DedupSnap: A lightweight snapshot backup system

1st Jobin Ottaveedu
Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
jobinjosephh2005@gmail.com

2nd Anshul Purandare
Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
anshul.purandare@gmail.com

3rd Chirag Meghani
Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
meghanichirag841@gmail.com

4th Aryan Sivanandan
Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
aryan.sivanandan04@gmail.com

5th Prof. Sachin Pande
Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
sspande@pict.edu

Abstract— The rapid increase in digital data from virtualization and cloud services has intensified the demand for both space-efficient and verifiable storage systems. Traditional full-image backups are grossly inadequate due to the wasted capacity and bandwidth, while offering little assurance regarding the integrity of the data. The application of deduplication and authenticated data structures offers a way forward since it reduces redundancy and enables tamper-evident verification.

DedupSnap is a lightweight snapshot and backup framework that integrates content-defined chunking, SHA-256-based content-addressable storage, and deterministic Merkle-root manifests to create compact, auditable snapshots. This survey reviews the key building blocks behind such systems, including chunking algorithms, indexing and garbage-collection strategies, Merkle-tree engineering, and retrievability proofs like Proofs of Retrievability (PoR). We analyze trade-offs between throughput, deduplication ratio, and audit cost, and outline open challenges in adaptive chunking, index compaction, privacy-preserving deduplication, and audit–GC co-design.

By bridging storage efficiency with verifiable integrity, DedupSnap shows how cryptographic proofs and deduplication can jointly enable scalable, privacy-aware, and tamper-evident backup solutions for compliance-sensitive environments.

Index Terms—Deduplication, Content-Defined Chunking (CDC), Content-Addressable Storage (CAS), Merkle trees, Proofs of Retrievability (PoR), Privacy-preserving dedupe

I. Introduction

The explosive growth of user data, virtual machine images, and enterprise workloads has increased the burden on backup archival systems, making storage efficiency, integrity, and verifiability central design objectives for modern snapshots, and backup tools.

Conventional backup methods that save complete files or full images with each backup cycle have become less viable, as they demand substantial storage resources, generate high network transfer overhead, and hinder straightforward, tamper-evident auditing of historical backups. Industry reports and practitioner studies highlight that organizations encounter escalating expenses and operational challenges when depending exclusively on certain traditional solutions for frequent, long-retention snapshots and SaaS data protection.

Recent developments continue to highlight the importance of verifiable deduplication in both academic and enterprise contexts. Zhang et al. [13] and Yu et al. [14] show that integrating proof-of-integrity with deduplication reduces audit cost while maintaining data privacy, motivating DedupSnap's own design emphasis on cryptographically linked proofs. Broader surveys such as Amdewar and Sudhakar [15] and Kaur et al. [4] reinforce the observation that chunking strategy, hash choice, and index granularity remain the dominant trade-offs affecting deduplication scalability and storage efficiency.

Further down the line, new works like FASTEN [16] and PM-Dedup [18] introduce fault-tolerant and edge-aware deduplication variants, respectively, reflecting an interesting convergence between cloud backup, edge migration, and verifiable storage paradigms. DedupSnap further advances these directions with a lighter-weight design targeted at medium-scale and compliance-sensitive customers, combining authenticated chunking with periodic Proof-of-Retrievability (PoR) challenges.

Drawing these ideas together, DedupSnap is a lightweight snapshot and backup design that uses SHA-256 keyed CAS, configurable content-defined chunking, and deterministic Merkle-root manifests to deliver space-efficient, auditable snapshots suitable for clients with sensitive data. In this survey, we synthesize algorithmic and system-level work relevant to these components, evaluate trade-offs between throughput, deduplication effectiveness, metadata cost, fragmentation, and auditability, and propose a cohesive set of recommendations and operational policies for a practical DedupSnap deployment. Our analysis is grounded in the chunking optimizations (Rabin/FastCDC and recent hash-less extrema/ram approaches), empirical deduplication studies and hybrid inline/post-process architectures, Merkle-tree standardisation and proof batching techniques, and the literature on Proofs of Retrievability and Provable Data Possession that bridge logical commitments to physical availability.

