

Secure Data Wiping for Trustworthy IT Asset Recycling: A Critical Review and Conceptual Verifiable-Sanitization Architecture

¹Kahadane Shruti Sandip, ²Prof.Bangar Abhishek P.,

³ Prof.Bhosale Sachin B., ⁴Prof.Karad Vandana A.

¹Student, Department of Computer Engineering, Jaihind College of Engineering, Pune, Maharashtra, India

^{2,3,4}Department of Computer Engineering, Jaihind College of Engineering, Pune, Maharashtra, India

Abstract- Enterprises, schools and households retire large volumes of laptops, servers, phones and storage devices each year, and the recycling or resale of these assets is only trustworthy if the data they held can no longer be recovered. Failures are rarely cryptographic; they are procedural and technical: hidden storage areas are skipped, flash translation layers retain stale copies, wiping is not verified, and no auditable evidence is produced. A structured, PRISMA-informed narrative review was carried out covering standards (NIST SP 800-88 Rev. 1, IEEE 2883-2022), empirical studies of disk sanitization, flash and SSD erasure research, cryptographic erase, and the assured-deletion literature for cloud and IoT storage, including the recent Efficient and Verifiable Scheme for Secure Data Deletion (EVSD) [20] as a related primary study. The evidence shows that (i) single-file overwriting is unreliable on flash media, (ii) device-level sanitize commands and cryptographic erase are the preferred techniques for modern drives but depend on firmware correctness, (iii) verification is the weakest link in practice, and (iv) cryptographic assured-deletion schemes offer verifiable proofs but prove only limited properties and have not been evaluated for physical asset recycling. The review contributes (i) a media-to-method evidence table, (ii) a comparative analysis separating asset-recycling evidence from cloud-deletion evidence, (iii) an eight-layer conceptual architecture for verifiable, auditable data wiping with a gated validation loop, (iv) a leakage-resistant evaluation protocol, and (v) a prioritised research agenda. The architecture is conceptual and has not been implemented or experimentally validated.

Keywords— Secure Data Wiping; Media Sanitization; IT Asset Disposition; Cryptographic Erase; Assured Deletion; Attribute-Based Encryption; Merkle Hash Tree; Data Remanence

I. INTRODUCTION

Information technology assets reach end of life long before the data they contain loses value. Laptops carry credentials and customer records, servers carry databases and backups, and phones carry personal communications.

When these assets are recycled, refurbished or resold, any residual data becomes a liability: a leak can trigger regulatory penalties, identity theft and loss of trust in the recycler. Trustworthy IT asset recycling therefore requires more than running a wiping tool; it requires that sanitization is effective for the specific media, verified, and documented in a way a third party can audit [1], [2].

The problem has been visible for decades. An early empirical study of second-hand disks found that many had been discarded with recoverable data and that sanitization practices were inconsistent [3]. Since then, storage has changed: solid-state drives (SSDs) remap writes through a flash translation layer (FTL), self-encrypting drives (SEDs) move protection into firmware, and much data now lives in cloud services whose physical media the owner never touches. Classic advice derived from magnetic media, such as multi-pass overwriting [4], does not transfer cleanly to these technologies [5], [6].

Two largely separate research communities address the problem. Storage-security and standards work asks how to destroy data on a physical device [1], [2], [5], [6]. Cryptographic work on assured deletion asks how an owner can make outsourced data

irrecoverable and obtain a proof, typically by destroying keys or modifying ciphertext [8]–[20]. The recent EVSD scheme, for example, uses elliptic-curve operations, a linear secret sharing scheme (LSSS) and Merkle Hash Trees (MHT) so that a data owner can verify that a cloud server applied a deletion operation [20]. Whether such ideas can strengthen verification in physical asset recycling has not been examined systematically.

