mboned
Internet-Draft
Intended status: Informational
Expires: 6 May 2026

O. Ramadan
Blockcast Inc

2 November 2025

SONAR: Statistical Observation Network for Attestation and Reach
draft-ramadan-mboned-sonar-01

Abstract

This document specifies SONAR (Statistical Observation Network for Attestation and Reach), a protocol for verifiable multicast delivery claims without trusted intermediaries. SONAR combines: (1) O(1) IP multicast efficiency versus O(N) unicast to detect cheating, (2) cryptoeconomic accountability via on-chain stake deposits, VRF-based unpredictable sampling, and blockchain attestations, and (3) ALTA-based real-time multicast authentication. SONAR separates content authentication from coverage verification: ALTA authenticates all packets with ~6% bandwidth overhead, while statistical coverage verification adds minimal overhead (320 KB challenge messages per 15-60 minute test period, 0.7-2.8 Kbps). Coverage estimation samples 0.1% of receivers using German Tank Problem inference. For privacy and cost efficiency at scale, zkSNARK proof aggregation (recommended for >1,000 sampled users) maintains O(1) on-chain verification cost, enabling populations exceeding 10^8 receivers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 May 2026.

Table of Contents

1. Introduction

Multicast distribution offers significant efficiency advantages over unicast for large-scale content delivery, reducing bandwidth costs by 99.99% or more. However, the lack of verifiable delivery mechanisms prevents widespread commercial adoption. Content providers cannot verify that infrastructure operators actually delivered content to claimed receivers, while infrastructure operators cannot prove delivery to enable billing. This bilateral trust deficit blocks the formation of liquid markets for multicast capacity.

Existing multicast authentication schemes ([RFC4082], [I-D.ietf-mboned-ambi], [I-D.krose-mboned-alta]) address content authentication but do not provide per-receiver coverage proof. Per-receiver encryption defeats multicast efficiency by requiring O(N) bandwidth where N is the number of receivers.

SONAR solves this problem through statistical sampling: rather than proving delivery to every receiver, SONAR proves delivery to a random sample with known statistical confidence. This enables verification of populations exceeding 10^7 receivers with constant bandwidth overhead.

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

This document uses the following terms:

Coverage Group: A set of receivers that can be addressed with a

single multicast transmission, typically defined by geographic location or network topology.

Attestation: A cryptographically signed statement by a receiver asserting successful reception of specific content with packet statistics.

Sample Size: The number of receivers randomly selected to provide attestations in a given test window. Denoted as m.

Population Size: The total number of receivers in the coverage group. Denoted as N.

German Tank Estimate: A statistical estimator for population size based on maximum observation in a sampled sequence.

Test Window: A time period during which packet reception is tracked for coverage verification. Duration denoted as P.

zkSNARK: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge. A cryptographic proof system enabling verification of aggregate statements with constant-size proofs.

3. Architecture Overview

3.1. Design Principles

SONAR is designed around the following principles:

* Content authentication is independent of coverage verification
* Statistical sampling provides O(1) verification cost regardless of population size
* Broadcast bandwidth overhead MUST be <10% to preserve multicast efficiency
* Return path (user attestations) uses existing internet infrastructure
* Cryptographic security complemented by economic incentives

3.2. Protocol Layers

3.2.1. Layer 1: Content Authentication

All receivers verify content authenticity using ALTA protocol [I-D.krose-mboned-alta]. This provides:

* Non-repudiation: Content provider cannot deny transmission
* Source authentication: Receivers verify authorized origin
* Integrity: Content not modified in transit
* Low latency: 1-10ms authentication delay

ALTA is chosen over alternatives (TESLA, AMBI) because:

* No time synchronization required (vs TESLA)
* Real-time authentication (vs TESLA 100-6000ms delay)
* Broadcast-only (vs AMBI unicast manifests)
* Strong non-repudiation (periodic Ed25519 signatures)

Bandwidth overhead: Approximately 6% for typical configurations.

3.2.2. Layer 2: Statistical Coverage Verification

Random sample of m receivers (typically 0.1% of population) provide attestations via internet return path. Statistical inference provides population coverage estimate with confidence interval.

Sample selection uses Verifiable Random Function (VRF) to prevent adversarial selection. Attestations include packet statistics enabling loss rate estimation and fraud detection.

Broadcast overhead: Only sample selection message (320 KB per test).

Return path overhead: Distributed across m users (128 bps per selected user).