The remainder of this paper is organized as follows: Section II describes the background and motivation. Section III surveys existing work on chunking, indexing, Merkle proofs, and retrievability. Section IV outlines key challenges and future directions, and Section V concludes.

II. Background And Motivations

Backup and snapshot technologies historically aimed at durability and recoverability: storing full images or filelevel copies ensures simple restores, but at large storage and bandwidth cost. The empirical effectiveness of data-centric deduplication was demonstrated by earlier analyses by Meyer

and Bolosky [1] and Paulo and Pereira [12], but more recent methods place an emphasis on combining integrity verification with storage reduction. The dominance of CDC algorithms in backup architectures due to their adaptability to changing workloads is highlighted by recent surveys [15] and system-level experiments [3, 9]. Further illustrating how deduplication and verifiability can coexist without imposing prohibitive bandwidth or CPU cost are complementary frameworks like VeriDedup [14] and enhanced cloud auditing models [13].

First, chunk granularity and boundary selection determine how much redundancy is exposed to the index: contentdefined chunking (CDC) recovers shifted/edited duplicates and yields higher space savings than whole-file or fixed-block approaches, but CDC increases CPU effort, produces variable chunk-size distributions, and inflates metadata counts unless bounded by min/max policies.

Second, deduplication indexing and lookup designs drive peak I/O and memory pressure; pragmatic systems therefore trade some immediate deduplication for lower write latency via prioritized inline caches or post-processing pipelines, a pattern that recent hybrid proposals demonstrate effectively.

Third, cryptographic commitments and retrievability guarantees address different failure modes: a Merkle root succinctly vouches for the correctness and ordering of a snapshot's chunk identifiers but does not by itself prove physical storage of the referenced payloads — an issue resolved only when inclusion commitments are paired with PoR-style challenge with response audits, ideally using aggregation-friendly authenticators to keep audit costs modest in a deduplicated store. These interactions create operational constraints for any lightweight verifiable system: canonical leaf encodings and domain separation are required so roots are unambiguous and signable; garbage collection must be gated by retained signed roots to prevent silent loss; and privacy-preserving deduplication is necessary when cross-tenant leakage is a concern.

Taken together, these considerations motivate DedupSnap's architectural choices: selectable CDC with conservative min/max bounds and fast-engineered defaults, a CAS layer with sharded indices, Bloom/LRU prefilters for inline optimization, canonical Merkle manifests with signed roots recorded in an append-only log, and sampled, aggregationfriendly PoR audits to bind logical commitments to probabilistic availability guarantees. This blueprint is designed to deliver practical space savings and verifiable auditability while preserving the operational properties required by sensitive-data clients.

The overall architecture of the DedupSnap system is illustrated in Fig, 1. User data first goes through the chunking and hashing layers, which break files into chunks of variable size and assign each chunk a unique identifier using SHA-256. The chunks are stored in a Content-Addressable Storage (CAS) layer, which is responsible for deduplication, indexing, and reference counting. The snapshot metadata module records a reference to the chunk to represent the chunk in the precise version. A Merkle tree is built over this representation; a hashed representation of the Merkle tree is produced as the root and appended to a signed audit log in a manner that can be cryptographically verified later. Meanwhile, the Proof Engine periodically generates Proof-of-Retrievability (PoR) or Proof-of-Duplication (PDP) challenges that clients or auditors can verify without retrieving the full file data. Finally, Garbage Collection (GC) reclaims any unreferenced chunks based on the retention policy; the entire process aims toward providing space efficiency, while parallelly offering verifiable integrity.

