Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Verification Method and Framework for the Digital Holy Quran

Wail Zita

wailzitazita@gmail.com

Istanbul Medipol University https://orcid.org/0009-0007-0744-7795

Nassma Zita

Bahçesehir University https://orcid.org/0009-0007-4707-3396

Mohammed Sayim Khalil

Haliç University https://orcid.org/0000-0002-1539-5629

Rawan Ibrahim Mohamed ElShenieky

Haliç University

BÜŞRA ŞAHİN

Haliç University

Research Article

Keywords: Digital Quran verification, text authenticity, digital integrity, certification-based authentication, visual tamper detection, digital preservation, cybersecurity

Posted Date: May 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9577829/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Verification Method and Framework for the Digital Holy Quran

Wail Zita [1,3*], Nassma Zita [2], Mohammed Sayim Khalil [3], Rawan Ibrahim Mohamed ElShenieky[3], and BÜŞRA ŞAHİN[3]

[1]Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey
[2]Department of Computer Engineering, Bahcesehir University, Istanbul, Turkey
[3]Department of Computer Engineering, Haliç University, Istanbul, Turkey
∗Correspondence to: Wail Zita, wailzitazita@gmail.com .
[N.Z]nassmazita01@gmail.com,[M.S.K]sayimkhalil@halic.edu.tr,
[R.I.M.E]rwn@alshenieky.com, [B.Ş]busrasahin@halic.edu.tr

Submission Status

This manuscript is under peer review. The content may change following review.

Abstract

The rapid digitization of the Holy Quran across mobile applications, web platforms, and digital archives necessitates rigorous mechanisms to preserve textual authenticity and prevent unauthorized modification. This paper presents a formal cryptographic framework for Digital Quran (DQ) verification that integrates canonical Arabic text normalization, hierarchical SHA-256 hashing, and visual authentication within a unified multi-layer architecture. Unlike existing approaches that rely primarily on text matching or image-based watermarking, the proposed framework establishes a deterministic and reproducible integrity model for Quranic text at multiple structural levels. The method applies canonical normalization to resolve Unicode-related inconsistencies in Arabic script, then computes SHA-256 hashes at the word level, aggregates them into ayah-level digests, and recursively binds them into surah-level cryptographic fingerprints. This hierarchical construction ensures strong tamper sensitivity, preservation of textual order, and structural integrity across different granularities. A certification layer further binds computed integrity references to digitally signed and timestamped records, supporting non-repudiation and long-term traceability. Experimental evaluation was conducted on multiple Quranic chapters using controlled orthographic and formatting variants. The results demonstrate two complementary properties of the

proposed framework: strict sensitivity to meaningful textual modifications such as diacritic removal, and robustness against irrelevant formatting perturbations such as redundant whitespace and tatweel after normalization. These results provide evidence that this integrity model is deterministic, accurately reproduced throughout hierarchies, and can be reliably used in real-world applications. This modular and extensible trust framework can be used to provide a trusted system to maintain the integrity of the Quran, while also providing a generalised solution to safeguard other sensitive linguistic and cultural heritage digital documents.

Keywords: Digital Quran verification, text authenticity, digital integrity, certification-based authentication, visual tamper detection, digital preservation, cybersecurity.

1 Introduction

Digital platforms have grown quickly, fundamentally changing how the access, distribution, and preserving religious texts. The Holy Quran, as a fixed and canonical religious text, needs to be exceptionally preserved so that every digital presentation retains its original language structure, spelling accuracy, and meaning. As more people have access to different types of Quranic applications, online databases containing copies of the Quran, and scanned copies (known as Mushaf), the need to maintain authenticity in digital forms is no longer just a technical concern but also an academic obligation. Early computational efforts toward Quran verification focused primarily on linguistic and rule-based validation. Shamsudin and Farooq [1] introduced one of the earliest artificial intelligence approaches for structural analysis of Quranic text using meta-syntactic decomposition. Their work demonstrated the feasibility of automated anomaly detection but did not address cryptographic integrity or cross-platform consistency challenges. Similarly, Alshareef and El Saddik [2] proposed a rule-based algorithm for verifying Quranic quotations on web platforms, relying on pattern matching and textual comparison. While effective for quotation authentication, this approach remained limited to surface-level text verification without establishing immutable cryptographic binding.

Parallel research explored image-based authentication techniques for digital Mushaf copies. Khalil *et al.* [3] proposed a two-layer fragile watermarking mechanism secured with chaotic maps for detecting tampering in Quranic images. Complementary work by Kurniawan *et al.* [4,5] further developed watermark-based tamper detection frameworks capable of identifying pixel-level modifications in scanned Quran pages. These methods significantly improved visual integrity assurance but operated independently from text normalization and cryptographic canonicalization processes.

Despite these advancements, existing approaches tend to treat textual verification and image authentication as separate problems. Moreover, most systems rely either on pattern matching or pixel-level watermarking without addressing a critical challenge in Arabic digital processing: canonical equivalence under Unicode encoding. Identical Quranic words may be represented using distinct binary sequences due to diacritic composition, character decomposition, or formatting variations. Such inconsistencies can produce false mismatches or, conversely, allow subtle manipulations to bypass superficial verification mechanisms.

Canonical Unicode Normalization (UCN) is the method used to change text into a unique standardized form based on Unicode, where the same binary representation is used to encode semantically similar characters. In Arabic digital text, identical-looking characters can be represented by completely different Unicode forms due to the placement of a diacritic or the decomposition of the character itself. By enforcing the use of UCN, you ensure that two or more equivalent words from the Quran generate the same binary sequence prior to computing a cryptographic hash, which prevents false mismatches when performing an integrity check.

To address these limitations, this paper proposes a unified Digital Quran Verification Framework that integrates canonical text normalization, hierarchical cryptographic hashing, digital certification, and visual authentication within a single trust architecture. The framework introduces a deterministic multi-level SHA-256 hash construction operating at word, ayah, and surah granularity. By recursively binding word-level digests into higher-level cryptographic fingerprints, the system ensures strong tamper sensitivity while preserving structural hierarchy.

In addition to cryptographic integrity enforcement, the framework incorporates certification-based validation mechanisms capable of producing signed and timestamped integrity references, thereby supporting non-repudiation and long-term traceability of verified digital Quran copies. Unlike prior isolated solutions, the proposed model unifies linguistic consistency control with cryptographic assurance and image authentication, creating a scalable and extensible preservation architecture.

The primary contributions of this work are summarized as follows:

- A hierarchical cryptographic hash construction model for Quranic text verification at word, ayah, and surah levels.
- Integration of canonical normalization mechanisms to eliminate Unicode-induced inconsistencies in Arabic text.
- A unified multi-layer verification framework combining text integrity, digital certification, and fragile watermark-based visual authentication.
- A multi-surah experimental validation study using controlled orthographic and formatting variants to demonstrate deterministic reproducibility, hierarchical tamper propagation, and normalization-aware stability.

By combining cryptographic formalization with canonical text processing and visual integrity techniques, the proposed framework advances the state of digital Quran preservation and establishes a foundation for safeguarding sensitive linguistic and cultural heritage documents in modern digital ecosystems.

Figure 1 illustrates the hierarchical cryptographic architecture of the proposed Digital Quran Verification Framework. The system operates through a layered verification pipeline beginning with canonical normalization of the input text to enforce Unicode consistency and eliminate encoding ambiguities. The normalized content is then processed through a deterministic hierarchical SHA-256 hashing mechanism, where each word is individually hashed, aggregated at the ayah level, and recursively bound into a surah-level cryptographic fingerprint.