This review has a deliberately critical scope. Its objectives are to: (1) review data remanence mechanisms across media; (2) examine sanitization techniques and standards; (3) examine verification and audit practices; (4) review cryptographic assured-deletion schemes and their guarantees; (5) analyse EVSD [20] as a related primary study; (6) compare assetrecycling and cloud-deletion evidence; (7) identify research gaps; and (8) propose a literature-supported conceptual architecture. The main contributions are a verified evidence base that separates analogous from direct evidence, an architecture that makes verification and uncertainty explicit, and an evaluation protocol designed to avoid optimistic bias.

II. REVIEW METHODOLOGY

1. Search strategy

The review followed a structured narrative design informed by the PRISMA 2020 reporting guideline [25]. Search strings combined terms such as "secure data wiping", "media sanitization", "IT asset disposition", "data remanence", "SSD secure erase", "cryptographic erase", "assured data deletion", "verifiable deletion", "attribute-based encryption deletion" and "Merkle hash tree deletion proof". Sources searched were standards bodies, publisher platforms, conference proceedings and institutional repositories. The window prioritised recent work (2018 onward) while retaining foundational studies on disk sanitization and flash erasure.

Note for authors: database-level hit counts (IEEE Xplore, Scopus, Web of Science, ACM DL) must be logged and reported here before submission; they are intentionally not invented in this draft. Placeholders in Fig. 1 are marked [n = ?]. **2.2**

2. Eligibility criteria and screening

Table 1: Eligibility criteria applied during screening

Criterion	Included	Excluded
Topic	Media sanitization, data remanence, secure erase, crypto erase, asset disposition, assured/verifiable deletion in cloud/IoT	Pure access-control work with no deletion or sanitization element; unrelated forensics
Evidence type	Peer-reviewed articles, conference papers, reviews; standards documents	Vendor marketing, unpublished theses, venues whose peer review cannot be verified
Verifiability	Bibliographic record confirmed against publisher, DOI or standards body	Records whose authors, venue or DOI cannot be confirmed
Period	Priority 2018 onward; earlier foundational studies	Superseded duplicates

Each record was screened by title and source, then assessed at abstract or full-text level. Where only metadata or an abstract was accessible, this is stated in the comparative tables and the corresponding claims are limited accordingly.

3. Study selection

The selection flow follows the structure of Fig. 1: identification [n = ?], title/source screening with documented exclusions [n = ?], records not retrievable [n = ?], abstract/full-text assessment [n = ?], and retained sources. In this draft the retained evidence base comprises two standards [1], [2], empirical or experimental storage studies [3]–[7], and cryptographic assured-deletion studies [8]–[20]. Foundational references on algorithms and reporting (ABE, ECC, MHT, PRISMA, leakage) are cited for definitions only.

III. DATA REMANENCE MECHANISMS AND SANITIZATION TARGETS

1. Why deleted data persists

Operating-system deletion removes directory entries, not data. Even after a full format, user data may remain in unallocated sectors, swap and hibernation files, backups, and application caches. Beyond the logical layer, physical media retain data in ways that differ by technology. Magnetic disks may retain data in sectors that are remapped or hidden from the operating system (for example host-protected or reallocated areas). Flash media keep stale copies because the FTL redirects writes to new cells and erases whole blocks later, so overwriting a logical address does not overwrite the old physical page [5]. Volatile memory can retain contents for seconds to minutes after power loss, which has enabled cold-boot recovery of encryption keys [7]. Early studies of discarded drives confirmed that remanent data on retired assets is a practical, not only theoretical, risk [3].

2. Standards and sanitization categories

NIST SP 800-88 Rev. 1 groups sanitization into **Clear** (logical techniques against simple recovery), **Purge**

(techniques that make recovery infeasible even with laboratory methods) and **Destroy** (physical destruction) [1]. The standard also recognises cryptographic erase, which sanitizes by destroying the encryption key of a self-encrypting or softwareencrypted volume, as a purge technique when the encryption and key handling are sound. IEEE Std 2883-2022 extends this guidance with media-specific sanitization methods and verification expectations [2]. The practical consequence is that the correct method depends on media type, data sensitivity and whether the asset will leave the owner's control.