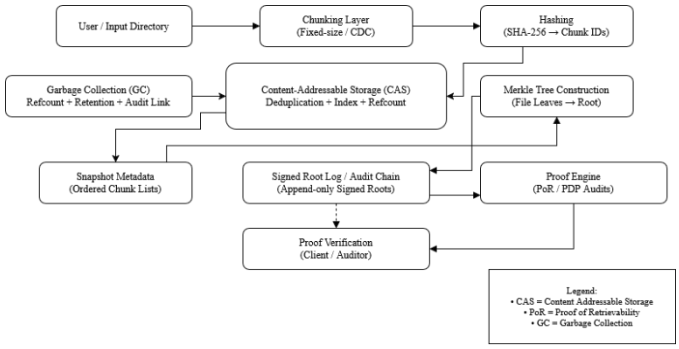

Fig. 1. Basic system architecture of DedupSnap

III. Survey Of Existing Work

The literature relevant to a verifiable, deduplicated snapshot service such as DedupSnap is interdisciplinary, and divides into a small set of tightly coupled areas: chunking and boundary selection; indexing, prefilters and garbage collection; authenticated manifests and Merkle engineering; retrievability proofs (PDP/PoR) and audit scheduling; privacy-preserving deduplication; and production systems / practitioner evidence. The subsections below summarise results in each area and emphasize the operational lessons that inform a DedupSnapclass design.

A. Chunking And Boundary Selection

Chunking determines the atomic units of deduplication and therefore has first-order impact on space savings, CPU cost, metadata volume, and fragmentation. FastCDC [10], UltraCDC [9], and AE/RAM methods [8] improve boundary detection as well. Early treatments contrast fixed-size blocking with content-defined chunking (CDC); fixed blocks are inexpensive and predictable but fail under small insertions/deletions, whereas CDC (Rabin rolling fingerprint and descendants) yields high resilience to boundary shifts at increased per-byte work and with variable chunk-size distributions. Engineering research addressed these costs by proposing lighter rolling/anchor functions and opportunistic testing: FastCDC demonstrates that careful anchor selection, min/max bounds and low-overhead rolling primitives can preserve CDC's dedupe quality while substantially increasing throughput, making CDC viable for high-rate ingestion. Parallel work on hash-less extrema and asymmetric window methods (AE, RAM, UltraCDC) shows that comparison-based anchors can reduce CPU usage and improve behavior on low-entropy runs, a favorable trade for resource-constrained clients. Practical configuration tools such as SmartChunker validate that chunk-size choice materially affects dedupe and that sampling-based estimators can recommend deploymentspecific defaults; nonetheless, no single chunker uniformly dominates across heterogeneous traces, motivating a modular chunking layer with engineered defaults and low-cost fallbacks.

New hybrid algorithms such as Dynamic Chunking [6] and DedupBench [7] offer benchmarking evidence that algorithmic selection should adjust to workload entropy, going beyond the fundamental CDC designs. According to surveys [4, 15], adaptive chunking, which may be informed by online statistics, provides realistic traces with up to 25% better deduplication ratios. Therefore, adaptive CDC switching based on data type and volatility could be implemented in future DedupSnap deployments, in line with PM-Dedup's optimization themes [18].

B. Indexing, Prefilters and Lookup Economics

At scale, fingerprint indices and associated metadata dominate cost. A flat, exact index is memory- and I/O-intensive; effective systems therefore adopt layered strategies: probabilistic prefilters (Bloom filters) reduce false lookups, small LRU caches exploit temporal locality, sparse or sampled indices reduce index footprint, and sharding (tenant or hash-prefix) distributes metadata load. Hybrid inline/post-process architectures (prioritized inline caches with deferred reconciliation) reduce peak storage and write latency in practice but introduce additional complexity for correctness, consistency and GC in distributed deployments. Recent production-oriented work quantifies these tradeoffs, showing how cache sizing, prefilter false-positive rates and index tiering directly influence write amplification, peak memory, and restore latency; these results indicate that index design must be treated as a first-class systems problem rather than an implementation detail.