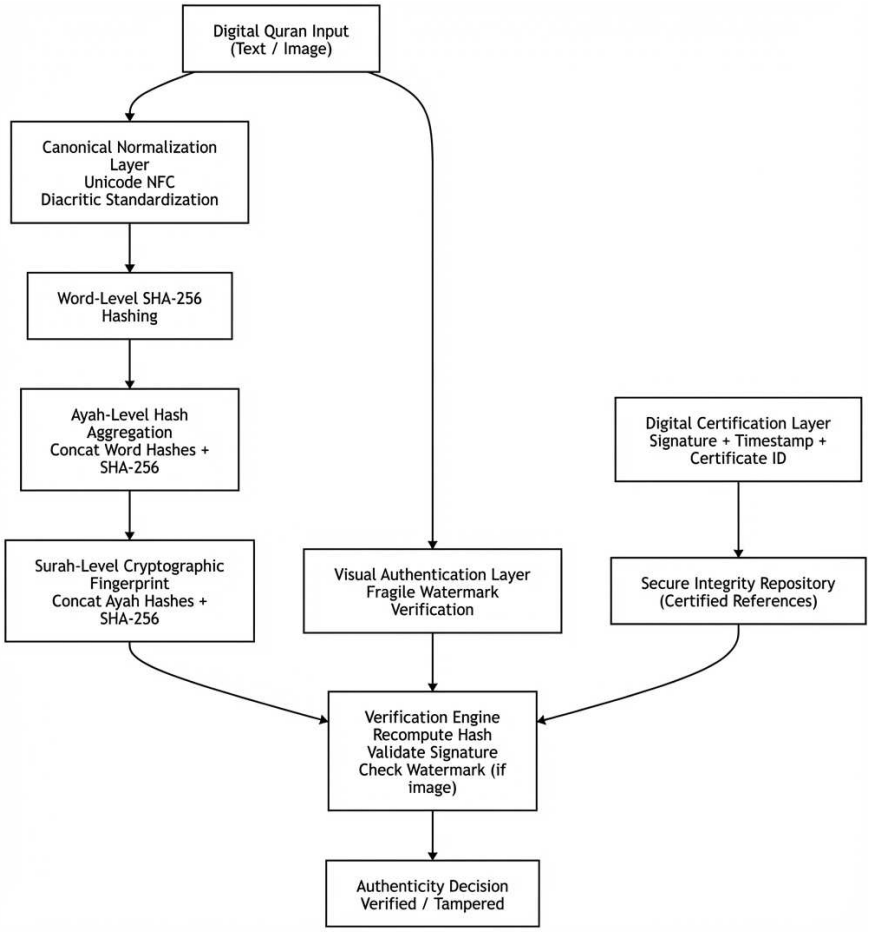


Figure 1: Hierarchical cryptographic architecture of the proposed Digital Quran Verification Framework integrating canonical normalization, multi-level SHA-256 hashing, visual authentication, and certification-based trust validation.

For image-based representations, a parallel visual authentication layer performs fragile watermark verification to detect pixel-level modifications. The resulting integrity references are validated through a certification-based trust model, which anchors signed and timestamped records within a secure integrity repository. The engine computes the hierarchical hash, validates the certification credentials, and checks the visual integrity before issuing a final authenticity decision to verify clothing items—and will be done in one unified platform that ensures structure integrity, tamper-sensitivity, and long-term traceability for many digital representations of the Quranic contents.

2 Related Work

The verification of digital representations of the Holy Quran lies at the intersection of computational linguistics, cryptography, digital watermarking, and long-term digital preservation. Prior work may be organized into four closely related directions: (i) text-based Quran authentication, (ii) image- and document-based watermarking methods, (iii) general crypto-

graphic and preservation-oriented integrity mechanisms, and (iv) Arabic text normalization and structured digital information management. This classification helps position the proposed framework within both Quran-specific and broader digital integrity research.

2.1 Text-Based Authentication Approaches

Early Quran verification studies focused primarily on linguistic and rule-based validation. Shamsudin and Farooq [1] introduced an artificial intelligence-oriented meta-syntactic framework for analyzing Quranic text structure and detecting inconsistencies at the linguistic level. This line of work demonstrated the feasibility of automated Quran analysis, but it did not provide cryptographic guarantees of integrity and was not designed to address encoding-level ambiguity.

Alshareef and El Saddik [2] proposed a rule-based algorithm for authenticating Quranic quotations retrieved from online materials using pattern matching and textual comparison. Related work by Alginahi *et al.* [7], El-Sakka [9], Kamsin *et al.* [10], and Sabbah and Selamat [11] further extended Quranic quotation verification and authenticity checking to web environments, online forums, and broader digital content settings. Al-Khalaf and Nayfeh [8] also addressed integrity checking of Quranic script in a more formalized setting.

Collectively, these studies are important because they address the problem of textual correctness and authenticity in digital environments. However, they remain largely dependent on linguistic rules, textual similarity, or application-specific checking logic. As a result, they do not provide strong bit-level immutability, hierarchical cryptographic propagation, or certification-oriented non-repudiation. They are also vulnerable to encoding inconsistencies and orthographic representation differences that can arise in Arabic digital text.

2.2 Watermarking and Image-Based Verification

A second major line of research has addressed the integrity of digital Mushaf images and scanned Quran documents through watermarking and steganographic techniques. Khalil *et al.* [3] proposed a two-layer fragile watermarking scheme secured with chaotic maps for authenticating digital Holy Quran images. Their approach demonstrated strong tamper sensitivity for image-based Quran content and showed that fragile watermarking can effectively reveal pixel-level modifications.

Related contributions by Kurniawan *et al.* [4,5] developed image authentication and tamper detection mechanisms specifically for scanned Quran pages. Al-Ahmad and Alshaikhli [6] extended similar ideas to PDF-based digital Quran documents by combining fragile watermarking with encryption-oriented protection. More generally, foundational works on digital watermarking and steganographic protection, such as Cox *et al.* [15] and Hetzl and Mutzel [16], provide the theoretical background for understanding how integrity markers can be embedded and verified in digital media.

These approaches are highly relevant for protecting visual Quran representations, especially scanned pages and document images. However, they operate primarily at the image or document layer and do not resolve a central issue in Arabic digital text: semantically equivalent text may still appear in distinct binary forms due to Unicode composition, diacritics, or

orthographic variation. Consequently, watermarking-based solutions remain complementary to, rather than substitutes for, text-aware cryptographic verification.

2.3 Cryptographic Integrity and Digital Preservation Principles

The broader foundations of digital integrity verification are rooted in cryptographic hashing, timestamping, and fixity preservation. Rivest [12] introduced the MD5 message-digest algorithm, which played an important historical role in fixity verification, while Haber and Stornetta [13] established the principle of cryptographic timestamping for verifiable digital records. Although stronger modern hash functions are now preferred, these works remain foundational in the design of digital integrity systems.

Within digital preservation, the National Digital Stewardship Alliance guidelines [14] emphasize routine fixity checking as a core requirement for ensuring that preserved digital objects remain unchanged over time. More recent blockchain-oriented studies, such as Casino *et al.* [21] and Hewa *et al.* [22], extend the discussion toward decentralized trust, traceability, and tamper-evident integrity anchoring. These studies are relevant to the certification and long-term preservation aspects of digital authenticity, particularly where auditability and distributed trust are important.

Despite their importance, general cryptographic and preservation frameworks typically operate at coarse granularity, such as files, records, or complete digital objects. They do not usually model the internal linguistic hierarchy of structured sacred texts, nor do they incorporate Arabic-specific normalization before integrity computation. The present work differs by binding integrity recursively across word, ayah, and surah levels rather than only treating the Quran as a flat digital artifact.

In addition to cryptographic integrity mechanisms, system-level security and trust-oriented frameworks have emphasized the importance of secure authentication and resilient infrastructure in protecting digital assets. For instance, authentication architectures based on QR code login and role-based access control have been proposed to mitigate unauthorized access in web environments [25]. Similarly, recent work on supply chain security highlights the necessity of addressing multi-layer vulnerabilities across hardware, software, and deployment stages to ensure end-to-end system trust [26]. These perspectives reinforce the need to complement integrity verification with broader security mechanisms in the design of robust digital preservation systems.

2.4 Arabic Text Normalization and Language-Aware Processing

Arabic digital text introduces additional complexity because visually similar strings may be represented through different Unicode sequences, character compositions, and diacritic patterns. The Unicode Consortium [17] formalized canonical normalization standards, including NFC, to reduce representational ambiguity and ensure consistent encoding of equivalent characters. In Arabic natural language processing, Habash [18], Farghaly and Shaalan [19], and Attia [20] highlighted the importance of normalization and diacritic handling for reliable Arabic text processing. More recently, transformer-based Arabic language models such as AraBERT [23] have further underscored the practical importance of robust preprocessing pipelines for Arabic linguistic data.

