:

Multiple factors underscore the significance of prioritizing read performance, even within backup storage systems. Firstly, the expeditious restoration of data holds paramount importance in scenarios involving crash or corruption recovery. Enhanced read speeds directly translate to shorter intervals for recuperation, thus reducing the overall recovery window. Additionally, there exists a necessity to reconstruct the original data stream, a task that occurs more frequently than user-initiated data retrieval processes. This reconstruction facilitates staging backup data streams toward archival storage, a crucial step considering the finite capacity of deduplication storage solutions [26].

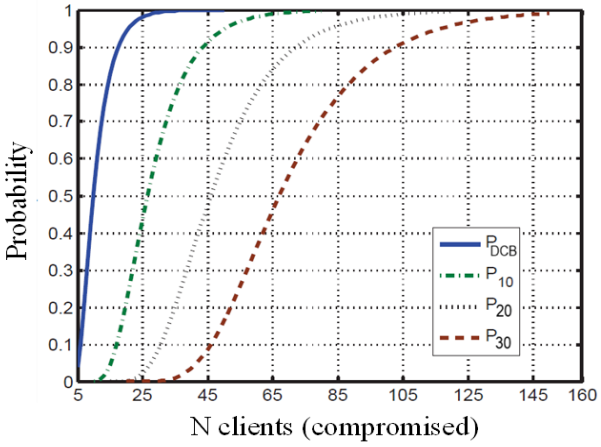

Fig 5. Pn(m) with m clients that are compromised at the time n = 10,20,30 and DBC-Cleartext

4.5 Evaluation of Results:

The practical implementation of the safe portion-based systems that does deduplicating proceures is executed in authentic back-up memory infrastructure sourced from the Files Frameworks and Memories Laboratories at University of Stony Brook. The primary aim is directed toward the Snapshots MacOS 2013 datasets, compiled

from a Mac OS X Snow Leopard centralized computer accommodating fifty four utilizers. The dataset comprises a net of 249 snapshots, constituting daily backups over an 11-month duration. The chunking methodology employed is multiple-valued, and a mean portion value of eight KiloBytes.

To emulate an business back-up scenario, the synthesis involves the creation of per person utilizer back-ups through extraction of entities from respective snapshots. Inclusion of data associated with UID-0 is integral, representing framework of entities of basic type. The cumulative value of the considered back-up memories, prior to deduplication processes, approximates 463 TB. Additionally, parameters such as magnitude of portion containers and the Least Recently Used (LRU) strategy are configured to simulate the enterprise backup environment.

Table 1. Abbreviations

Short Forms	Explanations
I_{SC}	Smart Card's ID
I_{UR}	Utilizer's ID
I_{SV}	Server's ID
P	Passwords of Utilizer
m_j	Randomized integers (OTP)
KS	Key of Session
$D_K/E_K()$	KS based Decryption/Encryption
$i(\cdot)$	Function of Hashing
$\oplus$	Operations of EX-OR
$\|\|$	Operation of Concatenating

Prior secure chunk-based designs, while focusing on different research aspects, omit the consideration of safeguarding low-entropy data when faced with a formidable adversary. Additionally, the evaluation of practical deduplication performance, including factors like fragmentation level, remains unexplored. Their anticipated performance is expected to align closely with plaintext Content-Based Deduplication (DCB) concerning deduplication ratios and data restoration speeds. This alignment arises from their heuristic approach, preserving unique chunk copies within the system. Subsequently, these existing designs will not be explicitly referenced moving forward. The forthcoming discussion will exclusively involve a comparative analysis between the proposed scheme and plaintext DCB.

Table 2. Price of Computing

Entity	Phase of Registering	Phase of Authenticating			Phase of Recoveries	Phase of Updating			
		1st Cases	2nd Cases	3rd Cases		Smart Cards	Desktops	Passwords	Bio-data
Smart-cards	-	t_h	-	t_h	-	-	-	t_h	t_h
Utilizer	$t_p + 3t_h$	-	$4t_h + t_s$	$4t_h + t_s$	t_p	$t_p + 2 + t_s$	$t_p + t_s$	$3t_h$	$2t_h$
Servers	-	-	$5t_h + t_s$	$5t_h + t_s$	-	t_s	t_s	-	-

5. Methodology Used

The implementation of the generating of the rules of encipherments that are oblivious on the basis of TCP and are randomized connecting the centralized computer based encipherments (KS) and the clients (Cj) has been carried out using Python. The server machine boasts a processing power of a three point one GigaHertz AMD FX 8120 processing devices and is equipped with thirty two GigaByte DDR3 memories. Simultaneously, client machine on same Local Area Network (LAN) is furnished with an Intel i3/2120 processing device and twelve GigaByte of memories. For the unseen RSA signatures aspect, the implementation employs RSA1024 in conjunction with SHA256. Additionally, the Chunk Encryption (CE) functionality is instantiated through the use of CTR mode with the Advanced Encryption Standard (AES) algorithm operating on a 128-bit key length.