3.2.3. Layer 3: Zero-Knowledge Aggregation (Recommended for m > 1,000)

zkSNARK proofs SHOULD be employed when sample size m exceeds 1,000 users, primarily for privacy protection. Individual viewing patterns become correlatable on-chain, enabling de-anonymization attacks. zkSNARKs provide aggregated proof of coverage statistics while hiding individual user attestations.

Additional benefits: 80-90% cost reduction via off-chain storage, constant-size verification (328 bytes regardless of m), and scalability to populations exceeding 10^8.

Challenge protocol enables spot-checking of individual attestations via Merkle proof while maintaining aggregate privacy.

4. Content Authentication

4.1. ALTA Protocol Configuration

SONAR employs ALTA with the following parameters:

Scheme Parameters:

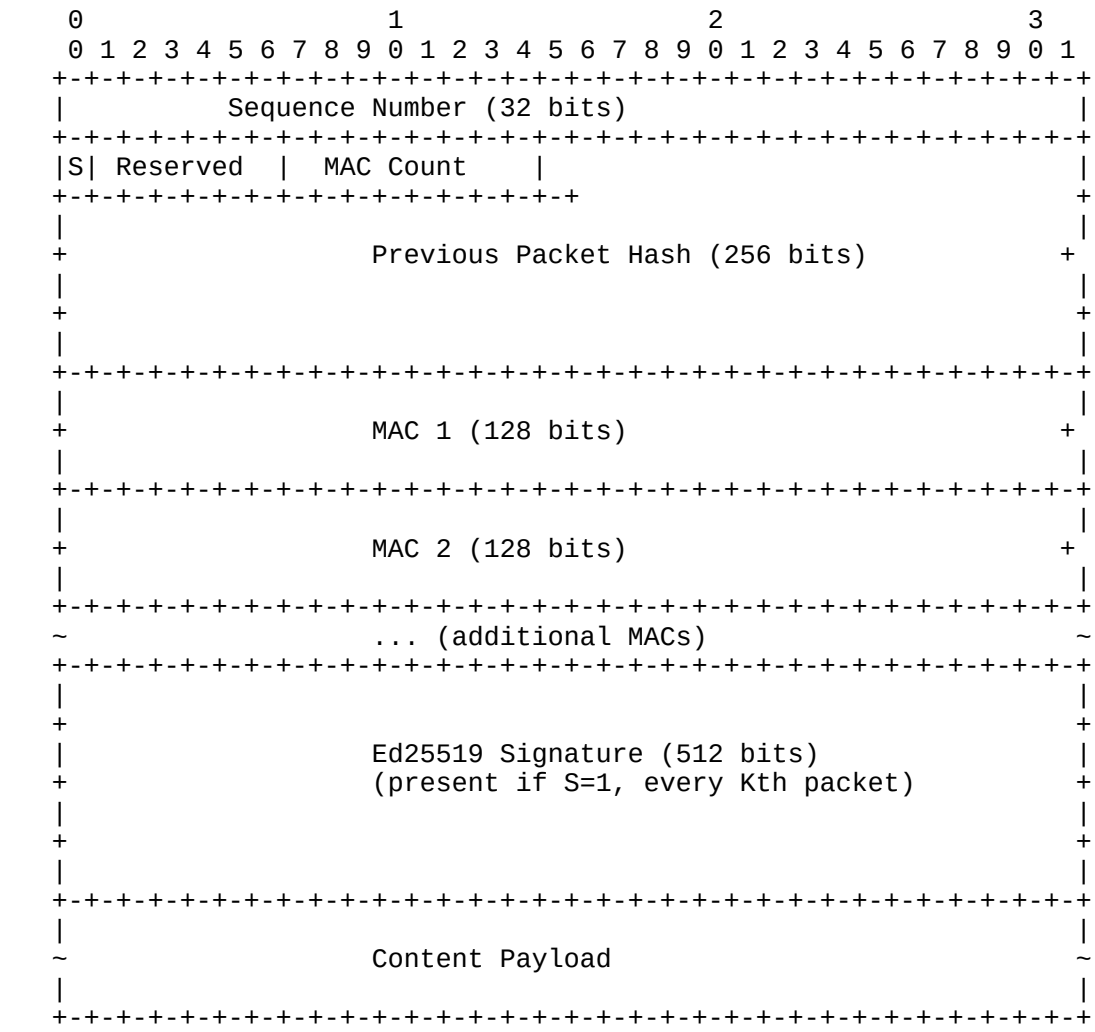
* a = 3 (backward reference interval)
* p = 5 (redundancy factor)
* K = 50 (signature interval)