These works are highly relevant because they show that Arabic text processing requires explicit normalization policies to avoid inconsistent downstream behavior. However, they were not designed as cryptographic integrity frameworks and therefore do not provide hierarchical authenticity guarantees for Quran preservation. In other words, they solve part of the linguistic consistency problem, but not the full verification problem addressed in this paper.

2.5 Data-Centric Information Management Context

Although not directly focused on Quran verification, recent studies in large-scale information systems reinforce the importance of reliable data organization, traceable processing, and trustworthy analytical infrastructures. Mezzoudj *et al.* [24] examined the trade-offs between data lakes and data warehouses for big data analytics, highlighting the importance of choosing structured data management strategies appropriate to the characteristics of the digital assets being handled. Trautwein *et al.* [27] demonstrated the value of data-based analytical insight in a large-scale application domain, while Uysal [28] emphasized the increasing role of data-intensive and AI-supported workflows in policy and management contexts.

These studies are not direct predecessors of Quran authentication systems, but they support a broader observation: trustworthy digital infrastructures increasingly depend on structured, reproducible, and well-managed data pipelines. This observation strengthens the motivation for a formal verification architecture in which digital Quran content is processed under fixed normalization, hierarchical integrity, and certification rules.

2.6 Research Gap

Despite substantial progress across these research directions, existing solutions remain fragmented. Text-based Quran authentication methods focus on quotation correctness, rule-based validation, or linguistic consistency without providing cryptographic immutability [1,2,7–11]. Watermarking approaches protect image or document representations but do not address canonical text normalization or structured textual integrity [3–6, 15, 16]. General cryptographic preservation frameworks provide fixity and timestamping foundations but do not model Quranic linguistic hierarchy or Arabic-specific normalization [12–14, 21, 22]. Arabic NLP research addresses normalization and diacritic processing, yet does not integrate these mechanisms into a cryptographically bound Quran verification pipeline [17–20, 23].

Therefore, to the best of our knowledge, there is still no unified framework that simultaneously combines canonical Unicode normalization, hierarchical cryptographic hashing across multiple textual levels, digital certification, and optional visual authentication within a single trust-oriented architecture for the Digital Holy Quran. The proposed framework addresses this gap by integrating these previously disconnected dimensions into a coherent and extensible verification model.

Table 1 summarizes the main differences between existing approaches and the proposed framework.

Table 1: Comparative Analysis of Existing Digital Quran Verification Approaches

Work	**Text Verification**	**Image Authentication**	**Cryptographic Hashing**	**Canonical Normalization**	**Certification / Timestamping**
Shamsudin & Farooq (2000) [1]	Linguistic analysis	No	No	No	No
Alshareef & El Saddik (2012) [2]	Rule-based matching	No	No	No	No
Kurniawan *et al.* (2013) [4, 5]	No	Watermark-based tamper detection	Partial (embedded identifiers)	No	No
Al-Ahmad & Alshaikhli (2013) [6]	Limited (PDF-level)	Watermarking	Encryption-based protection	No	No
Proposed Framework	Hierarchical (Word/Ayah/Surah)	Fragile watermarking	Full SHA-256 multi-level binding	Unicode NFC enforcement	Digital signature + timestamp

3 Methodology

The proposed **Digital Quran Verification Framework** adopts a layered, cryptographically formalized, and modular architecture designed to ensure authenticity, structural integrity, and long-term trustworthiness of the Holy Quran in digital environments.We take a different approach from previous methods by integrating linguistic verification, cryptographic integrity and visual verification together, through one cohesive multi-level verification pipeline.

3.1 Dataset Description

The textual corpus used in this study is the Uthmani script text, a widely accepted reference representation of the Holy Quran comprising all 114 Surahs and their corresponding Ayahs in Arabic text form. This corpus serves as the canonical reference baseline for the proposed verification framework.

For experimental evaluation, the framework was validated on selected Quranic chapters using controlled orthographic and formatting variants derived from the reference corpus. The reference dataset itself is not altered; rather, it functions as the authoritative baseline against which submitted or modified digital Quran representations are verified. This design ensures that the framework evaluates integrity strictly against a fixed and trusted textual reference.

3.2 System Model and Notation

Let $\mathcal{Q}$ denote the complete Quran corpus, structured hierarchically as:

$$\mathcal{Q} = \{\mathcal{S}_i\}_{i=1}^{114}$$

where each Surah $\mathcal{S}_i$ consists of an ordered set of Ayahs:

$$\mathcal{S}_i = \{v_{i,j}\}_{j=1}^{m_i}$$

Each Ayah $v_{i,j}$ is further composed of an ordered sequence of words:

$$v_{i,j} = \{w_{i,j,k}\}_{k=1}^{n_{i,j}}$$

The verification transformation is defined as:

$$\mathcal{Q}^* = \Psi(\mathcal{Q}) = \{C(\mathcal{Q}), I(\mathcal{Q}), D(\mathcal{Q}), V(\mathcal{Q})\} \tag{1}$$

where:

- $C(\cdot)$ denotes canonical normalization,
- $I(\cdot)$ denotes hierarchical cryptographic integrity binding,
- $D(\cdot)$ denotes digital certification,
- $V(\cdot)$ denotes visual authentication.

3.3 Canonical Text Normalization Layer

Arabic digital text may be represented using multiple Unicode encodings due to diacritic composition, character decomposition, and formatting variations. To eliminate false mismatches caused by encoding inconsistencies, the framework enforces canonical normalization prior to cryptographic processing.NFC is preferred as it produces a composed form, which is the standard representation for stored text and reduces the number of code points, simplifying subsequent processing.

Let $\mathcal{N}_{\text{NFC}}(\cdot)$ denote Unicode Normalization Form C (NFC). For each word:

$$\tilde{w}_{i,j,k} = \mathcal{N}_{\text{NFC}}(w_{i,j,k}) \tag{2}$$

This ensures semantically identical Quranic words yield identical binary representations before hashing.

3.4 Hierarchical Cryptographic Integrity Binding

After canonical normalization, integrity is established using a deterministic hierarchical SHA-256 construction.

Let $H(\cdot)$ denote the SHA-256 hash function.

Word-Level Hashing:

$$h_{i,j,k} = H(\tilde{w}_{i,j,k}) \tag{3}$$

Ayah-Level Hashing:

$$A_{i,j} = H(h_{i,j,1} \| h_{i,j,2} \| \dots \| h_{i,j,n_{i,j}}) \tag{4}$$

Surah-Level Hashing:

$$S_i = H(A_{i,1} \| A_{i,2} \| \dots \| A_{i,m_i}) \tag{5}$$

This recursive construction guarantees:

- Sensitivity to any word-level modification,
- Preservation of structural ordering,
- Detection of insertion, deletion, or reordering attacks.

The final corpus-level integrity reference is defined as:

$$\mathcal{H}_{\mathcal{Q}} = H(S_1 \| S_2 \| \dots \| S_{114}) \tag{6}$$

3.5 Digital Certification and Trust Anchoring

To establish non-repudiation and long-term verifiability, the computed integrity reference $\mathcal{H}_{\mathcal{Q}}$ is bound to a digital certification record:

$$\langle \mathcal{H}_{\mathcal{Q}}, \sigma, T, \text{CertID} \rangle$$

where:

- σ is a digital signature generated using a trusted private key,
- T is a cryptographic timestamp,
- CertID uniquely identifies the certification record.

Verification is performed by validating the signature using publicly verifiable credentials and recomputing the hierarchical hash.

3.6 Visual Authentication Layer

For image-based representations, a fragile watermark W embedding the integrity reference is inserted into Mushaf images:

$$I^* = \Phi(I, W) \tag{7}$$

where I denotes the original image and Φ represents the watermark embedding function. Any pixel-level alteration invalidates the extracted watermark, ensuring visual tamper detection.

3.7 Threat Model

The framework assumes a polynomial-time adversary capable of:

- Modifying diacritics or orthographic markers,
- Re-encoding text using alternative Unicode forms,
- Inserting or deleting words or verses,
- Modifying image pixels,
- Attempting certificate forgery.

