ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Technical Report

(06/2026)

XSTR.verm

Framework for verification of messages and content

Technical Report ITU-T XSTR.verm

Framework for verification of messages and content

Summary

This Technical Report ITU-T XSTR.verm provides a framework for creating messages and content verification for a range of device capabilities and protocols. Types of verification will cover technologies such as SMS, web, and content sent on social media or other Internet-based platforms. The framework includes a method considering source nodes, intermediate nodes (that may carry messages or content) and destination or receiver nodes, along with a set of verification nodes that can serve as endpoints that may be contacted by any intermediate or destination node for partial or complete verification.

Keywords

Internet, malicious content, media, message, metadata, misinformation, phishing, SMS, social media, spam, verification.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

© ITU 2026

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
3.1	Terms defined elsewhere	1
3.2	Terms defined in this Technical Report	1
4	Abbreviations and acronyms	2
5	Background	2
5.1	Content provenance frameworks	3
5.2	Remote attestation frameworks	3
5.3	Distributed ledger approaches	3
5.4	Messaging security frameworks	3
6	Architectural framework	3
7	Verification node and API	5
7.1	Role of the verification node	5
7.2	Minimal verification node API	5
8	Verification process	6
9	Optional DLT integration	6
10	Threat modelling and security considerations	6
10.1	Threat actors	8
10.2	Threats in the message delivery path	8
10.3	Threats in the verification process	9
10.4	Privacy considerations	9
10.5	Security considerations	9
11	Verification procedure	11
12	Illustrative implementation of the performance observation	11
12.1	Architecture and deployment	11
12.2	Observations and performance	12
13	Benefits and use cases	12
14	Conclusion	13
Appendix I – Case studies – Government SMS deployment (Ghana)		14
I.1	Problem context	14
I.2	Deployment scenario	14
I.3	Verification process	14
I.4	Impact and benefits	14
I.5	Stakeholders	15
I.6	Deployment constraints	15
Bibliography		16

Introduction

The Internet and its diverse communication modes including short message service (SMS), multimedia messaging service (MMS), social media platforms and web-based services have become essential infrastructure for e-commerce, banking, healthcare, government services, and democratic participation. These communication channels are relied upon for authentication workflows, financial notifications, emergency alerts, and public information dissemination. However, users are increasingly exposed to the content of varying reliability, ranging from fully accurate to partially misleading or deliberately fabricated. This challenge is particularly significant in regions where mobile messaging serves as the primary communication channel and where low distribution costs, combined with advances in generative artificial intelligence, amplify both legitimate and malicious content at scale.

Key security issues emerging from this landscape include phishing, involving the spoofing of message origins, impersonation of trusted institutions and misinformation, involving the fabrication or modification of messages to appear authentic. In addition, many communication platforms lack standardized mechanisms for effective retraction, update, or revocation of previously distributed content. The resulting impacts may range from consumer fraud and reputational harm to broader societal consequences affecting public trust and collective decision-making.

Several approaches currently seek to address these challenges. These include filtering and blocking mechanisms, do-not-contact registries, cryptographic message signing, reporting frameworks, and content provenance systems such as those developed by [b-C2PA]. System-level attestation mechanisms, including remote attestation procedures defined by [b-RFC 9334], aim to establish trust in device and platform integrity. Distributed ledger technology (DLT)-based solutions have also been explored to provide tamper-evident recording of attestations. While these approaches provide important capabilities, many assume capable end-user devices, embedded metadata support, platform-specific integration, or direct interaction with cryptographic or ledger infrastructure. Such assumptions may not hold in heterogeneous environments, particularly where communication relies heavily on SMS or other lightweight channels and where end-user devices may have limited computational capability.

This Technical Report introduces a general framework for message and content verification designed to address these gaps. The framework is centred on verification nodes, which are queryable endpoints through which individuals or automated systems may submit all or part of a message, or its associated metadata, in order to obtain an attestation response. By abstracting verification logic from the end-user device and enabling query-based validation across channels, the framework supports cross-channel verification (for example, verifying an SMS message via a web interface, or verifying content received online via SMS). In this model, advanced or cryptographic verification may be performed by the verification node on behalf of the user, thereby removing the requirement for complex processing capabilities on the end device.

The framework further supports optional integration with DLT mechanisms, including ethereum virtual machine (EVM)-compatible smart contracts, to enable tamper-evident anchoring and structured revocation of attestations. In this architecture, interaction with blockchain infrastructure remains abstracted behind the verification node, preserving a simplified user experience while enabling transparency and immutability where appropriate.

Technical Report ITU-T XSTR.verm

Framework for verification of messages and content

1 Scope

This Technical Report specifies a framework for verification services associated with communications technologies such as short message service (SMS), multimedia messaging service (MMS), mobile instant messaging, web services and Internet-based content distribution platforms. The framework enables message recipients, processing points, relays, or automated systems to check the validity of a message, its content, or available metadata by querying a verification service provided, authorized, or designated by the message sender or another accountable entity. The framework supports verification across different access channels, including SMS, MMS and Internet-based interfaces.