The FASTEN framework [16] by Ahmed et al. shows how fault tolerance and storage efficiency can coexist when index and replication policies are co-optimized. The trade-off between replication and deduplication is still a fundamental systems challenge. Hierarchical index tiering, which spans cloud and edge nodes, can reduce active memory while

deduplication ratios, according to migration-aware designs like PM-Dedup [18]. These lessons align with DedupSnap's sharded CAS indices and Bloom/LRU prefilters, which prioritize consistent throughput under realistic retention workloads.

C. Authenticated manifests and Merkle engineering

Authenticated hash trees are the canonical mechanism for committing to large manifests with compact proofs. Merkle trees (binary or k-ary) compress a snapshot into a single root and permit O(log N) inclusion and consistency proofs. Practical research emphasizes canonical leaf encodings (domain separation for leaf/internal nodes and inclusion of reconstruction metadata such as chunk length and sequence index) to prevent ambiguity and second-preimage attacks. Engineering variants (higher-arity trees, multi-proof batching, sparse encodings) trade proof size, update cost and proof-generation latency; append-only signed root logs and periodic external anchoring are recommended to mitigate equivocation and replay. Crucially, Merkle commitments bind snapshot structure and identifiers, but they do not by themselves guarantee payload availability — the boost in auditability they provide must be combined with retrievability mechanisms for end-to-end assurance.

The validity of authenticated structures is once again highlighted with anonymity-preserving mechanisms like those by Kavita et al. [5] which sanction Merkle proofs not only for integrity checks but also for ownership validation. Variations of this kind, including VeriDedup [14] and Zhang et al. [13], utilize Merkle proofs in conjunction with proof of ownership protocols for the purposes of eliminating redundancies in uploads while maintaining audibility. These developments substantiate DedupSnap's rationale for using canonical leaf encodings and append-only signed roots as the basis for efficient verification.

D. Garbage collection and retention semantics

Garbage collection (GC) in deduplicated storage systems must reconcile two competing goals: reclaiming unused chunks to conserve space and preserving the cryptographic auditability of retained snapshots. Naïve deletion of unreferenced data can silently invalidate proofs or corrupt snapshot histories if dependencies are not carefully tracked. Modern systems therefore implement **reference counting** or **reachability analysis** across chunk graphs, ensuring that only data not referenced by any valid snapshot manifest is reclaimed. When verifiable storage is required, GC must operate under the constraint of **retention semantics tied to signed Merkle roots**—that is, any chunk reachable from a published root must remain immutable until that root is explicitly revoked. This linkage prevents replay or selective deletion attacks and ensures that audit proofs remain valid even after partial cleanup. DedupSnap extends this approach by integrating **reference tracking with audit checkpoints**, allowing space reclamation only after corresponding proofs of retrievability have been refreshed or expired. Such coupling between GC and verifiability policies reduces the risk of accidental data loss and maintains cryptographic consistency across evolving snapshot chains.

More recent work recognizes that retention semantics and GC safety form the backbone of long-term verifiable archives. For example, PM-Dedup [18] and blockchain-based GC mechanisms [17] discuss maintaining lightweight cryptographic receipts for reclaimed data to ensure continuity of audits in case of migration to cold tiers. Integrating such receipts or proof-of-retention hashes with the signed-root framework of DedupSnap could further reduce verification downtime and strengthen compliance visibility across successive snapshot generations.

The literature provides a validated component set for DedupSnap engineered CDC defaults, Bloom/LRU prefilters, sharded CAS indices, canonical Merkle commitments, sampled PoR audits and server-aided keying but the integration challenges remain substantial. Key open directions identified across the surveyed work include adaptive online chunking that approaches offline-optimal parameters without re-chunking, principled index compaction and tiering that preserves proof issuance capability, aggregation-friendly and privacy-aware audit tags, and standardized benchmarks and mixed-workload traces for reproducible evaluation. Addressing these gaps requires coordinated advances across systems engineering, applied cryptography and community benchmarking; the nearterm literature nevertheless suggests a practical DedupSnap blueprint that can be implemented and iterated in real deployments.