Security relies on the collision resistance and second-preimage resistance of SHA-256 and the unforgeability of digital signatures under standard cryptographic assumptions.

3.8 Integrated Verification Decision Rule

A submitted digital Quran $\mathcal{Q}'$ is considered authentic if and only if:

$$\mathcal{H}_{\mathcal{Q}'} = \mathcal{H}_{\mathcal{Q}} \wedge \text{Verify}(\sigma) = \text{true} \wedge V(\mathcal{Q}') = \text{valid} \tag{8}$$

This establishes a deterministic multi-layer trust decision model.

3.9 Methodological Contributions

The upgraded methodological contributions are:

- A formally defined hierarchical SHA-256 binding mechanism at word, ayah, surah, and corpus levels.
- Integration of Unicode canonical normalization into cryptographic integrity pipelines.
- Certification-based trust anchoring with digital signatures and timestamps.
- Unified text–image authentication within a single verification architecture.

3.10 Illustrative Example: Surah Al-Kawthar

To demonstrate the hierarchical construction, we consider Surah Al-Kawthar in the Fig 2, which contains three Ayahs:

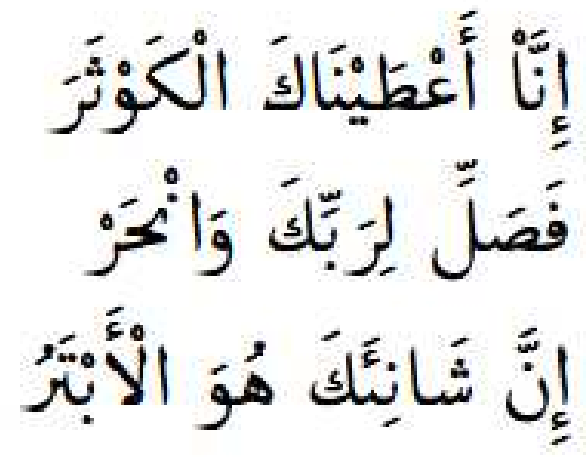


Figure 2: Baseline Uthmani representation of Surah Al-Kawthar used as an illustrative example in the hierarchical hashing process.

Each word $w_{i,j,k}$ is first normalized using NFC and then hashed using SHA-256:

$$h_{i,j,k} = H(\tilde{w}_{i,j,k})$$

The resulting word-level hashes are concatenated to compute the Ayah-level digest:

$$A_{i,j} = H(h_{i,j,1} \| \dots \| h_{i,j,n})$$

Finally, the Surah-level hash is computed as:

$$S_i = H(A_{i,1} \| A_{i,2} \| A_{i,3})$$

This hierarchical structure ensures that even minor orthographic variations such as diacritic differences or formatting symbols produce distinct Surah-level integrity references.

It is important to note that NFC normalization resolves encoding-level inconsistencies but does not remove orthographic elements such as tashkīl or hamza variations. Therefore, such character-level differences remain detectable by the hashing layer.

4 Experimental Results and Discussion

This section presents the hierarchical hashing analysis conducted on two orthographic variants of Surah Al-Kawthar. The objective is to evaluate the sensitivity of the proposed multi-level SHA-256 construction to textual variations, specifically differences in tashkīl (diacritical marks) and hamza representation.

4.1 Threat and Modification Categories Considered in Evaluation

The evaluation is organized around three controlled modification categories, each designed to test a specific property of the proposed verification framework. Let Q denote the reference Quranic text and Q' a modified version. The objective is to examine whether the framework correctly distinguishes between modifications that should change the integrity reference and perturbations that should be neutralized during preprocessing.

1) Orthographically meaningful modifications. This category includes character-level changes that affect the canonical textual form, such as removal of selected diacritical marks, alteration of hamza representation, or other orthographic substitutions. These operations preserve much of the visual appearance or semantic interpretation of the verse, but modify its binary representation. Formally, if a word $w_{i,j,k}$ is changed into $w'_{i,j,k}$ such that

$$w_{i,j,k} \neq w'_{i,j,k},$$

then the corresponding hash is expected to change:

$$H(w_{i,j,k}) \neq H(w'_{i,j,k}).$$

Because the framework uses hierarchical aggregation, the effect propagates to the ayah-level and surah-level digests. This category is used to evaluate the tamper sensitivity of the model.

2) Formatting-only perturbations. This category includes variations such as redundant whitespace, tatweel insertion, and similar display-level artifacts introduced during editing, rendering, or conversion. These perturbations do not alter the intended canonical content after normalization. Let $N(\cdot)$ denote the normalization function. For formatting-only variants, the expected behavior is

$$N(Q') = N(Q),$$

which implies that the resulting integrity references should remain unchanged. This category is used to verify normalization-aware stability and resistance to false mismatch caused by non-semantic formatting noise.

3) Encoding-level variation. Arabic digital text may contain visually identical strings represented using different Unicode compositions. Although these variants can appear equivalent at the rendering level, their raw binary encodings may differ. Direct hashing of the unnormalized input may therefore produce inconsistent digests. The framework addresses this issue by applying NFC canonicalization before hashing, so that Unicode-equivalent inputs satisfy

$$N_{\mathrm{NFC}}(Q') = N_{\mathrm{NFC}}(Q).$$

As a result, encoding-level variation should not affect the final integrity reference.

These three categories define the evaluation space used throughout the experiments. Orthographically meaningful modifications are expected to change the integrity hierarchy, whereas formatting-only and Unicode-equivalent encoding variations are expected to be neutralized by preprocessing. This categorization provides a structured basis for assessing both strict integrity enforcement and normalization-aware robustness..

4.2 Experimental Procedure

Both variants were processed using the hierarchical hashing model defined in Section III. The procedure consisted of three deterministic stages:

1. Each word was individually hashed using SHA-256.
2. Word-level hashes were concatenated in their original order and hashed again to produce the Ayah-level digest.
3. All Ayah-level hashes were concatenated and re-hashed to produce the final Surah-level integrity reference.

This recursive structure ensures that any character-level modification propagates deterministically to the final Surah-level hash.

4.3 Variant Description and Hash Results

Variant 1 uses a plain textual representation of the three verses as is shown in Fig 3:

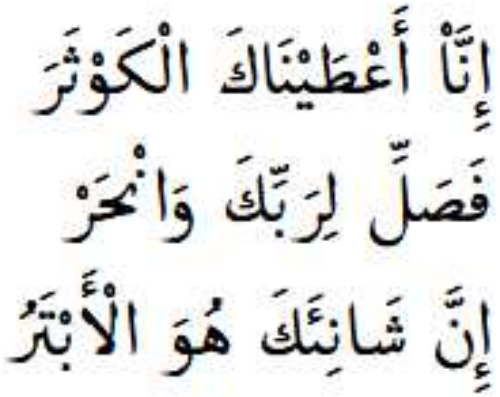


Figure 3: Orthographic variant of Surah Al-Kawthar obtained by modifying diacritical marks in selected words.

When processed by the prototype, this input produced the following Surah-level integrity reference:

`efa230669550978b1a64a6edbaa9cf98e20ccf3c1b2727b444d0bedf580602ae`

This value serves as the baseline verification output for an authentic digital representation.

Variant 2 uses the same verses but contains orthographic differences in hamza representation and tashkīl usage as is shown in the fig 4:

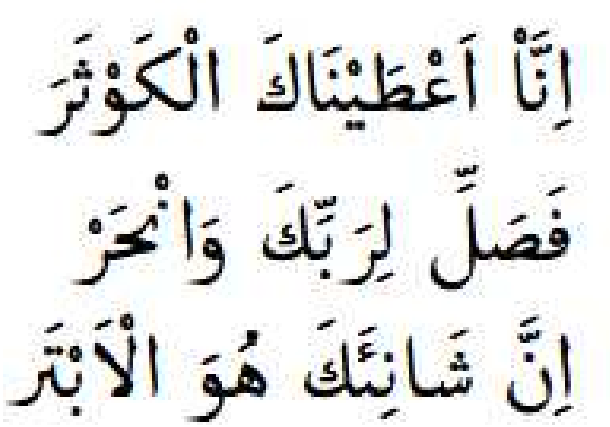


Figure 4: Formatting-based variant of Surah Al-Kawthar with non-semantic perturbations such as spacing or tatweel .