Algorithms:

* MAC: HMAC-SHA256 truncated to 128 bits
* Signature: Ed25519 (64 bytes)

4.2. Packet Format

Each multicast packet MUST include ALTA authentication data:



Sequence Number: Monotonically increasing packet identifier

S (Signature Present): 1-bit flag indicating Ed25519 signature included

Reserved: 7 bits reserved for future use, MUST be zero

MAC Count: Number of MACs included (typically 3-5)

Previous Packet Hash: SHA-256 hash of previous packet for chain verification

MAC n: HMAC-SHA256 of packet at offset (i - n*a) truncated to 128 bits

Ed25519 Signature: Signature of packets i through i-K+1 (when S=1, every Kth packet)

Content Payload: Application data

Total overhead calculation:

* Base: 4 + 32 = 36 bytes

* MACs: 16 * MAC_Count bytes

* Signature (amortized): 64 / K bytes

* For MAC_Count=4, K=50: 36 + 64 + 1.28 = 101.28 bytes

* Percentage for 1500-byte packets: 6.75%

4.3. Verification Procedure

Upon receiving packet i, receiver performs the following steps:

1. Verify sequence number is monotonically increasing

2. Compute SHA-256(packet_{i-1}) and compare with Previous Packet Hash field

3. For each MAC_j in packet, retrieve stored packet at offset (i - j*a)

4. Recompute HMAC-SHA256 for each referenced packet and compare

5. If S=1, verify Ed25519 signature over packets [i-K+1, i]

6. If all verifications pass, accept packet as authentic

Receiver MUST buffer packets until sufficient MACs received for verification (depth = p packets).

5. Statistical Coverage Verification

5.1. Sample Selection Protocol

5.1.1. VRF-Based Random Selection

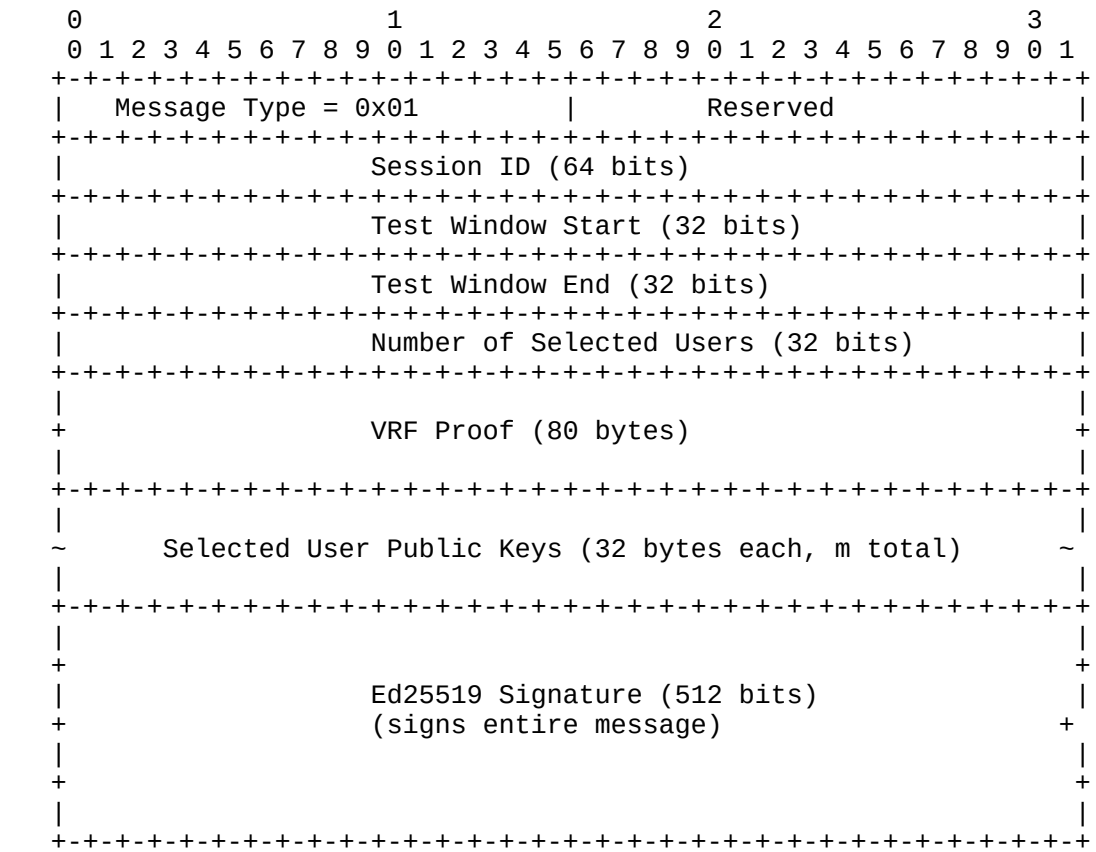
Content provider generates verifiable random sample using VRF to prevent adversarial selection:

1. Obtain blockchain randomness source (e.g., block hash at height H)
2. Apply VRF with content provider private key: seed = VRF_prove(sk, blockhash || session_id)
3. Use seed for Fisher-Yates shuffle of registered user public keys
4. Select first m users from shuffled list

VRF properties ensure:

* Unpredictability: Adversary cannot predict selection before blockhash revealed
* Verifiability: Anyone can verify selection was computed correctly
* Uniqueness: Only one valid output for given input

5.1.2. Challenge Message Format



Size calculation for m=10,000 users:

* Header: 24 bytes

* VRF Proof: 80 bytes

* User pubkeys: 32 * 10,000 = 320,000 bytes

* Signature: 64 bytes

* Total: 320,168 bytes 320 KB

Broadcast frequency: Once per test period P (recommended: P = 900-7200 seconds)

Bandwidth: 320 KB / P seconds

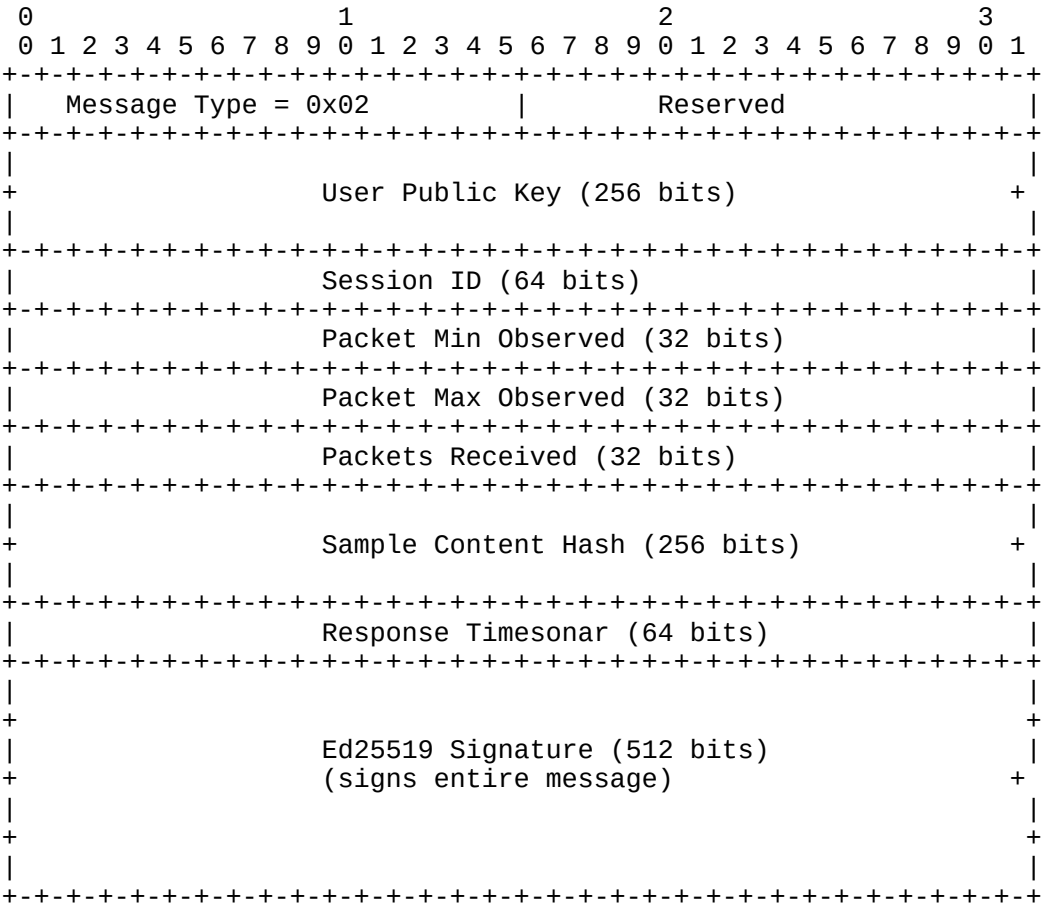
* P = 900s (15 min): 356 bytes/s = 2.8 Kbps