3. Implications for wiping

Table 2 summarises media-specific evidence. Three implications follow. First, no single method is correct for every medium; a recycling workflow must identify the media first. Second, overwrite-based methods that rely on logical addressing are least reliable on flash [5], [6]. Third, device-level commands and cryptographic erase delegate trust to firmware, which has been shown to be imperfect on some drives [5], so verification cannot be skipped.

Table 2: Media types, remanence mechanisms and evidence on sanitization

Media	Remanence mechanism	Methods reported	Main findings	Ref.
Magnetic HDD	Unallocated sectors; remapped or hidden areas	Overwrite, ATA secure erase, degauss, shred	Single-pass overwrite generally regarded as adequate for modern drives; hidden areas must be addressed	[1], [4]
Flash / SSD	FTL remapping, overprovisioning, stale pages	Device sanitize commands, crypto erase, destroy	File-level overwriting often leaves copies; some drive-level commands were found unreliable	[5], [6]
Selfencrypting drives	Key stored in firmware	Crypto erase (key destruction)	Effective if key hierarchy and firmware are correct; correctness is hard to verify externally	[1], [2]
Volatile memory (DRAM)	Short-term charge retention	Power-off delay, cold-boot mitigations	Keys recoverable shortly after power loss	[7]
Cloudoutsourced data	Replicas, backups, provider control	Key deletion, policy-based assured deletion	Proofs possible under semi-honest assumptions	[8]–[20]

IV. DATA-WIPING AND DELETION TECHNOLOGIES

Table 3: Technologies relevant to trustworthy asset recycling

Technology	Principle	Strengths	Limitations	Recycling suitability	Ref.
Software overwrite	Write patterns to logical blocks	Simple, no special hardware	Unreliable on flash; skips hidden or remapped areas; slow on large drives	Acceptable for HDD Clear with verification	[1], [4]
ATA / NVMe sanitize commands	Firmware-level block or crypto erase	Fast, reaches hidden areas	Depends on firmware correctness; support varies	Preferred for SSD when verified	[2], [5]
Cryptographic erase	Destroy media encryption key	Near-instant, scales to large drives	Needs prior full-disk encryption and sound key management	Strong where encryption was enabled from first use	[1], [2]
Degaussing	Strong magnetic field	Effective on magnetic media	Useless on flash; renders HDD inoperable	Destroy category; prevents reuse	[1]
Physical destruction	Shredding, disintegration	Highest assurance	No reuse; environmental cost	Last resort / highest sensitivity	[1]
Cryptographic assured deletion (cloud)	Key or ciphertext modification with proof	Remote, verifiable proof	Trust assumptions; proofs limited in scope	Relevant to cloud data, analogous for assets	[8]–[20]

1. Overwrite and the legacy of multi-pass methods

Multi-pass overwrite schemes originated in analyses of magnetic recording [4]. Their relevance to current high-density drives is limited, and standards now emphasise verified single-pass or device-level methods for modern media [1]. On flash media, overwriting cannot guarantee that every physical copy is replaced [5].

2. Flash erasure and device commands

Experimental work on SSDs found that sanitizing a single file by overwriting was usually ineffective, while whole-drive sanitize commands were more reliable but not universally correct across tested drives [5]. A systematisation of secure deletion further showed that guarantees differ sharply by layer (application, file system, storage) and that few solutions are verified end to end [6]. This establishes the need for independent verification of any sanitization claim.

3. Cryptographic erase and key management

Cryptographic erase converts the problem of erasing a large medium into destroying a small key. Its assurance rests on three conditions: data was encrypted with a strong cipher from first use, keys were never copied to unprotected locations, and key destruction is itself irreversible. Cold-boot results show that keys in memory can be exposed if poweroff procedures are careless [7]. Crypto erase therefore reduces but does not remove the verification burden.