For this input, the prototype generated a different Surah-level integrity reference: `25d303203df4ff323e857e38e9ca8cb37e3a137b4eac57f916ff83b6b292af2`

Table 2 summarizes the hash outputs obtained for the two orthographic variants of Surah Al-Kawthar and highlights the ability of the proposed framework to detect even minor character-level modifications.

Table 2: Comparison of Orthographic Variants of Surah Al-Kawthar

Variant	Description	Surah-Level SHA-256 Hash (Truncated)	Modification Detected
Variant 1	Baseline Uthmani textual representation	efa23065078b1...a64a20ccf3c1b27...	No
Variant 2	Orthographic variation involving hamza representation and tashkīl usage	25d303df4ff3...a8cb37e3a37b4eac...	Yes

4.4 Orthographic Modification Propagation

Although the semantic meaning of the Surah remains unchanged, Variant 2 differs at the character level due to variations in hamza forms and diacritical marks. Since SHA-256 operates on the binary encoding of the input text, even a single-character difference alters the corresponding word-level hash:

$$w_{i,j,k} \neq w'_{i,j,k} \Rightarrow H(w_{i,j,k}) \neq H(w'_{i,j,k})$$

This difference propagates hierarchically:

$$h_{i,j,k} \rightarrow A_{i,j} \rightarrow S_i$$

As a result, the final Surah-level hash differs completely between the two variants.

4.5 Extended Validation Using Surah Al-Fatiha

To strengthen the experimental evaluation beyond the initial case study of Surah Al-Kawthar, the proposed framework was additionally validated using Surah Al-Fatiha. This surah was selected because it contains multiple ayahs with diverse lexical and orthographic structures, making it suitable for evaluating both sensitivity to meaningful textual modifications and robustness against irrelevant formatting perturbations. Three controlled versions were analyzed: (i) the original baseline text, (ii) an orthographically modified variant in which selected diacritical marks were removed from the first ayah, and (iii) a formatting-noise variant in which redundant whitespace and tatweel characters were introduced without changing the normalized content.

The baseline version of Surah Al-Fatiha used in the experiment is given below in Fig5:

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ
الرَّحْمَنِ الرَّحِيمِ
مَالِكِ يَوْمِ الدِّينِ
إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ
اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ
صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ

Figure 5: Baseline Uthmani representation of Surah Al-Fatiha used in the extended validation experiments.

For the baseline version of Surah Al-Fatiha, the hierarchical hashing pipeline produced stable ayah-level digests and the following surah-level integrity reference:

$$H_{\text{Fatiha}}^{(\text{orig})} = \texttt{8f8fcca633bef42602d1b265e832aaefa1cc5d3f1cc4ddc82f7e5c6275d84ed2} \tag{9}$$

The baseline ayah-level hashes were computed as follows:

$$A_1 = \texttt{75016b6e1a1a5a9553511b5877f1f623}\\ \texttt{9bcd7b73340f00b5221dbbf630e433ea} \tag{10}$$

$$A_2 = \texttt{70475cba06d612c9f98ec536706982bc}\\ \texttt{2145cdaf000c1194938d9dc1feba3b16} \tag{11}$$

$$A_3 = \texttt{0ab9205cd1f836bb116bc623d4676a1d}\\ \texttt{782e92c6c566219db237f110821b1863} \tag{12}$$

$$A_4 = \texttt{e332ec8e05db522cf1260f0045f956ac}\\ \texttt{6a50e8b473ddf4ffb38cbd9158a04aca} \tag{13}$$

$$A_5 = \texttt{74b064fa3f520b3907de259b7a270ea4}\\ \texttt{c938176271871b92f5aaf9f0f6a61238} \tag{14}$$

$$A_6 = \texttt{aa8f09e6f8755dc9efe50e4ce13c51bb}\\ \texttt{abc36e6a8644ecc781c76e4d40f4a1cf} \tag{15}$$

In Variant 2, only the first ayah was modified by removing selected diacritical marks from several words, while the remaining five ayahs were preserved unchanged. This produced a different first ayah hash:

$$A_1^{(\mathrm{v2})} = \texttt{f153592ce1b83418d048cbcbb6622ed1}\\ \texttt{a83baa7614cdb4eb7a25925456835602} \tag{16}$$

whereas the hashes of Ayahs 2–6 remained identical to the baseline. Despite this localized modification, the final surah-level digest changed completely:

$$H_{\mathrm{Fatiha}}^{(\mathrm{v2})} = \texttt{501bcb390905201f750461a0a622b433}\\ \texttt{1d55340594801860de276f18e39928d6} \tag{17}$$

This result confirms the hierarchical propagation property of the proposed framework: a character-level modification in a single ayah changes the corresponding word hashes, then the ayah digest, and finally the surah-level integrity reference. The result also demonstrates the strict orthographic sensitivity of the model, since removal of diacritical marks is preserved by the normalization policy and therefore remains detectable at the cryptographic level.

In Variant 3, the same baseline text was intentionally perturbed using formatting noise only, including redundant whitespace and tatweel characters. Since the preprocessing layer removes tatweel and collapses repeated whitespace during normalization, the normalized text for all six ayahs became identical to the baseline. Consequently, all ayah-level hashes remained unchanged and the final surah-level integrity reference exactly matched the original value:

$$H_{\mathrm{Fatiha}}^{(\mathrm{v3})} = \texttt{8f8fcca633bef42602d1b265e832aaef}\\ \texttt{a1cc5d3f1cc4ddc82f7e5c6275d84ed2} \tag{18}$$

This outcome is particularly important because it demonstrates that the framework is not only sensitive to meaningful orthographic modifications, but also robust against irrelevant

Table 3: Comparative Results for the Three Surah Al-Fatiha Variants

Variant	Description	Hash (Truncated)	Change
Original	Baseline text	8f8fcca6...	No
Variant 2	Diacritics removed	501bcb39...	Yes
Variant 3	Formatting noise	8f8fcca6...	No

formatting artifacts that do not alter the intended textual content after normalization. Therefore, the proposed method achieves a desirable balance between strict integrity enforcement and normalization-aware stability.

Table 3 summarizes the comparative results obtained for the three Surah Al-Fatiha variants.

Overall, the Al-Fatiha experiments provide a stronger and more complete validation of the proposed framework. Unlike the earlier single-case demonstration, these results show that the system can simultaneously detect semantically subtle but orthographically meaningful modifications and ignore superficial formatting perturbations that are neutralized during normalization. This extended evaluation therefore reinforces the practical reliability, determinism, and methodological soundness of the hierarchical verification model.

4.6 Additional Validation Using Surah An-Nasr and Surah Al-Masad

To further evaluate the consistency and generalizability of the proposed framework across multiple Quranic chapters, two additional validation experiments were conducted using Surah An-Nasr and Surah Al-Masad. These surahs were selected because they differ in structural length and lexical composition, thereby providing additional evidence that the proposed hierarchical verification mechanism is not limited to a single chapter or orthographic pattern. For each surah, three controlled versions were analyzed: (i) the original baseline text, (ii) an orthographically modified variant in which diacritical marks were removed from the first ayah, and (iii) a formatting-noise variant in which redundant whitespace and tatweel characters were introduced without altering the normalized content.

4.6.1 Validation Using Surah An-Nasr

The baseline version of Surah An-Nasr used in the experiment is given below in Fig 6:

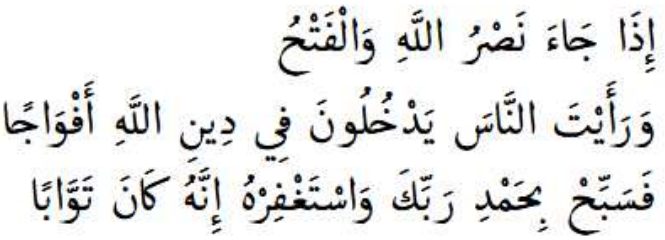


Figure 6: Baseline textual representation of Surah An-Nasr used to evaluate hierarchical propagation under controlled orthographic modification.