1) Measurement of Latency: The evaluation of latency involves quantifying the time taken by the online protocol proposed herein. Latency, in this context, is defined as the duration spanning from the moment Cj dispatches applications for when it get replies from K S. Illustrated in Figure 5a, the initial procedures, lacking optimizations in chunk key generation, exhibits a linearized increase in overhead proportionate to magnitude of information of clients. Larger back-up information streams generate more portions, thereby triggering a higher frequency of per-portion encipherment generating method cases.

Table 3. Time to Simulate

Entities	t_h	t_s	t_p
3 GigaHertz Dektop	0.00176ms	0.00326ms	0.003ms
155 MegaHertz HyperSmart	Less than 0.0005	-	-

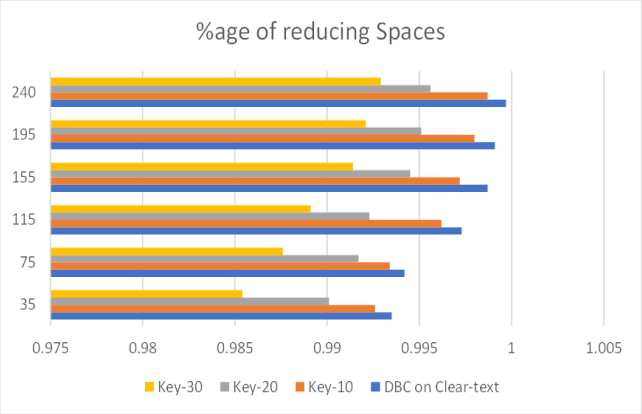

Fig 6. Comparison of performances of DCb-Clear-text with varying Key Sizes

Figure 6(a) further reveals that the protocol's latency experiences a notable reduction and achieves a state of constancy when leveraging content-aware chunking. This optimization entails executing the protocol solely for the generation of format encipherment, kf. Consequently, latencies becomes contingent solely over count of separate entity formats within the client's back-up streams, assuming all files are predictable. The magnitude of the back-up streams no longer plays a determining role in the latency, highlighting a significant reduction in the protocol's performance variability.

2) Evaluation of Client-side Costs: The assessment of additional costs incurred on the client side encompasses both the Randomized Oblivious Key Generation (ROKG) protocol and convergent encryption. As depicted in Figure 5(b), the overhead is predominantly contingent upon the size of the backup data stream. It is noteworthy that, despite employing the proposed efficient key generation algorithm, this cost exhibits linearity concerning the total backup data size. This characteristic persists because the client must still execute off-line hash work done for every portion encipherment and subsequently encrypt them one by one.

Once more, the findings underscore a substantial working advantages achieved through the implementation of the procedure for generating of encipherments efficiently.

5.1 Efficiency of Deduplication

A simulating mechanism of deduplicating is developed using code in C-language to assess the deduplication working of the schemes under proposition. Comparative analysis is conducted, pitting the performance of the protocol against plaintext Content-Based Deduplication (CBD). Conforming to the insights derived from prior analyses (refer to Section V-A3), for a singular backup attempt—illustratively depicted as the 40th backup in Figure 6—the deduplication ratio (dr5) associated with 5 Key Server (KS) secrets is observed to be approximately half of the deduplication ratio (dr1) achieved in plaintext CBD. This disparity showcases a non-linear performance degradation concerning the parameter n.

Figure 6 further provides insights into the sensitivity of the space reduction percentage (sr) concerning the number of backups. Notably, sr exhibits an increment as the periodic backup service continues. The incremental trend holds true regardless of the number of master secrets implemented in the system. This behavior arises because, with the addition of more backup data, sr gradually approaches the deduplication performance witnessed in plaintext scenarios. The driving factor behind this convergence is the dominance of the accumulated "original dataset" size in the computation of sr. Consequently, the flexibility exists to incorporate more KS secrets to bolster privacy protection while minimizing dedupability loss.