V. CRYPTOGRAPHIC ASSURED DELETION: METHODS AND EVIDENCE

1. Families of schemes

The assured-deletion literature for cloud storage falls into three families. *Time-based* schemes delete keys after a predetermined interval, often using distributed hash tables, as in Vanish [8] and its refinements. *Policy-based* schemes bind keys to

access policies so that deletion can be expressed by revoking attributes or policy components [9]–[11]. *Attribute-based* schemes use key-policy or ciphertext-policy attribute-based encryption (KP-ABE, CP-ABE) to support fine-grained control and revocation [12]–[17]. Most schemes inherit bilinear-pairing operations, which are costly on constrained devices [22], [23].

2. Verification as a first-class goal

Several recent schemes add proofs that deletion occurred. Xue et al. verify deletion through a Merkle Hash Tree over ciphertext components [13]; Cheng et al. combine CP-ABE and linear secret sharing with ciphertext updates verified through an MHT [15]; later work extends these ideas to fog, IoT and expressive policies [12], [14], [17], [19]. The shared pattern is that the server transforms ciphertext so that revoked users can no longer decrypt, and returns hashes that the owner can compare with locally computed values.

3. Computational efficiency

Bilinear pairing is the dominant cost in many ABE schemes, and analyses cited in the literature estimate that roughly twenty ECC scalar multiplications correspond to one pairing [22], [23]. This motivates pairing-free constructions for lightweight devices [21], [22], including EVSD [20].

VI. REVIEW OF EXISTING RESEARCH

1. Asset-level sanitization studies

The empirical record on retired storage is thin and dated. The early study of used disks demonstrated that sanitization failures were common among real disposals [3], and the SSD study demonstrated technical reasons why such failures persist on flash [5]. The systematisation of secure deletion [6] provides the most complete cross-layer view but is itself a survey rather than an evaluation of recycling workflows. Standards [1], [2] define methods and verification expectations but do not report field failure rates. No reviewed source provides a large, recent, independent audit of sanitization outcomes in commercial IT asset disposition (ITAD) operations.

2. Cryptographic deletion schemes

Cryptographic schemes report formal security arguments and micro-benchmarks. The ordinary pattern is a security reduction (often to Diffie-Hellman or similar assumptions), an operation-count comparison with prior schemes, and timing on a single laptop-class machine [13]–[20]. These studies address data held by a cloud provider, not physical devices handed to a recycler.

3. Critical analysis of EVSD [20]

EVSD is the most recent related primary study provided for this review. Its design has eight algorithms (Setup, Registration, Key-Generation, Encryption, Decryption, DelRequest, DataDeletion, Verify). The data is encrypted symmetrically; the symmetric key is wrapped with ECC scalar multiplication; access is expressed through an LSSS matrix; deletion multiplies the stored ciphertext components by a random value *del*; and verification compares three MHT roots recomputed by the owner against those returned by the cloud server [20]. Reported results include encryption of about 10 ms with 5 attributes and 31 ms with 15 attributes for a 1 MB file on a laptop, partial decryption growing from 11 ms to 32 ms, and final decryption constant at about 1 ms, with a smartphone test showing the same trends at somewhat higher cost [20].

The scheme's strengths are clear: pairing-free operations, outsourced partial decryption, a fine-grained policy, and a compact proof. Several limitations deserve attention when considering transfer to asset recycling:

- **Scope of the proof.** Verification shows that the server applied the transformation to the current ciphertext and produced matching hashes; it does not by itself show that the *original* ciphertext or other replicas and backups were destroyed. A server that retains a copy of the pre-deletion ciphertext remains outside what the scheme can detect. In physical recycling, the analogue is a wiped drive whose hidden areas still hold data.
- **Owner-side state.** Verification requires the owner to recompute roots from the original ciphertext components, which implies retaining them, in tension with fully outsourced storage.