This Technical Report does not specify a centralized global verification service, a federation model for verification authorities, or mandatory use of distributed ledger technology. It also does not define protocol-specific message formats for each communication channel. These topics may be considered in future work.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following terms defined elsewhere:

3.1.1 content [b-ITU-T J.380.2]: The video, audio, and data streams taken together as a single identifiable unit. Content may refer to the original entertainment (programming) content, an ad spot, an interactive or enhanced application asset, or any other similar asset.

3.1.2 message [b-ITU-T J.380.2]: The unit of communication between two logical services.

3.1.3 metadata [b-ITU-T X.1255]: Structured information that pertains to the identity of users, systems, services, processes, resources, information or other entities.

3.1.4 revocation [b-ITU-T X.1252]: The annulment of something previously done by someone having the authority.

3.1.5 verification [b-ITU-T X.1285]: Process intended to test or prove the truth or accuracy of a fact or value.

3.2 Terms defined in this Technical Report

This Technical Report defines the following terms:

3.2.1 anchoring: The recording of a content hash, attestation reference, or related verification data in a tamper-evident registry, optionally implemented using distributed ledger technology (DLT).

3.2.2 attestation: A structured statement issued by an authorized entity asserting properties of a message or content, such as authenticity, integrity, or origin.

3.2.3 capable device: An end-user device that supports advanced cryptographic operations, embedded credential processing, or direct interaction with distributed ledger infrastructure.

3.2.4 content binding: A cryptographic or logical association between a message and a verification reference, such as a content hash or ledger identifier.

3.2.5 cross-channel verification: Verification of content using a channel different from the one through which the content was originally received.

3.2.6 simple device: An end-user device with limited computational capability, typically lacking support for complex cryptographic processing or embedded credential validation.

3.2.7 threat model: A formal description of the adversarial assumptions, attack vectors, and security boundaries relevant to the verification framework.

3.2.8 verification node: A queryable endpoint that receives content or content references and returns a verification result indicating whether the content corresponds to a valid attestation.

3.2.9 verification query: A request submitted to a verification node containing all or part of a message, a content reference, or associated metadata for the purpose of verification.

3.2.10 verification response: A structured reply from a verification node indicating a verification result and, optionally, supporting evidence or metadata.

3.2.11 verification result: An outcome returned by a verification node indicating whether submitted content or metadata is verified, not verified, or unknown.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

API	Application Programming Interface
C2PA	Coalition for Content Provenance and Authenticity
DLT	Distributed Ledger Technology
EVM	Ethereum Virtual Machine
HTTPS	Hypertext Transfer Protocol Secure
MIM	Mobile Instant Messaging
MMS	Multimedia Messaging Service
PoC	Proof of Concept
RATS	Remote Attestation Procedures
REST	Representational State Transfer
SMS	Short Message Service
URI	Uniform Resource Identifier

5 Background

Several existing approaches address aspects of content authenticity and message verification.

5.1 Content provenance frameworks

Content provenance initiatives such as the coalition for content provenance and authenticity (C2PA) and project origin embed cryptographic credentials directly within digital media or content objects. These credentials enable verification of the origin and modification history of digital assets.

While effective for media files, these approaches rely on support within the content itself and typically require compatible client applications to interpret the embedded credentials.

5.2 Remote attestation frameworks

The [b-RFC 9334] architecture provides mechanisms for verifying the integrity and trustworthiness of computing platforms. In this model, a verifier evaluates attestation evidence produced by a device.

Although remote attestation procedures (RATS) support strong cryptographic guarantees regarding device integrity, it is primarily designed for system integrity verification, rather than for validating the authenticity of messages transmitted across communication channels.

5.3 Distributed ledger approaches

Various systems use distributed ledger technologies (DLT) to record and verify the provenance of digital content. These systems provide tamper-resistant records of content creation and modification.

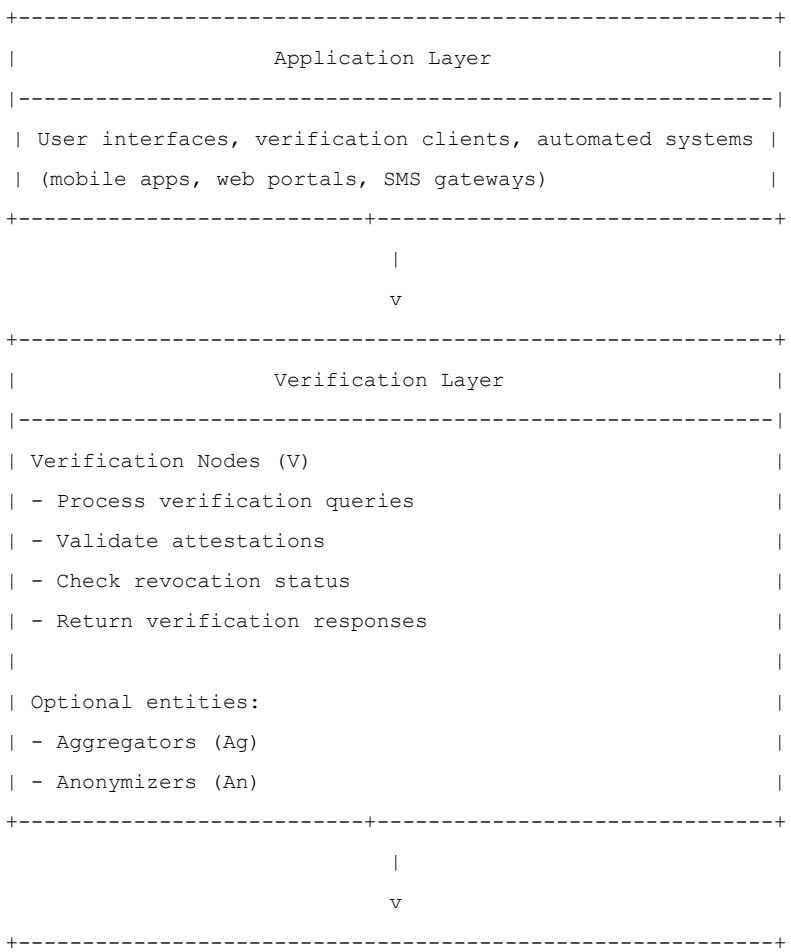
However, DLT-based systems often assume that the presence of blockchain-aware clients require interaction with ledger infrastructures, which may limit deployment in constrained environments.

