International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Cryptographic Hashing in Cyber Security: A Review of Hash Algorithms, Collision Attacks and Forensic File Integrity Verification

Shete Priti Rahul[1], Prof. S. R. Borhade[2], Prof. M. C. Kolse[3], Prof. S. K. Said[4]
Department of Artificial Intelligence and Data Science
Jaihind College of Engineering
Pune, Maharashtra, India

Abstract—Cryptographic hash functions are a basic building block of cyber security, used to verify that files remain accurate, authentic and unaltered during storage, transmission and sharing. This paper reviews cryptographic hashing with emphasis on file integrity verification and digital forensics. It explains the properties of hash functions, namely fixed-length output, determinism, the avalanche effect, and preimage, second-preimage and collision resistance, and analyses the principal algorithms MD5, SHA-1, SHA-256, SHA-512 and SHA-3, including the Merkle–Damgård and sponge constructions. The history of practical attacks is traced, from collisions on MD5 to identical-prefix and chosen-prefix collisions on SHA-1, explaining why these algorithms must no longer be used for security. The verification process of hashing a file, storing the reference value securely, recomputing and comparing is described, together with applications in malware identification using known-file hash sets, preservation of digital evidence according to forensic guidelines, similarity hashing for detecting near-identical files, and secure software distribution. A five-layer file integrity verification architecture is proposed, along with research gaps, objectives, a comparison of algorithms and challenges such as protecting stored hashes, performance on large data and migration from legacy algorithms. The review concludes that SHA-256, SHA-512 and SHA-3, combined with protected reference values, provide reliable integrity assurance.

Index Terms—Cryptographic hashing, file integrity, SHA-256, SHA-512, SHA-3, MD5, SHA-1 collisions, digital forensics

I. INTRODUCTION

1. Motivation

When a forensic investigator copies a suspect's hard disk, a court must be convinced that the copy is identical to the original and has not been altered since. When a user downloads software, they need assurance that it has not been tampered with. In both cases, cryptographic hashing provides the answer: a short digest that changes completely if even one bit of the file changes.

The security of this approach depends on the hash algorithm. MD5 and SHA-1 were once widely used but have been broken by collision attacks [1], [2]. Modern standards such as SHA-2 [3] and SHA-3 [4] are now recommended.

This paper reviews hash algorithms, attacks on them and their use in file integrity and forensics, and proposes a verification architecture.

2. Objectives Of The Review

- To explain the properties and constructions of cryptographic hash functions.
- To compare MD5, SHA-1, SHA-256, SHA-512 and SHA-3.
- To review collision attacks and their practical impact.
- To examine hashing in file integrity, malware identification and forensics.
- To propose a secure file integrity verification architecture.

II. OVERVIEW OF KEY CONCEPTS

1. Properties Of Cryptographic Hash Functions

A hash function H maps an input of any length to a fixed-length digest. It must be deterministic and fast, and small input changes must change about half the output bits. Security requires preimage resistance (given h, hard to find m with $H(m) = h$), second-preimage resistance and collision resistance (hard to find any $m_1 \neq m_2$ with $H(m_1) = H(m_2)$). By the birthday bound, a generic collision search for an n-bit hash needs about

$$2^{(n/2)} \text{ evaluations} \quad (1)$$

so a 256-bit hash offers about 128-bit collision security.

© 2026 IJSRET

2. Classification Of Hash Algorithms

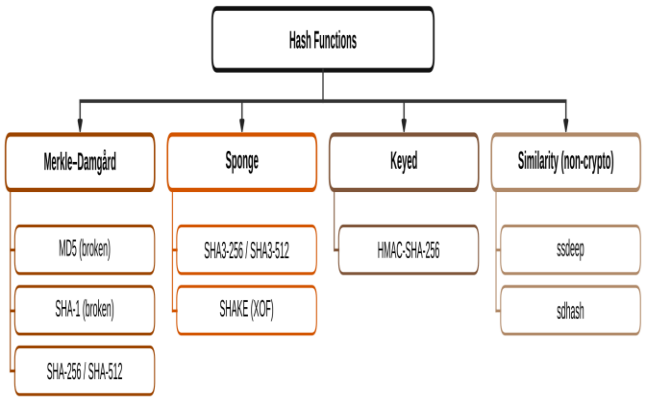


Fig. 1 classifies hash functions by construction and security status.