- **Trust model.** Table 1 of [20] marks the scheme as not relying on a third party, yet partial decryption is performed by the Central Authority; the trust placed in the CA should be stated explicitly. **Security argument.** The proof is a DDH-style reduction for ciphertext confidentiality; it does not formally prove an unrecoverability or deletion-soundness property, and the stated advantage is given inconsistently (as half of the adversary advantage, and in terms of a different symbol in the derivation).
- **Reporting consistency.** The text gives the 15-attribute encryption time as 28 ms in one place and 31 ms in another (with the 1 MB setting), and tests are limited to a laptop and one smartphone; the scheme has not been tested on real constrained IoT hardware.

These observations do not diminish the technical contribution; they illustrate the same pattern seen in sanitization research: claims are strongest for the verified operation and weakest for what happens to unobserved copies.

4. Thematic synthesis

Across the literature, four patterns emerge. (i) **Methods exist; assurance is the gap.** Both physical and cryptographic approaches can make recovery infeasible under assumptions, but assumptions are rarely verified in the field. (ii) **Media and layer matter.** Guarantees at the file or logical level do not extend to firmware-managed storage [5], [6]. (iii) **Proofs are narrow.** Cryptographic proofs establish that a defined operation occurred, not that no other copy exists [20]. (iv) **Evaluation is lab-scale.** Both communities rely on small benchmarks on a few devices, with few independent audits.

VII. COMPARATIVE LITERATURE ANALYSIS

Reported results are reproduced as published and are not directly comparable because threat models and media differ.

‘NR’ denotes information not reported in the accessible source.

Table 4: Comparative analysis of asset-level sanitization studies and standards

No.	Source	Objective	Media	Method	Main finding	Limitations
1	Garfinkel & Shelat, 2003 [3]	Survey used-disk sanitization practice	Secondhand HDDs	Forensic examination of purchased drives	Recoverable data common on disposed drives	Dated; HDD only
2	Gutmann, 1996 [4]	Analyse magnetic remanence and overwrite	Magnetic media	Theoretical and technical analysis	Origin of multi-pass overwrite patterns	Not applicable to modern highdensity drives
No.	Source	Objective	Media	Method	Main finding	Limitations
3	Wei et al., 2011 [5]	Evaluate erasure on SSDs	Flash SSDs	Experimental testing of commands and overwrite	File overwrite often leaves copies; some drive commands unreliable	Limited drive models; older firmware
4	Reardon et al., 2013 [6]	Systematise secure deletion	All layers	Survey and taxonomy	Guarantees differ by layer; few verified end to end	No field audit

5	Halderman et al., 2008 [7]	Recover keys from DRAM	Volatile memory	Cold-boot experiments	Keys recoverable after power loss	Specific hardware generation
6	NIST SP 800-88 Rev. 1, 2014 [1]	Define sanitization practice	All	Standard (Clear/Purge/Destroy)	Method depends on media and sensitivity	No failure statistics
7	IEEE 2883-2022 [2]	Standardise storage sanitization	All	Standard	Media-specific methods and verification	No field data

Table 5: Comparative analysis of cryptographic assured-deletion schemes

No.	Authors, year	Objective	Technique	Verification	Reported performance	Limitations
1	Geambasu et al., 2009 [8]	Self-destructing data	Key shares in DHT	None (timebased)	NR here	Attacks on DHT; no proof
2	Tang et al., 2010/2012 [9], [10]	Overlay assured deletion	Policy-based key control	Limited	NR here	Needs key manager
3	Cachin et al., 2013 [11]	Policy-based deletion	Graph and secret sharing	Policy-level	NR here	Abstract only used
4	Yu et al., 2018 [12]	Fog-based assured deletion	CP-ABE	Partial	Timing vs. peers	Pairing cost
5	Xue et al., 2019 [13]	Revocation-based deletion	KP-ABE + MHT	MHT roots	Operation counts	Pairing cost
6	Hao et al., 2019 [14]	User-controlled IoT deletion	KP-ABE, puncturing	Proof via server	Reduced client cost	Pairing cost
7	Cheng et al., 2019 [15]	Verifiable deletion by revocation	CP-ABE + LSSS + MHT	Ciphertext update	Operation counts	Pairing cost
8	Ma et al., 2021 [16]	Verifiable deletion	CP-ABE	Verifiable	Timing vs. peers	Pairing cost
9	Mei et al., 2022 [17]	Expressive sharing and deletion	Puncturable ABE	Verifiable	Timing vs. peers	Complex policy conversion
10	Xu et al., 2024 (EVSD) [20]	Efficient verifiable deletion	ECC + CP-ABE-style LSSS + MHT	Three MHT roots	Enc. ~10–31 ms; final dec. ~1 ms (1 MB)	Single-machine evaluation; proof scope