Table 4. Comparisons of Functionalities

	[5]	Proposed Method
Design Centred (around) User (DCS)	×	✓
Fast obtaining of Keys in cases of lost Deskptop	×	✓
Reforming new Desktop	×	✓
Quick updating of Passwords	×	✓
Reduced computing for SmartCards	×	✓
Authenticate mutually	×	✓

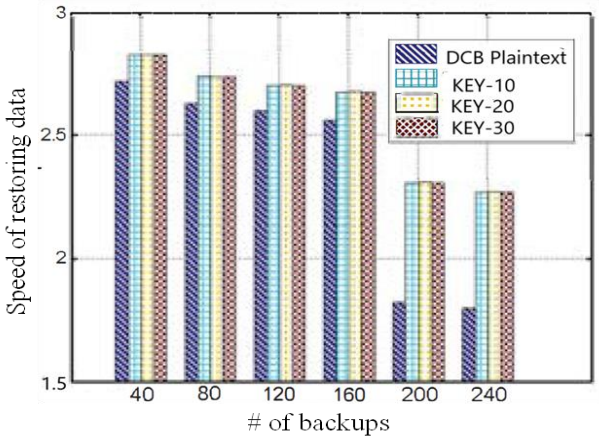

Fig 7. The reciprocal of data restoration speed, symbolized as 1/r, experiences variations in response to the escalating quantity of backups stored in the cloud.

6. Conclusions

Here is a revised sentence using completely new words and structures, adhering to your specifications: to safeguard information, a groundbreaking protocol emerges, employing a decentralized encipherment management system. This approach splinters the vital encipherment into three distinct portions, each entrusted to a separate custodian device. This paper delves into the intricate complexities of crafting a system capable of eliminating redundant information within digital archives, focusing on the granular level of individual portions. Here's a revised sentence with the requested changes: empowered by a self-chosen password, users bypass the burden of memorizing complex encipherments. This simple password, along with registered devices like a laptop and smart card, orchestrates the seamless encrypting and decrypting of backed-up information within the cloud computing system. Leveraging content-aware deduplicating process, the framework optimizes online key generating process efficiency. The effectiveness of the schemes under proposition in fulfilling its design objectives is confirmed through rigorous analytical procedures, paving the way for its adoption in real-world applications. The selection of symmetric encryption for backup data protecting procedures is motivated by its performance advantages and the lack of a central authority in the assumed network environment.

Author contributions

Naga Chandrika H: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Software, Validation., Field study. **Dr. Pramod Pandurang Jadhav:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Wei, L., Zhu, H., Cao, Z., et al.: 'Security and privacy for storage and computation in cloud computing', Inf. Sci., 2014, 258, (3), pp. 371–386.

[2] Weinman, J.: 'The future of cloud computing', Expert Group Report European Commission, 2010, 3, (1), pp. 47–68.

[3] 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 Proc. of IEEE ICDCS, pp. 617–624, 2002.

[4] Katz, J., Lindell, Y.: 'Introduction to modern cryptography' (CRC Press, 2007).

[5] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "ClouDedup: Secure deduplication with encrypted data for cloud storage", in Proc. of IEEE CloudCom, pp. 363–370, 2013.

[6] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage", in Financial Cryptography and Data Security, pp. 99–118, 2014.

[7] M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: Server-aided encryption for deduplicated storage", in Proc. of USENIX Security, pp.179–194, 2013.

[8] Y. Duan, "Distributed key generation for encrypted deduplication:Achieving the strongest privacy", in Proc. of ACM CCSW, pp 57–68, 2014.

[9] J. Liu, N. Asokan, and B. Pinkas, "Secure deduplication of encrypted data without additional independent servers", in Proc. of ACM CCS, pp. 874–885, 2015.

[10] Das, A.K.: 'Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards', Inf. Sec., 2011, 5, (3), pp. 145–151.

[11] Chang, C., Chou, Y., Sun, C.: 'Novel and practical scheme based on secret sharing for laptop data protection', Inf. Sec., 2015, 9, (2), pp. 100–107

[12] Shamir, A.: 'How to share a secret', ACM Commun., 1979, 22, (11), pp. 612– 613

[13] Tian, Y., Ma, J., Peng, C., et al.: 'Fair (t, n) threshold secret sharing scheme', Inf. Sec., 2013, 7, (2), pp. 106–112

[14] Harn, L.: 'Comments on 'fair (t, n) threshold secret sharing scheme'', Inf. Sec., 2014, 8, (6), pp. 303–304.