For the baseline version of Surah An-Nasr, the hierarchical hashing pipeline produced the following surah-level integrity reference:

$$H_{\text{Nasr}}^{(\text{orig})} = \texttt{b60c055f2c2c3e5370e4b0bb9222e93b} \\ \texttt{a78b37482ff00147887e6c2d420c1b08} \tag{19}$$

The corresponding ayah-level hashes were computed as follows:

$$A_1 = \texttt{80d1a830c5f844b7c832b8e3c1360e0d} \\ \texttt{81d9dced6d479b61a8a7eed39a2e7e55} \tag{20}$$

$$A_2 = \texttt{e91f509b2672eb38a17ba0ab28c32616} \\ \texttt{fb85d83dd035236f8c8209a86dcad4f2} \tag{21}$$

$$A_3 = \texttt{63426144c50c3d366beeec809c7c4337} \\ \texttt{5e74435b202dd9edbf9785778208f1a4} \tag{22}$$

In Variant 2, only the first ayah was modified by removing the diacritical marks, whereas Ayahs 2 and 3 were preserved unchanged. This orthographic modification produced a different first ayah hash:

$$A_1^{(\text{v2})} = \texttt{5706e878fb4612e01a5f55b65cd2c5d5} \\ \texttt{9930876a8c83585dff40a47f7497ed6b} \tag{23}$$

while the hashes of Ayahs 2 and 3 remained identical to the baseline. Despite the fact that the modification was limited to a single ayah, the final surah-level digest changed completely:

$$H_{\text{Nasr}}^{(\text{v2})} = \texttt{6bce56946233565b2cbc6fc5696c0640} \\ \texttt{916524f8b1461a7aa11577fbad7d363f} \tag{24}$$

In Variant 3, the same baseline text was perturbed using formatting noise only, including redundant whitespace and tatweel characters. Since the preprocessing layer removes tatweel and collapses repeated whitespace during normalization, the normalized text became identical to the baseline. Consequently, all ayah-level hashes remained unchanged and the final surah-level integrity reference exactly matched the original value:

$$H_{\text{Nasr}}^{(\text{v3})} = \texttt{b60c055f2c2c3e5370e4b0bb9222e93b} \\ \texttt{a78b37482ff00147887e6c2d420c1b08} \tag{25}$$

Table 4: Comparative Results for the Three Surah An-Nasr Variants

Variant	**Description**	**Surah-Level SHA-256 Hash (Truncated)**	**Change**
Original	Baseline Uthmani textual representation	`b60c055f2c2c3e53...`	No
Variant 2	Removal of diacritical marks in Ayah 1	`6bce56946233565b...`	Yes
Variant 3	Formatting perturbation using extra spaces and tatweel characters	`b60c055f2c2c3e53...`	No

These results confirm two complementary properties of the proposed framework. First, the system remains highly sensitive to orthographically meaningful changes, since removing diacritical marks from a single ayah alters the ayah digest and propagates to the final surah-level integrity reference. Second, the framework is robust to formatting-only perturbations that are neutralized during normalization. Therefore, the Surah An-Nasr experiment provides additional evidence for both the hierarchical propagation behavior and the normalization-aware stability of the proposed method.

Table 4 summarizes the comparative results for the three Surah An-Nasr variants.

4.6.2 Validation Using Surah Al-Masad

The baseline version of Surah Al-Masad used in the experiment is given below in Fig 7:

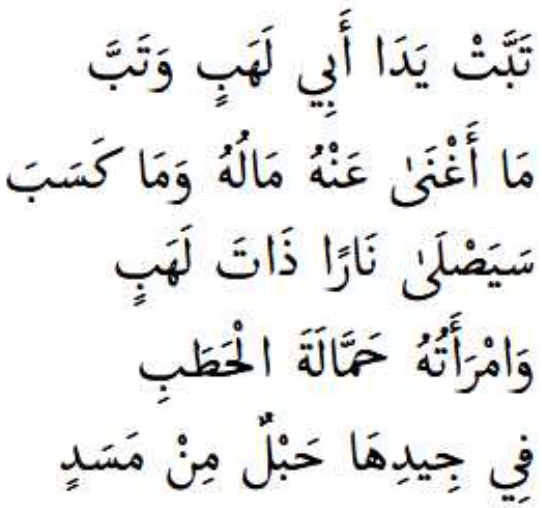


Figure 7: Baseline textual representation of Surah Al-Masad used for multi-ayah structural validation of the proposed framework.

For the baseline version of Surah Al-Masad, the hierarchical hashing pipeline produced

the following surah-level integrity reference:

$$H_{\text{Masad}}^{(\text{orig})} = \texttt{fa512cab12f421bb3f2f793871ff51da} \\ \texttt{1dfc1eea860152df743bfb08bec6be9d} \tag{26}$$

The corresponding ayah-level hashes were computed as follows:

$$A_1 = \texttt{cb637856e0035cacb26c44dd25acbee1} \\ \texttt{03b9178e6ebcefa32b27bb852a2afe10} \tag{27}$$

$$A_2 = \texttt{6273f503f3876328686c53577455dfe6} \\ \texttt{cbe4d8697cf7bcf1ba50daa4d023f3c3} \tag{28}$$

$$A_3 = \texttt{a48b4744b96043c4687c645c24338d49} \\ \texttt{6019f1f793995edf3bd01355ac2d2c63} \tag{29}$$

$$A_4 = \texttt{fce00309baef37c66882334077638861} \\ \texttt{efefd27f7afada151507a319ce97f2eb} \tag{30}$$

$$A_5 = \texttt{54a328d7aefc4e49590d7ea6623287a9} \\ \texttt{a3b3e297e575fd34754e1d21a660a7df} \tag{31}$$

In Variant 2, only the first ayah was modified by removing the diacritical marks, whereas Ayahs 2–5 were preserved unchanged. This produced a different first ayah hash:

$$A_1^{(\text{v2})} = \texttt{d2edac9c9e6fdfc1af7bb43f5ddd52e6} \\ \texttt{c471be5818945c7646b26fe1cf274a05} \tag{32}$$

while the hashes of Ayahs 2–5 remained identical to the baseline. As a result, the final surah-level integrity reference changed completely:

$$H_{\text{Masad}}^{(\text{v2})} = \texttt{c3e92c6eefde9271543fcf4623c0c844} \\ \texttt{5722b1d75c250e1c406356a94a809ab2} \tag{33}$$

In Variant 3, the same baseline text was perturbed using formatting noise only, including redundant whitespace and tatweel characters. Since the preprocessing layer removes tatweel and collapses repeated whitespace during normalization, the normalized text for all ayahs became identical to the baseline. Consequently, all ayah-level hashes remained unchanged and the final surah-level integrity reference exactly matched the original baseline value:

$$H_{\text{Masad}}^{(\text{v3})} = \texttt{fa512cab12f421bb3f2f793871ff51da} \\ \texttt{1dfc1eea860152df743bfb08bec6be9d} \tag{34}$$

The Surah Al-Masad experiment strengthens the empirical evidence in two ways. First, it confirms that orthographic changes remain detectable even in a structurally richer surah with five ayahs. Second, it demonstrates that the framework preserves stable integrity references under formatting-only perturbations after normalization. Together, these findings reinforce the reproducibility, robustness, and generalizability of the proposed hierarchical hashing framework across Quranic chapters of different sizes and structures.

Table 5: Comparative Results for the Three Surah Al-Masad Variants

Variant	**Description**	**Surah-Level SHA-256 Hash (Truncated)**	**Change**
Original	Baseline Uthmani textual representation	`fa512cab12f421bb...`	No
Variant 2	Removal of diacritical marks in Ayah 1	`c3e92c6eefde9271...`	Yes
Variant 3	Formatting perturbation using extra spaces and tatweel characters	`fa512cab12f421bb...`	No

Table 5 summarizes the comparative results for the three Surah Al-Masad variants.

Overall, the experiments on Surah An-Nasr and Surah Al-Masad substantially strengthen the evaluation of the proposed framework. In both surahs, orthographically meaningful modifications in a single ayah resulted in different surah-level integrity references, confirming the strict sensitivity and hierarchical propagation behavior of the SHA-256 construction. At the same time, formatting-only perturbations were successfully neutralized by the normalization layer, preserving identical final hashes. These results therefore demonstrate that the proposed verification framework achieves a desirable balance between tamper sensitivity and normalization-aware stability across multiple Quranic chapters.