VIII. PROPOSED CONCEPTUAL SYSTEM ARCHITECTURE

The proposed architecture is conceptual; it has not been implemented or experimentally validated. It is

derived from the requirements and failure modes identified in Sections 3–7. Solid arrows denote operational data flow; the validation loop in Layer 8 is offline and gated.

- **Layer 1 – Asset and data environment.** Retiring assets (laptops, servers, phones, removable media, cloud volumes) and their data classes. Media type, encryption state and sensitivity determine the required sanitization level [1].
- **Layer 2 – Intake and inventory.** Each asset receives a unique identifier recorded with serial numbers, owner, custody events and the originating data classification. Chain of custody begins here, since an unrecorded transfer breaks any later audit claim.
- **Layer 3 – Media profiling.** Automated detection of interface and media type (HDD, SSD, NVMe, eMMC, SED), reported capabilities, hidden or reserved areas, and whether full-disk encryption was enabled. Profiling decides which methods in Table 3 are even applicable [2], [5].
- **Layer 4 – Policy engine.** Maps data classification and media profile to a sanitization category (Clear, Purge or Destroy) using [1], [2]. Rules are versioned and explicit, so that an auditor can see why a method was selected.
- **Layer 5 – Sanitization execution.** Runs the selected method: crypto erase where encryption was sound, device sanitize commands, verified overwrite for HDD Clear, or routing to physical destruction. Execution is logged with tool and firmware versions.
- **Layer 6 – Verification.** Independent of the execution tool: sampled and full read-back where applicable, checks of hidden areas, confirmation of key destruction, and detection of tool-reported success that contradicts read-back. Evidence records (asset ID, method, parameters, results) are hashed into a Merkle structure, an idea adapted from cryptographic deletion proofs [13], [15], [20], so that individual records and the batch can be audited without disclosing data.
- **Layer 7 – Certification and audit dashboard.** Issues signed sanitization certificates, an append-only evidence log, exception reports (failed verification, unsupported media), and device or tool health. Certificates state the method category and the verification performed, not an unqualified claim that data is irretrievable.
- **Layer 8 – Validation and feedback (offline, human-in-the-loop).** Periodic forensic sampling by an independent lab of already-certified assets, review of failed or ambiguous cases, and updating of policy rules. A revised rule set is released to Layer 4 only after independent re-validation on assets not used to design it.

IX. ARCHITECTURE DESIGN RATIONALE

- **Profile before wipe.** Reported failures stem from applying a method to media for which it is unsound, such as file overwrite on flash [5]. Profiling first prevents this class of error.
- **Verification independent of the wiping tool.** A tool that reports success cannot be its own witness, since firmware commands have been found unreliable [5]. Separating execution (Layer 5) from verification (Layer 6) enforces this.
- **Cryptographic evidence, with honest scope.** Hash-tree evidence logs make tampering with records detectable and allow selective audit. Following the lesson of [20], the proof covers what was observed; the certificate must not claim more than the verification performed.
- **Policy as data.** Expressing sanitization decisions as explicit, versioned rules mirrors policy-based deletion [9]–[11], where deletion is a function of policy rather than ad hoc action.
- **Cloud and hybrid assets.** For data held in cloud services, key destruction and provider deletion proofs are the available lever [8]–[20]; Layer 6 should record the proof type and its assumptions (semi-honest server, replicas outside scope) explicitly.
- **Gated learning.** Updates to rules and thresholds pass a validation gate to prevent unverified feedback from weakening practice.