5.4 Messaging security frameworks

Standards and industry frameworks addressing messaging security include anti-spam systems and fraud mitigation strategies for mobile messaging networks. While these mechanisms improve network-level security, they do not provide end-user verification of message authenticity.

6 Architectural framework

The architectural framework adopts a layered architecture to support device-agnostic message verification across multiple communication channels. The framework separates user interaction, verification processing, optional attestation anchoring, and message delivery into distinct layers to ensure modularity and interoperability across different platforms, see Figure 1.



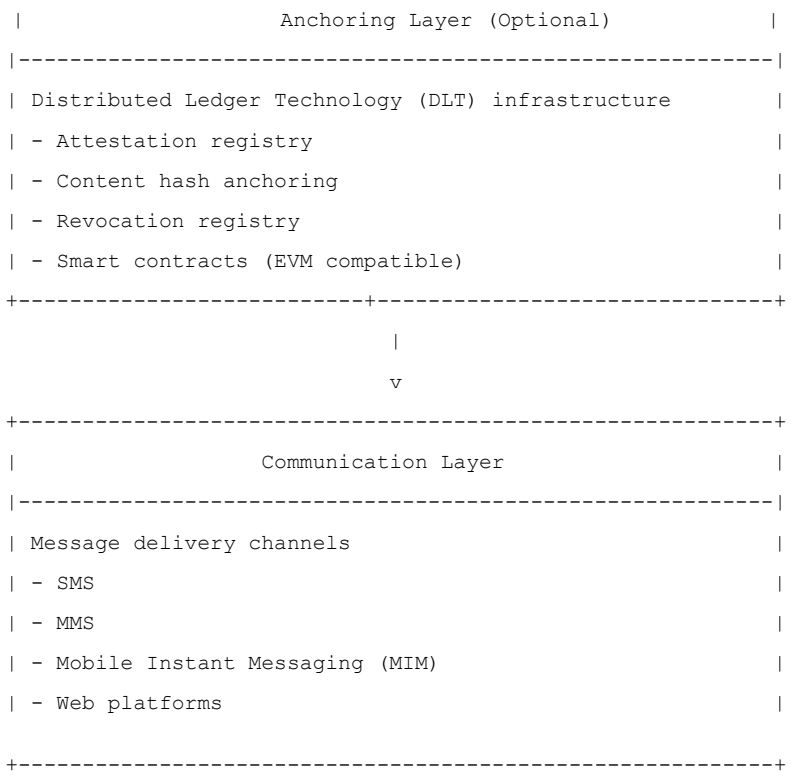


Figure 1 – Layered architecture of message verification framework

The application layer provides the user-facing interfaces through which verification requests are initiated. This layer includes verification clients and automated systems such as mobile applications, web portals, and SMS gateways. These interfaces allow users or systems to submit message content, excerpts, or hashes for verification without requiring cryptographic capabilities on the end device.

The verification layer constitutes the core functional component of the framework. It contains verification nodes (V) responsible for processing verification queries, validating message attestations, checking revocation status, and returning verification responses to requesters. The verification layer may also include optional supporting entities such as aggregators (Ag), which route queries to the appropriate verification nodes, and anonymizers (An), which protect user privacy by removing identifying information from verification requests.

The anchoring layer provides optional tamper-evident storage for message attestations. This layer may utilize distributed ledger technology (DLT) infrastructure to record message hashes, maintain attestation registries, and support revocation mechanisms. Smart contracts compatible with the ethereum virtual machine (EVM) may be used to anchor message hashes and maintain verification records while allowing independent verification of attested content.

The communication layer represents the message delivery infrastructure through which content is distributed to recipients. This layer includes communication channels such as short message service (SMS), multimedia messaging service (MMS), mobile instant messaging (MIM) platforms, and web-based communication services. Messages received through these channels may subsequently be submitted to the verification layer through the application layer for authenticity verification.

Table 1 – Functional roles per architectural layer

Layer	Role
Application layer	User interfaces and verification clients
Verification layer	Verification nodes and verification logic

Table 1 – Functional roles per architectural layer

Layer	Role
Anchoring layer (*optional*)	Blockchain/DLT infrastructure
Communication layer	Message delivery channels

By separating these functional components into layers, the framework enables scalable deployment across heterogeneous devices and communication environments while supporting optional blockchain-based attestation mechanisms.