III. REVIEW OF EXISTING WORK

- **Constructions.** Merkle [5] described building hash functions by iterating a compression function, the basis of the Merkle–Damgård design used in MD5, SHA-1 and SHA-2 [3]. Kelsey and Schneier [6] showed that this iterated structure allows second-preimage attacks on long messages with less work than expected, motivating new designs. SHA-3 [4] uses the sponge construction based on the Keccak permutation, which is immune to length-extension attacks.
- **Attacks.** Wang and Yu [1] found practical collisions for MD5, and Wang *et al.* [7] showed that SHA-1 collisions could be found far faster than brute force. Stevens *et al.* [2] produced the first actual SHA-1 collision, two different PDF files with the same digest, and Leurent and Peyrin [8] demonstrated a practical chosen-prefix collision on SHA-1 and used it to attack the PGP web of trust.
- **Forensics.** ISO/IEC 27037 [9] provides guidelines for acquiring and preserving digital evidence, in which hash values document integrity. The NIST National Software Reference Library [10] publishes hash sets of known software so that investigators can filter known files and focus on unknown ones. Kornblum [11] introduced context-triggered piecewise hashing (ssdeep) and Roussev [12] proposed similarity digests (sdhash), both of which detect near-identical files that ordinary hashes miss. HMAC [13] adds a secret key so that an attacker who modifies a file cannot simply recompute the reference hash. Table I summarizes these works.

TABLE I Summary of Literature Survey

Ref.	Authors	Year	Technique Used	Merits	Demerits
[5]	Merkle	1987	Iterated hash construction	Basis of MD / SHA-2 designs	Length extension
[1]	Wang and Yu	2005	MD5 collision attack	Practical MD5 collisions	Algorithm-specific
[7]	Wang et al.	2005	SHA-1 cryptanalysis	Collision cost below brute force	Theoretical at the time
[6]	Kelsey and Schneier	2005	Second-preimage on long messages	Weakness of iterated hashes	Very long inputs
[11]	Kornblum	2006	Piecewise hashing (ssdeep)	Near-duplicate detection	Not cryptographic
[12]	Roussev	2010	Similarity digests (sdhash)	Robust file similarity	Not cryptographic
[2]	Stevens et al.	2017	First SHA-1 collision	Two PDFs, same digest	High compute
[8]	Leurent and Peyrin	2020	SHA-1 chosen-prefix collision	Attack on PGP web of trust	Still costly

1. Identified Research Gaps

- Legacy systems still use MD5 and SHA-1 for integrity checks.
- Reference hashes stored next to files can be replaced by attackers.
- Cryptographic hashes cannot detect similar but modified malware.
- Hashing very large evidence images is time-consuming.
- Guidance for migrating algorithms is unevenly applied.

© 2026 IJSRET

IV. SYSTEM ARCHITECTURE

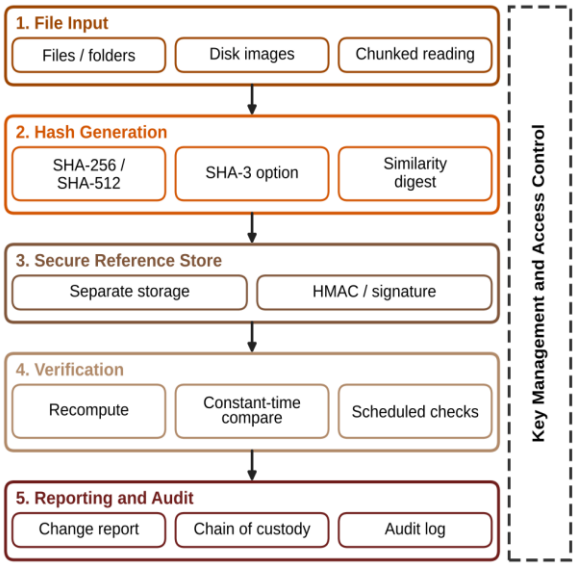


Fig. 2 shows the proposed architecture.

Fig. 2. Proposed five-layer file integrity verification architecture.

4.1 File Input Layer

Files, folders or disk images are selected; large inputs are read in chunks to limit memory use.

4.2 Hash Generation Layer

SHA-256 or SHA-512 digests are computed; SHA-3 is available as an alternative; an optional similarity digest supports near-duplicate detection.

4.3 Secure Reference Store Layer

Reference digests are stored separately with access control and protected with HMAC or a digital signature so that they cannot be silently replaced.

4.4 Verification Layer

Digests are recomputed on demand or on schedule and compared in constant time with the stored references.

4.5 Reporting And Audit Layer

Changes are reported with file path, time and old and new digests; an audit log supports chain-of-custody documentation.

4.6 Implementation Methodology

Fig. 3 describes the verification workflow. A file F is considered unchanged when

$$H(F_now) = H(F_ref) \quad (2)$$

and the reference itself is protected by a keyed tag,

$$T = HMAC_K(path \parallel H(F_ref) \parallel time) \quad (3)$$

which is checked before comparison.