Model Evaluation and Validation Framework

Because no experimental data are reported here, this section is a proposed protocol for future prototypes; no metric values are claimed.

Table 6: Proposed evaluation metrics and their purpose

Metric	Definition / use	Why it matters
Residual-data recovery rate	Share of sanitized assets from which forensic recovery yields user data	Primary effectiveness outcome
Verification detection rate	Share of deliberately failed wipes caught by Layer 6	Tests whether verification is real
False-pass rate	Failed wipes certified as successful	Directly linked to data leakage
False-fail rate	Successful wipes flagged as failed	Operational cost and rework
Coverage of hidden areas	Proportion of reserved or remapped regions checked	Known weak point for HDD and SSD
Throughput	Time per asset and per terabyte by method	Feasibility at ITAD scale
Evidence integrity	Ability to detect tampering with log records	Auditability

Certificate accuracy	Agreement between certificate claims and independent forensic findings	Trustworthiness of attestation
Metric	Definition / use	Why it matters
Cross-device / cross-vendor generalisation	Performance on media models and firmware not used in design	Decisive test of deployability

Fault injection and ground truth. Evaluation should plant known data patterns across logical, hidden and remapped regions, then measure recovery by independent forensic tools. Deliberately defective sanitization (for example, a skipped region) must be introduced to measure whether verification catches it.

Leakage and independence. Rules tuned on one set of drives and tested on the same models give optimistic estimates, a general source of inflated results in applied ML-style evaluation [26]. Test sets should use different vendors, firmware versions and operators, and the final evaluation should be run by a party independent of the designers.

Temporal validation. Firmware updates change erase behaviour; re-testing across firmware versions is required to set re-validation intervals.

Research Gaps and Challenges

Table 7: Validated research gaps and corresponding future directions

Gap	Supporting evidence	Future direction	Horizon
No recent independent audits of ITAD outcomes	Early disk study [3]; standards give no failure rates [1], [2]	Multi-site audits with forensic sampling	Nearterm
Flash and firmware trust	Unreliable commands on tested SSDs [5]	Vendor-independent verification procedures; firmware-aware test suites	Nearterm
Verification weaker than execution	Few end-to-end verified solutions [6]	Standard verification protocols and open test corpora	Nearterm
Narrow scope of cryptographic proofs	EVSD proves recomputed-hash agreement, not absence of copies [20]	Proofs of deletion soundness covering replicas; formal unrecoverability definitions	Medium-term

Cryptographic schemes unevaluated for physical assets	Cloud-focused designs [8]–[20]	Adapt evidence-log and policy ideas to device wiping	Medium-term
Lightweight claims tested on few devices	Laptop and one smartphone in [20]	Benchmarks on real IoT and embedded hardware	Nearterm
Key management and cryptoerase assurance	Cold-boot exposure [7]; key hierarchy opaque in SEDs	Auditable key lifecycle; attestation of key destruction	Medium-term
Lack of shared datasets	No open sanitization-failure corpus	Shared fault-injected image sets and recovery benchmarks	Medium-term
Certificates not standardised	Heterogeneous vendor reports	Interoperable signed certificates with stated verification scope	Medium-term

Future Research Directions

Near-term. The most valuable immediate contribution is an independent, multi-vendor audit of sanitization on real retired assets, with fault injection and forensic recovery, to supply the failure statistics that standards currently lack. Open-source verification tools that check hidden areas and tool-reported success independently of the wiping utility can be built and evaluated now.

Medium-term. Evidence logs based on Merkle structures, with signed certificates and explicit statements of verification scope, can be prototyped and compared with current vendor reports. Cryptographic deletion proofs should be extended to address replicas and backups, and evaluated on constrained hardware.

Speculative directions. Hardware attestation of key destruction in self-encrypting media, and formal frameworks that unify physical sanitization and cryptographic assured deletion under a single threat model, remain longer-term goals whose feasibility has not been demonstrated.