4.7 Determinism and Reproducibility of Integrity References

Determinism is a fundamental requirement of the proposed verification framework. For a fixed canonical input, the system must generate the same integrity references at word, ayah, surah, and corpus levels across repeated executions. This property follows from three factors: a fixed normalization policy, ordered hierarchical aggregation, and the deterministic behavior of SHA-256.

Let Q denote a reference Quranic input and $N(\cdot)$ the normalization function. If the preprocessing rules remain unchanged, then normalization produces a unique canonical form:

$$N(Q) = \tilde{Q}.$$

Since each word digest is computed directly from the normalized text, the corresponding word-level hashes are reproducible. The same holds for higher levels of the hierarchy, because ayah-, surah-, and corpus-level references are obtained through fixed ordered concatenation followed by repeated application of the same hash function. No random initialization, probabilistic sampling, or heuristic decision rule is involved in the computation pipeline.

Formally, for any two executions of the framework over the same normalized input $\tilde{Q}$, the resulting integrity references satisfy

$$H^{(1)}(\tilde{Q}) = H^{(2)}(\tilde{Q}),$$

and similarly for all intermediate structural levels. Therefore, identical input conditions lead to identical cryptographic outputs.

This property has two direct implications. First, it enables stable certification and trust anchoring, since a single certified reference remains valid across repeated verification sessions. Second, it supports consistent deployment across platforms, provided that normalization rules and textual ordering are preserved. In this sense, the generated integrity references are reproducible verification artifacts rather than execution-specific outputs.

The experimental results are consistent with this behavior. Baseline inputs produced identical digests across repeated processing, orthographically meaningful modifications produced different integrity references, and formatting-only perturbations neutralized during normalization reproduced the original references exactly. These observations confirm that the framework satisfies the reproducibility requirement necessary for reliable long-term digital Quran verification.

4.8 Discussion

The experimental results confirm that the proposed hierarchical hashing construction exhibits two complementary and desirable properties: strict orthographic sensitivity and normalization aware robustness.

First, the framework is highly sensitive to orthographic variations, including differences in tashkīl and hamza representation. Even when textual variants remain semantically equivalent, their underlying binary encodings differ, resulting in distinct 256-bit integrity references. This behavior is consistently observed across multiple Quranic chapters, where localized modifications at the word or ayah level propagate deterministically to the final surah-level hash. The avalanche property of SHA-256 ensures that even minimal character-level changes produce completely different output digests, thereby guaranteeing reliable detection of textual alterations under standard collision-resistance assumptions (2^{-256}). Second, the framework demonstrates robustness against normalization-neutral formatting perturbations. Variations such as redundant whitespace, tatweel characters, and other non-semantic formatting differences are effectively eliminated during the canonical normalization stage. As a result, such perturbations do not affect the normalized representation, and identical integrity references are preserved. This property prevents false positives in integrity verification while maintaining strict enforcement of meaningful textual consistency. Together, these findings indicate that the proposed model achieves a balanced verification behavior: it detects semantically relevant orthographic modifications while remaining stable under irrelevant formatting variations. In addition, there is a consistent reinforcement of the integrity of the digital Quranic text through all levels of the hierarchal database (word, ayah, and surah). This ensures that there is both a scalable and reliable approach to further establishing the authenticity and credibility of the digital Quran text.

5 Security Analysis

The security of the proposed Digital Quran Verification Framework relies on the cryptographic strength of the SHA-256 hash function and the hierarchical construction used to propagate integrity guarantees across structural levels.

5.1 Collision Resistance

SHA-256 provides a 256-bit output space, offering theoretical collision resistance of approximately 2^{-128} under the birthday bound and preimage resistance of 2^{-256}. This ensures that generating two distinct Quranic texts that produce identical Surah-level integrity references is computationally infeasible under standard cryptographic assumptions.

5.2 Avalanche Effect and Orthographic Sensitivity

The avalanche property of SHA-256 ensures that minimal character-level modifications—such as changes in tashkīl or hamza representation—produce completely different output digests. This property guarantees reliable detection of orthographic-level tampering.

Formally, for two distinct inputs $Q \neq Q'$:

$$H(Q) \neq H(Q')$$

with overwhelming probability.

5.3 Hierarchical Integrity Propagation

The recursive construction strengthens integrity enforcement by ensuring that modifications at the word level propagate to ayah-level and surah-level hashes:

$$w_{i,j,k} \rightarrow h_{i,j,k} \rightarrow A_{i,j} \rightarrow S_i$$

Thus, tampering cannot remain localized without affecting higher structural references.

5.4 Trust Anchoring and Certification

The use of digital signatures and timestamps creates verifiable trust anchors for integrity references, therefore providing non-repudiation, traceability, and lasting digital preservation in the framework.

5.5 Trust Assumptions and System Boundaries

The security guarantees of the proposed framework hold under a defined trust model. The reference corpus that is used for initialization must First be accurate and valid and be from a reliable source. All integrity references will be generated relative to that reference corpus, and any inconsistencies or non-canonical forms in the reference corpus will be retained by the system in the authoritative state.

Second, the certification infrastructure must remain trustworthy. The framework assumes secure management of the private signing key and verifiable distribution of the corresponding public credentials. If the certification environment is compromised, fraudulent integrity records may be issued even though the hashing procedure itself remains deterministic and correct.

Third, the integrity repository is assumed to preserve stored references without unauthorized modification. While digital signatures help detect forged or substituted records,

repository compromise may still affect availability, consistency, or version control. For this reason, repository protection, controlled access, and auditability remain necessary for operational deployment.

Under these assumptions, the framework provides deterministic orthographic integrity verification and multi-level tamper detection. However, its scope is limited to checking whether a submitted representation matches the certified canonical reference after normalization. It does not resolve theological interpretation, editorial variation, or semantic equivalence across orthographic traditions, and it does not by itself prevent unauthorized copying or redistribution of digital Quran content.

6 Limitations

While the proposed framework provides a rigorous and deterministic approach for orthographic integrity verification, several limitations should still be acknowledged.

First, the current implementation enforces exact character-level matching after normalization. As a result, variations in diacritical marks or alternative hamza representations—even when semantically equivalent in some interpretive contexts—are treated as distinct textual states. This design is appropriate for strict integrity enforcement, but it may be overly restrictive for applications that require semantic equivalence rather than exact orthographic preservation.

Second, although the experimental evaluation was expanded to include multiple Quranic chapters and controlled textual variants, the current study still does not constitute full corpus-wide validation across all 114 Surahs. Therefore, while the results provide stronger evidence of reproducibility, tamper sensitivity, and normalization robustness across different chapter structures, large-scale validation remains necessary to fully assess corpus-level scalability and operational performance.

Third,The framework makes assumptions about the trustworthiness and accuracy of the starting reference data set. If there are errors in the baseline corpus, such as inconsistencies, coding anomalies, and different versions of editorial items, then these representations will still be treated as trustworthy and authoritative integrity references. Therefore, the reliability of the framework relies upon the degree to which the reference source used to initialize it was of high quality.

Finally, while cryptographic integrity verification can reliably allow for verification of changes, it will not by itself prevent unauthorized distribution (copying and/or unauthorized use) of digital copies of the Quran. To address these problems, additional system level controls should be used, such as implementing information access policies; utilizing trusted repositories; and developing secure distribution architectures.

7 Future Work

Future research may extend the framework in several important directions.

First, a full corpus-wide validation across all 114 Surahs should be conducted to evaluate scalability, computational efficiency, and storage requirements under larger deployment set-

tings. Such an extension would also support corpus-level integrity analysis and strengthen the empirical basis of the framework.

Second, integration with blockchain-based anchoring mechanisms may be explored to provide decentralized and tamper-evident timestamping of integrity references. Recent studies suggest that blockchain infrastructures can strengthen archival authenticity, traceability, and trust preservation in long-term digital record management [21, 22].