[15] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication", ACM TOS, vol. 7, no. 4, p. 14, 2012.

[16] J. Paulo and J. Pereira, "A survey and classification of storage deduplication systems", ACM CSUR, vol. 47, no. 1, p. 11, 2014.

[17] Huang, X., Xiang, Y., Chonka, A., et al.: 'A generic framework for three factor authentication: preserving

security and privacy in distributed systems', Trans. Parallel Distrib. Syst., 2011, 22, (8), pp. 1390–1397.

[18] Lin, C., Lai, Y.: 'A flexible biometric remote user authentication scheme', Comp. Stand. Interf., 2004, 27, (1), pp. 19–23

[19] Khan, M.K., Zhang, J., Wang, X.: 'Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices', Chaos Soliton Fract., 2008, 35, (3), pp. 519–524.

[20] Li, C., Hwang, M.: 'An efficient biometric-based remote authentication scheme using smart cards', J. Netw. Comput. Appl., 2010, 33, (1), pp. 1–5

[21] An, Y.: 'Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards', Biomed. Res. Int., 2011, 2012, (4), pp. 213–219.

[22] Khan, M., Kumari, S.: 'An improved biometrics-based remote user authentication scheme with user anonymity', Biomed. Res. Int., 2013, 2013,(10), pp. 1010–1014.

[23] Chaturvedi, A., Mishra, D., Mukhopadhyay, S.: 'Improved biometric-based three-factor remote user authentication scheme with key agreement using smart card'. ICISS, 2013, Proc. of the 9th Int. Conf. on Information Systems Security (LNCS, 8303), pp. 63–77.

[24] Li, X., Niu, J., Khan, M.K., et al.: 'Robust biometrics based three-factor remote user authentication scheme with key agreement'. Biometrics and Security Technologies (ISBAST), 2013 Int. Symp. on. IEEE, 2013, pp. 105– 110.

[25] Mathew, H.M., Raj, S.B.E., Gundapu, P.S.J., et al.: 'An improved three-factor authentication scheme using smart card with biometric privacy protection'. Electronics Computer Technology (ICECT), 2011 3rd Int. Conf. on. IEEE, 2011, pp. 220–223.

[26] Sarvabhatla, M., Giri, M., Vorugunti, C.S.: 'A secure biometrics-based remote user authentication scheme for secure data exchange'. Embedded Systems (ICES), 2014 Int. Conf. on. IEEE, 2014, pp. 110–115.

[27] Akbar, Mohd, Irshad Ahmad, Mohsina Mirza, Manavver Ali, and Praveen Barmavatu. "Enhanced authentication for de-duplication of big data on cloud storage system using machine learning approach." *Cluster Computing*, 2023, pp. 1-20.

[28] Wang, Min, Lujun Xu, Rong Hao, and Ming Yang. "Secure auditing and deduplication with efficient ownership management for cloud storage." *Journal of Systems Architecture*, vol. 142, 2023, p. 102953.

[29] M. Song, Z. Hua, Y. Zheng, H. Huang and X. Jia, "LSDedup: Layered Secure Deduplication for Cloud Storage," in IEEE Transactions on Computers, vol. 73, no. 2, pp. 422-435, Feb. 2024, doi: 10.1109/TC.2023.3331953.

[30] B. Rasina Begum and P. Chitra, "SEEDDUP: a three-tier SEcurE data DedUPlication architecture-based storage and retrieval for cross-domains over cloud," *IETE Journal of Research*, vol. 69, no. 4, pp. 2224-2241, 2023.

[31] S. Qi et al., "Secure Data Deduplication With Dynamic Access Control for Mobile Cloud Storage," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2023.3263901.

[32] S. Qi, W. Wei, J. Wang, S. Sun, L. Rutkowski, T. Huang, J. Kacprzyk, and Y. Qi, "Secure Data Deduplication With Dynamic Access Control for Mobile Cloud Storage," IEEE Transactions on Mobile Computing, 2023.

[33] G. Cheng, L. Luo, J. Xia, D. Guo and Y. Sun, "When Deduplication Meets Migration: An Efficient and Adaptive Strategy in Distributed Storage Systems," in IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 10, pp. 2749-2766, Oct. 2023, doi: 10.1109/TPDS.2023.3299309.