Limitations of this Review and Unresolved Questions

This draft has limitations. Database-level searches and counts must still be logged and reported (Section 2). Some sources are standards or abstracts whose detailed contents were not re-analysed, and performance figures are reported as published. The EVSD analysis relies on the published text [20]; no independent reimplementations were performed.

Citations should be re-verified against the original publications before submission.

Future work must answer several questions: (1) What is the current real-world failure rate of sanitization in ITAD operations? (2) Which verification methods reliably detect failed SSD sanitization across vendors? (3) How can cryptographic deletion proofs be extended to cover unobserved copies? (4) What certificate content do regulators and customers actually need? (5) How often must procedures be re-validated as firmware changes?

X. CONCLUSION

Evidence from standards, storage experiments and cryptographic deletion research shows that data on retired assets can be made infeasible to recover, but only when the method matches the media, the method is executed correctly, and the result is verified independently. Overwriting is unreliable on flash, device-level commands and crypto erase depend on firmware and key management, and cryptographic proofs of deletion, including the efficient EVSD scheme, establish narrow properties under defined trust assumptions. The proposed eight-layer architecture addresses these weaknesses through media profiling, explicit policy, independent verification, tamper-evident evidence logs and a gated validation loop. Before it can be recommended for operational use it must be implemented and tested with fault injection, forensic recovery and independent multi-vendor evaluation.

Acknowledgements

[To be completed by the authors.]

REFERENCES

1. R. Kissel, A. Regenscheid, M. Scholl, and K. Stine, "Guidelines for media sanitization," NIST Special Publication 800-88, Rev. 1, Dec. 2014.
2. IEEE, "IEEE standard for sanitizing storage," IEEE Std 2883-2022, 2022.
3. S. L. Garfinkel and A. Shelat, "Remembrance of data passed: A study of disk sanitization practices," IEEE Security & Privacy, vol. 1, no. 1, pp. 17–27, 2003.
4. P. Gutmann, "Secure deletion of data from magnetic and solid-state memory," in Proc. 6th USENIX Security Symp., 1996.
5. M. Wei, L. M. Grupp, F. E. Spada, and S. Swanson, "Reliably erasing data from flash-based solid state drives," in Proc. 9th USENIX Conf. File and Storage Technologies (FAST), 2011.
6. J. Reardon, D. Basin, and S. Capkun, "SoK: Secure data deletion," in Proc. IEEE Symp. Security and Privacy, 2013, pp. 301–315.
7. J. A. Halderman et al., "Lest we remember: Cold-boot attacks on encryption keys," in Proc. 17th USENIX Security Symp., 2008.
8. R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in Proc. USENIX Security Symp., 2009.
9. Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "FADE: Secure overlay cloud storage with file assured deletion," in Security and Privacy in Communication Networks (SecureComm 2010), Springer, 2010, pp. 380–397.
10. Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," IEEE Trans. Dependable Secure Comput., vol. 9, no. 6, pp. 903–916, 2012.
11. C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti, "Policy-based secure deletion," in Proc. ACM SIGSAC Conf. Computer and Communications Security, 2013, pp. 259–270.
12. Y. Yu, L. Xue, Y. Li, X. Du, M. Guizani, and B. Yang, "Assured data deletion with fine-grained access control for fog-based industrial applications," IEEE Trans. Ind. Informat., vol. 14, no. 10, pp. 4538–4547, 2018.
13. L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," Inf. Sci., vol. 479, pp. 640–650, 2019.
14. J. Hao, J. Liu, W. Wu, F. Tang, and M. Xian, "Secure and fine-grained self-controlled outsourced data deletion in cloudbased IoT," IEEE Internet Things J., vol. 7, no. 2, pp. 1140–1153, 2020.
15. Y. Cheng, L. Yang, S. Yu, and J. Ma, "Achieving efficient and verifiable assured deletion for outsourced data based on access right revocation," in Cryptology and Network Security (CANS 2019), Springer, 2019, pp. 392–411.