Third, configurable Arabic NLP pipelines may be incorporated to support optional semantic-aware verification modes in addition to strict orthographic verification. This would allow the framework to distinguish between meaningful textual tampering and accepted orthographic variation under controlled linguistic policies. Advances in Arabic language modeling and transformer-based architectures provide promising directions for this extension [23].

Fourth, future implementations may support automated API-based verification services for integration into distributed Quran platforms, mobile applications, digital archives, and institutional preservation systems. Such services could enable real-time integrity checking and continuous trust validation in operational environments.

Finally, the framework may be generalized beyond Quranic preservation toward other culturally sensitive and linguistically complex digital texts that require deterministic authenticity verification, structured integrity binding, and long-term traceability.

8 Conclusion

This paper presented a layered and hierarchical Digital Quran Verification Framework that integrates canonical normalization, multi-level SHA-256 hashing, digital certification, and trust anchoring mechanisms to enforce strict orthographic integrity in digital Quran representations.

Experimental evaluation using multiple Quranic chapters and controlled textual variants demonstrated two complementary properties of the proposed framework. First, meaningful orthographic modifications, such as the removal of diacritical marks or character-level variation, produced distinct 256-bit integrity references, confirming the strict tamper sensitivity of the hierarchical hashing construction. Second, formatting-only perturbations, such as redundant whitespace and tatweel, were neutralized by the normalization layer and therefore preserved identical final hashes. Together, these results confirm the deterministic reproducibility, hierarchical propagation behavior, and normalization-aware stability of the proposed model.

By combining linguistic consistency enforcement with cryptographic integrity binding and structural propagation across word, ayah, and surah levels, the framework provides a structured and scalable approach for detecting textual modification in digital Quran content. The results also demonstrate broader implications for digital preservation systems. In particular, the hierarchical hashing construction enables fine-grained integrity verification by ensuring that even minimal localized changes propagate to higher structural levels, while canonical Unicode normalization reduces false mismatches caused by encoding-related inconsistencies in Arabic digital text processing.

Consequently, the proposed framework offers a practical and cryptographically grounded

solution for safeguarding the authenticity of the Digital Holy Quran. At the same time, it provides a generalizable verification architecture for protecting other culturally sensitive digital texts that require deterministic preservation, tamper detection, and long-term trust validation.

References

[1] A. F. Shamsudin and A. Farooq, "AI Natural Language in Meta-Synthetics of Al-Qur'an," in *Proc. IEEE TENCON 2000*, Kuala Lumpur, Malaysia, 2000, pp. 464–467.

[2] A. Alshareef and A. El Saddik, "A Quranic Quote Verification Algorithm for Verses Authentication," in *Proc. 2012 Int. Conf. on Innovations in Information Technology (IIT)*, Al Ain, UAE, 2012, pp. 339–343.

[3] M. S. Khalil, F. Kurniawan, M. K. Khan, and Y. M. Alginahi, "Two-Layer Fragile Watermarking Method Secured with Chaotic Map for Authentication of Digital Holy Qur'an," *The Scientific World Journal*, 2014, Article ID 803983.

[4] F. Kurniawan, M. S. Khalil, M. K. Khan, and Y. M. Alginahi, "Authentication and Tamper Detection of Digital Holy Quran Images," in *Proc. IEEE Int. Symp. on Biometrics and Security Technologies (ISBAST)*, Chengdu, China, 2013, pp. 291–296.

[5] F. Kurniawan, M. S. Khalil, M. K. Khan, and Y. M. Alginahi, "Exploiting Digital Watermarking to Preserve Integrity of the Digital Holy Quran Images," in *Taibah Univ. Int. Conf. on Advances in Information Technology for the Holy Quran and Its Sciences*, Madinah, Saudi Arabia, 2013.

[6] M. A. Al-Ahmad and I. F. Alshaikhli, "A New Fragile Digital Watermarking Technique for a PDF Digital Holy Quran," *Proceeding*, 2013.

[7] Y. M. Alginahi, O. Tayan, M. K. Kabir, *et al.*, "Verification of Qur'anic Quotations Embedded in Online Arabic and Islamic Websites," *Int. J. Islamic Applications in Computer Science and Technology*, vol. 1, no. 2, pp. 10–16, 2013.

[8] A. H. Al-Khalaf and A. L. Nayfeh, *Automatic Integrity Checking of Quran Script*, U.S. Patent 2011/0143325 A1, Jun. 16, 2011.

[9] T. M. El-Sakka, "Real-Time Interactive Verification of Quran Words in the Web Contents," in *Proc. Taibah Univ. Int. Conf. on Advances in Information Technology for the Holy Quran and Its Sciences*, Madinah, Saudi Arabia, 2013.

[10] A. Kamsin, A. Gani, I. Suliman, *et al.*, "Developing the Novel Quran and Hadith Authentication System," in *Proc. 5th Int. Conf. on Information and Communication Technology for the Muslim World (ICT4M)*, Kuching, Malaysia, 2014.

[11] T. Sabbah and A. Selamat, "A Framework for Quranic Verses Authenticity Detection in Online Forum," in *Proc. Taibah Univ. Int. Conf. on Advances in Information Technology for the Holy Quran and Its Sciences*, Madinah, Saudi Arabia, 2013.

[12] R. L. Rivest, "The MD5 Message-Digest Algorithm," IETF RFC 1321, Apr. 1992.

[13] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, 1991.

[14] P. De Stefano, C. Fleischhauer, A. Goethals, *et al.*, "What is Fixity, and When Should I Be Checking It? – NDSA Guidelines," National Digital Stewardship Alliance, 2014.

[15] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Francisco, CA, USA: Morgan Kaufmann, 2002.

[16] S. Hetzl and P. Mutzel, "A Graph-Theoretic Approach to Steganography," in *Proc. 9th Int. Conf. on Communications and Multimedia Security (CMS)*, LNCS 3677, Springer, 2005, pp. 119–128.

[17] Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms," Version 13.0.0, Mar. 2020.

[18] N. Habash, *Introduction to Arabic Natural Language Processing*. Morgan & Claypool, 2010.

[19] A. Farghaly and K. Shaalan, "Arabic Natural Language Processing: Challenges and Solutions," *ACM Trans. Asian Lang. Information Processing*, vol. 8, no. 4, Article 14, 2009.

[20] M. Attia, "Handling Arabic Diacritics in Text Normalization and Information Retrieval," *J. Computational Linguistics Research*, vol. 3, no. 2, pp. 11–20, 2011.

[21] F. Casino, T. K. Dasaklis, and C. Patsakis, "A Systematic Literature Review of Blockchain-Based Applications: Current Status, Classification and Open Issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2021.

[22] T. Hewa, Y. Hu, M. Liyanage, S. Kanhere, and M. Ylianttila, "Survey on Blockchain-Based Smart Contracts: Applications, Opportunities and Challenges," *IEEE Access*, vol. 9, pp. 87643–87672, 2021.

[23] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-Based Model for Arabic Language Understanding," *Proceedings of LREC 2020*, pp. 7089–7095, 2020.

[24] S. Mezzoudj, M. Khelifa, and Y. Saadna, "Data lakes versus data warehouses: choosing the right approach for big data analytics," *Journal of Electrical Systems and Information Technology*, vol. 12, p. 89, 2025, doi: 10.1186/s43067-025-00275-0.

[25] W. Zita, "Secure QR Code Login System with Role-Based Access for Web Applications," *TechRxiv*, vol. 2025, no. 0826, 2025. [Online]. Available: `https://doi.org/10.36227/techrxiv.175616741.16408346/v1`

[26] W. Zita and N. Zita, "Mitigating Supply Chain Risks in Communication Devices Across Conflict Zones," *TechRxiv Preprint*, 2025. doi: 10.36227/techrxiv.175234514.44631243/v1.

[27] I. Trautwein, D. Ravlija, and M. Sonntag, "Data-based insights into the usage of micromobility sharing," *Journal of Electrical Systems and Information Technology*, vol. 12, p. 62, 2025, doi: 10.1186/s43067-025-00251-8.

[28] B. Uysal, "Artificial intelligence and big data in health policy and management: a comprehensive bibliometric analysis," *Journal of Electrical Systems and Information Technology*, vol. 12, p. 93, 2025, doi: 10.1186/s43067-025-00286-x.