7 Verification node and API

7.1 Role of the verification node

The verification node is the central operational component of the framework. Its primary purpose is to provide a queryable service that attests to the authenticity or provenance of messages and digital content across different communication channels and devices. In environments where phishing, spoofing, and misinformation are prevalent, the verification node acts as a trusted intermediary that users or systems can query to determine whether a received message corresponds to a legitimate and previously attested record.

A key design objective of the verification node is the device-agnostic verification. Instead of requiring end-user devices to perform cryptographic validation or interpret embedded credentials, the verification logic is handled by the node itself. This design allows verification services to be accessible to a broad range of devices, including simple feature phones, smartphones, and personal computers. As a result, users can verify messages without needing specialized software or cryptographic capabilities.

The verification node also enables cross-channel verification, allowing content received through one communication channel to be verified through another. For example, an SMS message may be verified through a web portal, or by forwarding the message to an SMS gateway connected to the verification node.

7.2 Minimal verification node API

To support interaction between requesters and the verification infrastructure, the framework defines a minimal verification-node application programming interface (API). The API is designed to be simple, interoperable, and accessible to both advanced systems and lightweight clients such as SMS gateways or web forms.

Table 2 – Primary endpoints of API

Endpoint	Purpose
POST /verify	Accepts content references (e.g., message text, hashes, or short codes) and returns a verification result
POST /anchor	Allows authorized entities to submit content hashes for registration or anchoring, optionally using a distributed ledger technology (DLT) backend
GET /health	Provides the operational status of the verification node

This minimal interface allows the verification node to be implemented using either a local storage system or an optional DLT backend while maintaining a consistent interface for requesters.

8 Verification process

When a user or system receives a message and wishes to verify it, a query is sent to the verification node through the verify endpoint. The request may include the full message content, a cryptographic hash, or a short identifier associated with the message, along with optional contextual information such as sender identity or timestamp.

Upon receiving the request, the verification node checks its local store for a matching attestation record. If a record is found, the node evaluates the associated metadata such as issuer identity and potential revocation status. If the content is not found locally and a DLT backend is enabled, the node may query the blockchain to determine whether an attestation exists there.

The node then returns a verification response indicating whether the message is verified, not verified, or unknown, and may include supporting information such as issuer identity, timestamp, and optional anchoring evidence.

9 Optional DLT integration

The framework allows the verification nodes to optionally integrate with DLT, including EVM-compatible blockchains. In such deployments, message hashes or attestations may be anchored on-chain to provide tamper-evident records and independent verification.

The verification node remains the primary interface for requesters and performs all blockchain interactions on their behalf. This ensures that end users do not need to interact directly with the blockchain or manage digital wallets. For high-volume deployments, techniques such as batch anchoring can be used to aggregate multiple message hashes into a single blockchain transaction, reducing operational costs while preserving verifiability.

The verification node and its minimal API form the foundation of the framework. By centralizing the verification logic in a queryable service, the framework enables the authentication of messages across diverse communication channels and device types. This architecture supports both simple deployments using local storage and higher-assurance deployments using tamper-evident blockchain anchoring, providing a scalable and accessible approach to the message and the content verification.

10 Threat modelling and security considerations

The framework operates in environments where messages may be modified, spoofed, or manipulated during transmission across communication channels such as SMS, web platforms, and mobile messaging services. A threat model is therefore required to identify potential adversarial actions and to describe the security mechanisms used by the framework to mitigate such threats.

In this framework's model, an adversary may attempt to interfere with the message delivery path or the verification process in order to deceive recipients or disrupt verification services. Figure 2 illustrates the threat modelling scenario.