IV. Challenges and Future Directions

Compaction and tiering introduce more subtle risks to correctness and recovery, since losing Merkle node mappings, or serializing them inconsistently, can break proof generation as the index entries are relocated or pruned. Hence, there is a pressing need for designs that explicitly couple index tiering and compaction policies with proof-generation costs, so Merkle node persistence, partial tree caching, and audit issuance remain possible even following data migration to colder tiers. Providing clear quantification of index memory overhead per terabyte for realistic retention windows, together with demonstrations of low-cost online compaction with preservation of proof fidelity, would go a long way to lowering the operational barrier to adoption.

Another open frontier is that combining proof-of-storage economics with data deduplication incentives. As pointed out by Dorsala et al. [17], blockchain-based mechanisms can reward verifiers for maintaining retrievability proofs and thereby provide a way to decentralize audit infrastructures. Similarly, message-locked proofs of retrievability [20] are a unified construct where deduplication and proof generation share the same authenticators and reduce the overall audit cost. Extending DedupSnap with such mechanisms could yield verifiable backup ecosystems that are both economically and cryptographically self-sustaining.

Going forward, meaningful progress will necessitate co-design across chunking algorithms, index economics, privacy-preserving cryptography, audit theory, and system operations. Future solutions should tackle adaptive chunking with no full re-chunking, proof-fidelity index tiering, bounded information leakage in cross-tenant deduplication, and auditable garbage collection tightly tied to the retrievability guarantees. Achieving these goals demands close collaboration among systems researchers, applied cryptographers, and practitioners, using open benchmarks and realistic workload traces. The payoff is an operationally credible, verifiable snapshot architecture-one that unifies efficiency, privacy, and integrity within practical storage infrastructures.

Finally, integrity benchmarking remains largely uncharted territory. Benchmarking tools such as DedupBench [7] and empirical auditing frameworks [13,14] bring forth the need for reproducible datasets and standardized metrics in order to comparatively study verifiable deduplication systems. A community-driven benchmark suite that integrates chunking performance, index overhead, proof latency, and GC safety would bring about the much-needed transparency for the mainstream adoption of solutions like DedupSnap.

V. Conclusion

This survey examined the algorithmic primitives, system architectures, and cryptographic mechanisms that together enable a lightweight, auditable, and storage-efficient snapshot service such as DedupSnap. The review shows that redundancy elimination and compact authenticated commitments must be co-designed: high-throughput content-defined chunking (with engineered CDC defaults and hash-less fallbacks for constrained clients), content-addressable storage with pragmatic indexing, deterministic Merkle-root manifests with canonical leaf encodings, and sampled retrievability audits together form a practical stack for verifiable deduplicated snapshots. No single primitive is sufficient on its own; the end-to-end properties of efficiency, integrity, and availability emerge from careful interaction among chunking parameters, index design, proof formats, and GC semantics.

Several research challenges remain. Workload-aware adaptive chunking that adjusts parameters online without full rechunking would reduce manual tuning and improve efficiency across heterogeneous datasets. Scalable index compaction and cold-tier strategies that preserve proof-generation capacity while reducing active memory footprints are needed for longretention archives. Practical, privacy preserving deduplication mechanisms that permit safe cross-tenant savings without enabling confirmation attacks require further applied cryptographic and systems innovation. Finally, audit designs that more tightly integrate retrievability checks with GC policies and economic incentives would strengthen long-term assurances for archival data.

Finally, advancing this field will require coordinated effort from academia, industry, and standards bodies. Shared benchmarks, open traces and datasets, and standardized evaluation protocols are essential to enable fair comparison and measurable progress across chunking, deduplication, authenticatedcommitment, and retrievability techniques. With such foundations in place, DedupSnap-style systems can mature into reliable, auditable, and widely adopted components of modern data protection and compliance workflows—delivering verifiable, privacy-aware, and cost-efficient snapshot services at scale. We believe such DedupSnap-style architectures can significantly influence the design of verifiable, compliance-ready data protection systems in industry and research

References

[1] Meyer, Dutch Bolosky, William. (2012). A Study of Practical Deduplication. TOS. 7. 14. 10.1145/2078861.2078864.