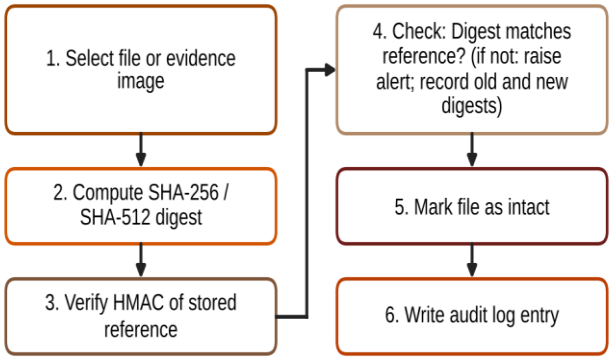


Fig. 3. Secure file integrity verification workflow.

V. PERFORMANCE COMPARISON

Table II compares hash algorithms. MD5 and SHA-1 are broken for collision resistance, while SHA-256, SHA-512 and SHA-3 remain secure. SHA-512 is often faster than SHA-256 on 64-bit processors.

TABLE II Comparison of Hash Algorithms

Algorithm	Digest (bits)	Construction	Status
MD5	128	Merkle–Damgård	Broken; avoid
SHA-1	160	Merkle–Damgård	Broken; avoid
SHA-256	256	Merkle–Damgård	Secure; recommended
SHA-512	512	Merkle–Damgård	Secure; fast on 64-bit
SHA3-256	256	Sponge	Secure; no length extension

1. Open Issues And Future Directions

Table III summarizes challenges. Future directions include tree hashing for parallel verification of large files, blockchain-anchored timestamps for evidence, and hash-based signatures that resist quantum computers.

© 2026 IJSRET

TABLE III Challenges and Mitigation

Challenge	Mitigation
Legacy MD5 / SHA-1	Migrate to SHA-256 / SHA-3
Replaced reference hashes	HMAC or signed, separate store
Modified malware variants	Add similarity hashing
Large evidence images	Chunked / tree hashing; hardware SHA
Future quantum threats	Longer digests; hash-based signatures

VI. CONCLUSION

Cryptographic hashing provides an efficient way to detect any change in a file, making it essential for file integrity, malware identification and digital forensics. The reviewed literature shows that MD5 and SHA-1 are no longer safe, while SHA-256, SHA-512 and SHA-3 remain strong. Combining modern algorithms with protected reference values and audit logs yields reliable and legally defensible integrity verification.

6.1 Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

REFERENCES

1. X. Wang and H. Yu, “How to break MD5 and other hash functions,” in Proc. EUROCRYPT, 2005, pp. 19–35.
2. M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, “The first collision for full SHA-1,” in Proc. CRYPTO, 2017, pp. 570–596.
3. National Institute of Standards and Technology, “Secure Hash Standard (SHS),” FIPS PUB 180-4, 2015.
4. National Institute of Standards and Technology, “SHA-3 standard: Permutation-based hash and extendable-output functions,” FIPS PUB 202, 2015.
5. R. C. Merkle, “A digital signature based on a conventional encryption function,” in Proc. CRYPTO, 1987, pp. 369–378.
6. J. Kelsey and B. Schneier, “Second preimages on n-bit hash functions for much less than 2^n work,” in Proc. EUROCRYPT, 2005, pp. 474–490.
7. X. Wang, Y. L. Yin, and H. Yu, “Finding collisions in the full SHA-1,” in Proc. CRYPTO, 2005, pp. 17–36.
8. G. Leurent and T. Peyrin, “SHA-1 is a shambles: First chosen-prefix collision on SHA-1 and application to the PGP web of trust,” in Proc. 29th USENIX Security Symp., 2020, pp. 1839–1856.
9. ISO/IEC 27037:2012, “Information technology—Security techniques—Guidelines for identification, collection, acquisition and preservation of digital evidence,” ISO, Geneva, Switzerland, 2012.
10. National Institute of Standards and Technology, “National Software Reference Library (NSRL).” [Online]. Available: https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl
11. J. Kornblum, “Identifying almost identical files using context triggered piecewise hashing,” Digit. Investig., vol. 3, suppl., pp. 91–97, 2006.
12. V. Roussev, “Data fingerprinting with similarity digests,” in Advances in Digital Forensics VI, IFIP AICT, vol. 337. Berlin, Germany: Springer, 2010, pp. 207–226.
13. H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-hashing for message authentication,” IETF RFC 2104, 1997.

© 2026 IJSRET