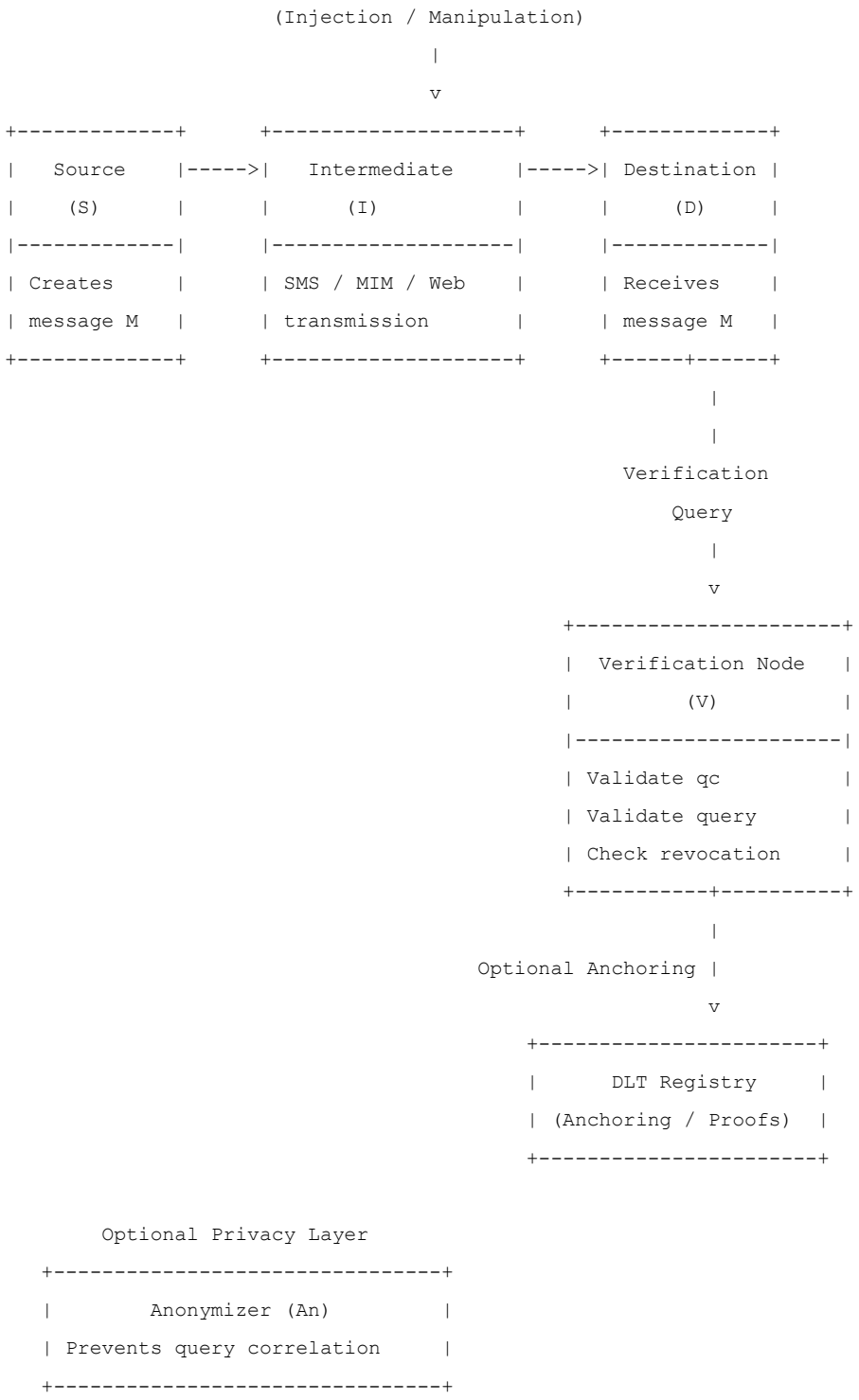


Figure 2 – Threat modelling scenario

In this verification framework, a source (S) originates a message (M), which is transmitted through one or more intermediates (I) using communication channels such as SMS, mobile instant messaging (MIM), or web platforms. The message is then received by the destination (D). After receiving the message, the destination may issue a verification query to a verification node (V) in order to determine whether the message is authentic and unaltered. The verification node validates the query, checks whether the content has a corresponding attestation record, verifies its status, and returns a verification response. Optionally, the verification node may consult a DLT registry where message hashes or attestations are anchored for tamper-evident verification.

10.1 Threat actors

The primary threat actor considered in the framework is an attacker who attempts to manipulate the message or verification process. The attacker may attempt to perform several malicious actions, including:

- Spoof messages, in which a malicious entity impersonates a trusted source and distributes fraudulent messages to recipients;
- Strip metadata, in which message metadata such as sender identity or timestamps are removed or altered to obscure provenance;
- Replay responses, where previously captured verification responses are reused to falsely validate manipulated messages;
- Flood queries, where the attacker sends large numbers of verification requests to overwhelm verification nodes and degrade service availability;
- Fake verification, where malicious actors attempt to deploy fraudulent verification services that falsely claim authenticity of manipulated messages.

These actions may occur through injection or manipulation during the message transmission path or during the verification process.

Table 3 – Threats and mitigations

Threat	Affected component	Mitigation
Spoofing	Message (M), source identity	Verification node (V) attests only anchored or previously observed content using binding element β (e.g., content hash).
Fake verification node	Verification node (V)	Registry of authorized verification nodes; optional DLT anchoring enables independent verification.
Metadata stripping	Metadata (μ) in transit	Verification based on content reference qc remains possible; node may store and return original metadata.
Query correlation	Verification node (V), Aggregator (Ag)	Use of anonymizer entities (An); minimal logging practices.
Denial of service	Verification node (V), DLT infrastructure	Rate limiting, authentication for write operations, and transaction batching.
Replay	Verification response (R)	Responses bound to the queried content reference qc.
Key compromise	Source or verification node keys	Key rotation, issuer registries with revocation capability, and monitoring.

10.2 Threats in the message delivery path

During message transmission from the source (S) through intermediates (I) to the destination (D), attackers may attempt to modify or inject malicious content. In such cases, the received message may appear legitimate even though the content has been altered.

The framework mitigates this threat by allowing recipients to submit the message content or its cryptographic hash to a verification node (V). The verification node compares the received content with previously attested records. If the message content has been altered, the computed hash will not match the stored attestation, and the verification result will indicate that the message is not verified or unknown.

10.3 Threats in the verification process

Attackers may attempt to manipulate the verification process itself. For example, an adversary may attempt to deploy a fraudulent verification service or intercept verification responses.