[2] : R.N.S. Widodo, H. Lim, M. Atiquzzaman, A new content-defined chunking algorithm for data deduplication in cloud storage, Future Generation Computer Systems (2017)

[3] Hamandawana, P.; Cho, D-J.; Chung, T-S. Speed-Dedup: A New Deduplication Framework for Enhanced Performance and Reduced Overhead in Scale-Out Storage. Electronics 2024

[4] Kaur, R., Chana, I. Bhattacharya, J. Data deduplication techniques for efficient cloud storage management: a systematic review. J Supercomput 74, 2035–2085 (2018).

[5] Kavita, P. Sindhu, S. Mahalakshmi, L. Keerthana, R. Jayasri, K. Sangeetha, M.. (2025). A Privacy-Aware and Storage-Efficient Data Deduplication Scheme using Merkle Tree Proof of Ownership.

[6] X. Yuan and A. Li, "A Dynamic Chunking Algorithm Approach for Data Deduplication," 2023 8th International Conference on Information Systems Engineering (ICISE), Dalian, China, 2023

[7] A. Liu, A. Baba, S. Udayashankar and S. Al-Kiswany, "DedupBench: A Benchmarking Tool for Data Chunking Techniques," 2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Regina, SK, Canada, 2023

[8] Y. Zhang et al., "A Fast Asymmetric Extremum Content Defined Chunking Algorithm for Data Deduplication in Backup Storage Systems," in IEEE Transactions on Computers, vol. 66, no. 2, pp. 199-211, 1 Feb. 2017

[9] P. Zhou, Z. Wang, W. Xia and H. Zhang, "UltraCDC:A Fast and Stable Content-Defined Chunking Algorithm for Deduplication-based Backup Storage Systems," 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 2022

[10] W. Xia et al., "The Design of Fast Content-Defined Chunking for Data Deduplication Based Storage Systems," in IEEE Transactions on

Parallel and Distributed Systems, vol. 31, no. 9, pp. 2017-2031, 1 Sept. 2020

[11] Kaur, Ravneet, Chana, Inderveer, Bhattacharya, Jhilik, The Journal of Supercomputing: "Data deduplication techniques for efficient cloud storage management: a systematic review",2018

[12] Paulo J, Pereira J (2014) A survey and classification of storage deduplication systems. ACM Comput Surv (CSUR)

[13] Di Pietro R, Sorniotti A (2016) Proof of ownership for deduplication systems: a secure, scalable, and efficient solution. Comput. Commun 82:71–82.

[14] Jindan Zhang, Urszula Ogiela, David Taniar, Nadia Nedjah. *Improved cloud storage auditing scheme with deduplication.* Mathematical Biosciences and Engineering, 20(5):7905-7921, 2023.

[15] X. Yu et al. *VeriDedup: A Verifiable Cloud Data Deduplication Scheme With Integrity and Duplication Proof.* Aalto Research, 2023.

[16] "A survey on deduplication systems." Amdewar G., Sudhakar C. International Journal of Grid and Utility Computing, Vol 15 No 2, 2024, pp 143-159.

[17] "FASTEN: Towards a FAult-tolerant and STorage EfficieNt Cloud: Balancing Between Replication and Deduplication." Sabbir Ahmed et al., arXiv, Dec 2023.

[18] Mallikarjun Reddy Dorsala, V. N. Sastry, Sudhakar Chapram. *Blockchain-based Cloud Data Deduplication Scheme with Fair Incentives.* arXiv, Jul 2023.

[19] Zhaokang Ke, Haoyu Gong, David H.C. Du. *PM-Dedup: Secure Deduplication with Partial Migration from Cloud to Edge Servers.* arXiv, Jan 2025.

[20] "A survey on Proof of Retrievability for cloud data integrity and availability." Tan Xiao, Hijazi (et al.). 2016.

[21] D. Vasilopoulos et al. *Message-Locked Proofs of Retrievability with Secure Deduplication.* ACM, 2016.

[22]