To address this risk, this framework relies on trusted verification nodes operated by recognized entities such as service providers, organizations, or regulatory bodies. These nodes validate verification queries, retrieve attestation records, and check the revocation status before returning responses.

The framework also supports optional anchoring to DLT infrastructure. When attestation records are anchored on a distributed ledger, verification results can be independently validated using blockchain records, providing tamper-evident evidence of message authenticity.

10.4 Privacy considerations

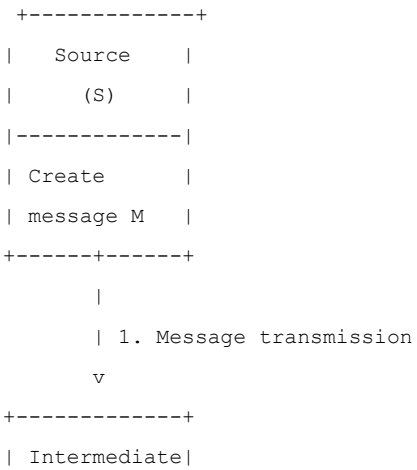
Verification queries may reveal information about the requester or the message being verified. To address this concern, the framework may incorporate an optional privacy layer using anonymizers (An). An anonymizer acts as an intermediary that removes the identifying information from the verification queries before forwarding them to the verification nodes. This mechanism reduces the risk of query correlation, where observers attempt to associate verification requests with specific users or activities.

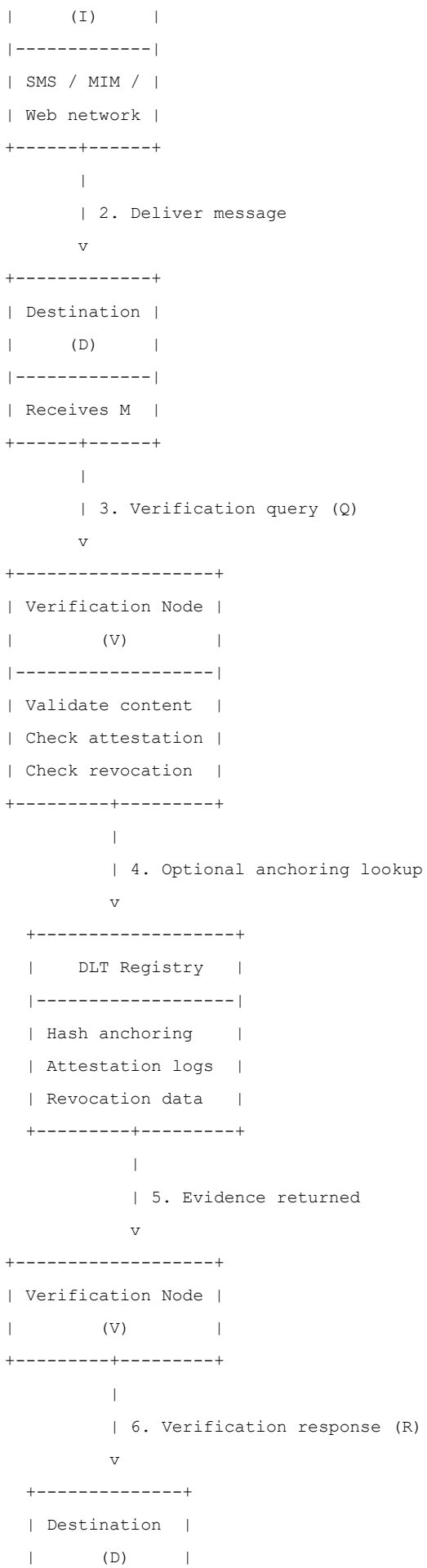
10.5 Security considerations

This framework incorporates several security considerations to maintain the integrity and reliability of the verification process:

- Content binding through cryptographic hashing, which ensures that verification responses correspond to the exact content originally attested by the source;
- Attestation validation, which allows verification nodes to determine whether a message has been previously registered or authenticated;
- Revocation checking, which allows previously attested content to be invalidated when necessary;
- Tamper-evident anchoring using DLT, which enables independent verification of message attestations;
- Privacy protection through anonymization mechanisms, which reduce the risk of linking verification requests to specific users.

Through these mechanisms, this framework provides a structured approach for detecting message manipulation, preventing fraudulent verification, and ensuring that recipients can reliably verify the authenticity of messages across communication channels.




(I)
SMS / MIM /
Web network
2. Deliver message
Destination
(D)
Receives M
3. Verification query (Q)
Verification Node
(V)
Validate content
Check attestation
Check revocation
4. Optional anchoring lookup
DLT Registry
Hash anchoring
Attestation logs
Revocation data
5. Evidence returned
Verification Node
(V)
6. Verification response (R)
Destination
(D)

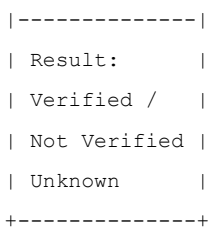


Figure 3 – Message and verification flow

11 Verification procedure

The verification process proceeds as follows:

1) **Message creation**: The source (S) generates a message (M), optionally associated with a content binding element;
2) **Message delivery**: The message is transmitted through a communication network via one or more intermediate entities (I);
3) **Verification request**: Upon receiving the message, the destination (D) submits a verification query (Q) to a verification node (V). The query may contain full content, partial content, or a content reference;
4) **Attestation validation**: The verification node validates the message reference against stored attestations;
5) **Optional anchoring verification**: If anchoring is used, the verification node retrieves the supporting evidence from the DLT registry;
6) **Verification response**: The verification node returns a structured response (R), indicating one of the following states:
   - verified
   - not_verified
   - unknown.

12 Illustrative implementation of the performance observation

12.1 Architecture and deployment

A minimal proof of concept (PoC) was implemented to demonstrate the feasibility of the proposed verification-node architecture. The implementation consists of a verification node exposing a representational state transfer (REST) interface over hypertext transfer protocol secure (HTTPS) and a backend store keyed by content hash (SHA-256). Each store entry contains an attestation record and, when the distributed ledger technology (DLT) is enabled, an anchor reference including the transaction hash, block number, contract address, and chain identifier.

The PoC supports multiple client types, including a web interface, command-line client, and an SMS gateway that forwards message bodies for verification. This design demonstrates that the framework can support device-agnostic verification across multiple communication channels without requiring cryptographic capabilities on end-user devices.

The system was evaluated on a single local machine running a single verification-node process with a store keyed by content hash. Tests were conducted under controlled conditions without network round-trip delays. A seeded government-SMS message was used as the payload to simulate a realistic deployment scenario. For blockchain-enabled anchoring, the system was also connected to an EVM test network using a configured RPC endpoint and signer key.

12.2 Observations and performance

Evaluation results indicate that the architecture is suitable for interactive verification scenarios. In the local environment, the verify endpoint demonstrated sub-millisecond latency (median ≈ 0.41 ms) and throughput exceeding 2 500 requests per second on a single node. These results suggest that verification nodes can support high-volume query workloads with modest infrastructure.

When DLT anchoring was enabled using an EVM test network, the anchor operation latency was dominated by blockchain confirmation times, with a median latency of approximately 11.9 seconds and representative gas consumption of about 68 544 gas per transaction.

Qualitative observations from the PoC confirm several key design objectives of the framework:

- Device-agnostic operation, allowing verification from simple SMS clients, web interfaces, or automated systems.
- Cross-channel verification, enabling content received on one channel (e.g., SMS) to be verified through another (e.g., a web portal).
- Optional DLT anchoring, which provides tamper-evident records and independent verification without requiring end users to interact with the blockchain directly.

The PoC also highlights some operational considerations, including dependency on verification-node operators when DLT is not used, and the need for clear revocation and abuse-resistance mechanisms.

Overall, the implementation demonstrates the practical feasibility of this framework for message and content verification, particularly in environments where SMS and lightweight web interfaces remain dominant communication channels.

13 Benefits and use cases

The intended benefit of this framework would be for governments, institutions, organizations, and service providers to provide these types of services to people or systems to deal with issues of misinformation or malicious communications. It should also be noted that aside from dealing with misinformation, it may also be used to build trust and even provide additional services where users or systems who know to query may be offered additional support, services, or a platform for communications.

An example use case is provided below:

– A government authority or company sends an SMS reminding citizens or customers to go to a new location to verify their identity cards.

  1) The message issuer prepares the verification service before distribution. This may include an HTTPS web service that receives the message text and returns a verification result, an SMS short code or social media identifier that automatically responds to forwarded messages, a web form or API connected to a backend verification node, or specialized hardware or software capable of validating digital signatures.
  2) The official message is processed and, where appropriate, associated with additional information such as a unique identifier, uniform resource identifier (URI), serial number, timestamp, or checksum to distinguish it from other messages.
  3) The message travels through the communication channel. Where the communication channel is configured to support verification, a service provider may generate or preserve verification-related information for use by the verification node.
  4) The recipient may submit the message, a reference, or associated information through the same channel or another channel. For example, an SMS may be forwarded to a secondary number, while an email, social media message, or video item may be associated with a code, checksum, or digital signature.

5) The client or receiving node presents the verification result to the user using an appropriate method, such as a textual status, trust score, colour indicator, pattern, icon, notification sound, or vibration alert.

As part of this Technical Report, various clients or interfaces may automate verification. A configuration may be provided to client software or a service provider to trigger a verification action or perform an examination automatically. For example, a received SMS may display a checkmark indicating that a defined level of checking has been completed, a trust score such as "1/10", a text label or category, a colour indicator such as red, yellow or green, a visual pattern, or a notification type such as sound or vibration when the message is received or opened.

14 Conclusion

This framework proposes a device-agnostic and channel-flexible model for message and content verification based on the use of verification nodes. The framework is designed to address growing challenges related to phishing, spoofing, and misinformation in modern communication systems, where messages may originate from multiple platforms and be consumed on devices with varying technical capabilities.

The analysis presented in this Technical Report compares the proposed framework with existing approaches, including content provenance mechanisms, system attestation frameworks, and distributed-ledger-based solutions. The results indicate that this framework addresses a distinct deployment gap by enabling verification across multiple communication channels without requiring cryptographic capabilities on end-user devices.

A key element of the framework is the use of a minimal verification-node API, which enables interaction with a wide range of clients, including web interfaces, automated systems, and SMS gateways. This design allows the verification services to be accessible from both advanced computing devices and simple communication devices such as feature phones.

Optional integration with DLT, including EVM-compatible smart contracts, provides tamper-evident anchoring and revocation capabilities while preserving a simple user experience. In this architecture, the verification node acts as the primary interface for users and systems, allowing blockchain-based evidence to be used without requiring direct interaction with the ledger.

The feasibility of the framework is illustrated through implementation considerations, a proof-of-concept evaluation, and a deployment scenario involving government SMS communications. These examples demonstrate how the framework can support message verification in environments where SMS and lightweight web services remain the dominant communication channels.

Overall, the framework provides a unified approach to message and content verification that complements existing standards while improving accessibility and deployability. The framework may serve as a basis for further study and potential standardization within ITU-T cybersecurity and messaging initiatives.

Future work may include operational deployments, further evaluation of revocation mechanisms and governance models, and additional studies on usability and performance in real-world environments.

Appendix I

Case studies – Government SMS deployment (Ghana)

I.1 Problem context

In many developing countries, including the Republic of Ghana, SMS and social media are primary communication channels for both official messaging and the spread of misinformation. Citizens often receive messages that may be accurate, outdated, modified, or deliberately spoofed. The absence of a simple and authoritative mechanism to verify such messages can undermine public trust and may have implications for public safety, service delivery, and democratic participation. This case study illustrates how this framework could be deployed to support verification of government-issued SMS communications.

I.2 Deployment scenario

In this scenario, a government authority or telecommunications operator deploys a verification node implementing the framework. The node exposes a REST-based verification API and may optionally be connected to an SMS gateway or short code service.

The deployment operates as follows:

- **Content anchoring**: When an official message is prepared for distribution, the verification node computes a hash of the message content and stores or anchors the hash locally or on an EVM-compatible blockchain.
- **Public verification endpoint**: Citizens are provided with a single, well-publicized verification path, such as a dedicated web portal or SMS short code, allowing them to verify received messages.

This architecture enables message authenticity checks without requiring specialized applications or cryptographic capabilities on user devices.

I.3 Verification process

Recipients of government SMS messages can verify authenticity through two primary mechanisms:

1) **Web portal verification**: The recipient pastes the received message content into a verification form on a public website. The verification node returns a response indicating whether the content matches an anchored record, along with optional metadata such as the issuer and timestamp.
2) **SMS short code verification**: The recipient forwards the received SMS to a designated short code. The verification node processes the message and sends a response indicating the verification result (e.g., "verified – national authority").

In both cases, the verification process is designed to be accessible on basic mobile devices and does not require digital wallets, specialized software, or cryptographic knowledge.

I.4 Impact and benefits

Deployment of this framework provides several potential benefits:

- **Improved authenticity verification**: Citizens gain an immediate mechanism to confirm whether a message originated from an official authority.
- **Inclusivity**: The system operates across both feature phones and smartphones, ensuring accessibility for a broad population.
- **Reduced misinformation impact**: A trusted verification channel reduces reliance on informal information sources and rumour networks.

- **Transparency and auditability**: Optional DLT anchoring provides tamper-evident records and enables independent verification of official communications.

I.5 Stakeholders

The following stakeholders participate in the deployment scenario:

Table 4 – Stakeholder mapping

Stakeholder	Role	Expected benefit
Citizen (recipient)	Receives SMS messages and performs verification	Confidence that messages originate from legitimate authorities
Government authority	Originates official messages and operates or oversees the verification node	Authenticity and integrity of official communications
Telecommunications operator (*optional*)	Provides SMS gateway or short code services and may host the verification node	Fraud reduction and compliance with messaging regulations

I.6 Deployment constraints

One of the primary constraints for deployment is public awareness. Citizens must know where and how to verify official messages. This challenge can be addressed by including verification instructions such as the verification URL or SMS short code in every official communication and through public awareness campaigns. Operational costs can be managed through appropriate infrastructure choices and, when DLT anchoring is used, by employing techniques such as batch anchoring or layer 2 (L2) blockchain solutions to reduce transaction costs.

Bibliography

[b-ITU-T J.380.2] Recommendation ITU-T J.380.2 (2011), *Digital program insertion – Advertising systems interfaces – Core data elements*.

[b-ITU-T X.1252] Recommendation ITU-T X.1252 (2021), *Baseline identity management terms and definitions*.

[b-ITU-T X.1255] Recommendation ITU-T X.1255 (2013), *Framework for discovery of identity management information*.

[b-ITU-T X.1285] Recommendation ITU-T X.1285 (2025), *OpenID Connect Core 1.0 – Errata Set 2*.

[b-C2PA] C2PA (2025), *C2PA Technical Specification.* Coalition for Content Provenance and Authenticity. <https://spec.c2pa.org/specifications/specifications/2.2/specs/_attachments/C2PA_Specification.pdf>

[b-RFC 9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and Pan, W. (2023), RFC 9334: *Remote ATtestation procedureS (RATS) Architecture*. <https://doi.org/10.17487/RFC9334>