* P = 3600s (1 hour): 89 bytes/s = 0.7 Kbps

5.2. User Attestation Protocol

5.2.1. Attestation Message Format

Selected users MUST respond within response window T_response (recommended: 60 seconds):



Packet Min/Max Observed: First and last sequence numbers received in test window

Packets Received: Total count of packets successfully received and authenticated

Sample Content Hash: SHA-256 hash of concatenated payload from

sampled packets (e.g., every 100th packet) to prove actual content reception

Response Timesonar: Unix timesonar when attestation created

Total size: 160 bytes per attestation

5.2.2. Submission Methods

5.2.2.1. Direct Blockchain Submission

Each selected user submits attestation as blockchain transaction:

* Attestation payload: 160 bytes
* Transaction overhead: ~40 bytes
* Total: ~200 bytes per user
* For m=10,000: 2 MB per test
* Cost at $0.0001/tx: $1.00 per test

Advantages: Simple, immediate verification

Disadvantages: High on-chain cost for large m

5.2.2.2. Aggregated Submission via zkSNARK

Users submit to off-chain aggregator that creates zkSNARK proof:

1. Users submit 160-byte attestations to off-chain storage
2. Aggregator collects m attestations
3. Aggregator builds Merkle tree with root R
4. Aggregator computes aggregate statistics
5. Aggregator generates zkSNARK proof π
6. Aggregator submits {R, statistics, π} on-chain

On-chain size: 328 bytes (constant regardless of m)

Cost reduction: 99.998% vs direct submission for m=10,000

5.3. Coverage Estimation

5.3.1. German Tank Problem Estimator

Given sender transmitted N packets and user j reports packet_max_j, population size estimate:

```
N_hat = max(packet_max_1, ..., packet_max_m)
        + max(...) / m - 1
```

Loss rate estimate for user j:

```
span_j = packet_max_j - packet_min_j + 1
loss_j = 1 - (packets_received_j / span_j)
```

Aggregate loss rate:

```
loss_avg = (1/m) * sum(loss_j for j in sample)
```

If packet_max_j N, user kept pace with real-time stream (multicast reception). If packet_max_j << N, user lagged significantly (potential unicast forwarding).

5.3.2. Confidence Intervals

For sample size m from population N, coverage estimate has confidence interval:

```
Standard error: SE = sqrt(p_hat * (1 - p_hat) / m)
95% confidence: CI_95 = p_hat ± 1.96 * SE
Population coverage: Coverage = N * p_hat
                     Coverage_CI = N * (p_hat ± 1.96 * SE)
```

Example: N = 10,000,000 users, m = 10,000 sample, p_hat = 0.95:

```
SE = sqrt(0.95 * 0.05 / 10000) = 0.00218
CI_95 = 0.95 ± 0.00427 = [0.946, 0.954]
Coverage = 9,500,000 ± 42,700 users
```

Minimum sample size for desired margin of error E:

```
m_min = (1.96^2 * p_hat * (1 - p_hat)) / E^2
```

For E = 0.001 (0.1% margin) with p_hat = 0.95:

```
m_min = (3.84 * 0.95 * 0.05) / 0.000001 = 18,240 samples
```

6. Zero-Knowledge Proof Aggregation

Zero-knowledge proof aggregation via zkSNARKs SHOULD be employed when sample size m exceeds 1,000 users. This threshold is determined by three factors:

1. Privacy Protection: Individual attestations become correlatable on-chain, enabling de-anonymization attacks. For small communities (N < 100,000), users are more easily identifiable, making privacy protection critical.

2. Cost Efficiency: Off-chain storage via Data Anchor costs $0.00001 per attestation versus $0.0001 for direct on-chain submission. For m=1,000, this represents 80-90% cost reduction: $0.02 (zkSNARK aggregated) versus $0.10 (direct).

3. Constant Verification: zkSNARK proofs maintain 200-byte size and O(1) verification cost regardless of m, enabling scalability to populations exceeding 10^8.

Implementation:

* User attestations sent to aggregator (off-chain)

* zkSNARK proof generated in Trusted Execution Environment (TEE)

* Proof verified on-chain via smart contract

* Challenge protocol enables spot-checking via Merkle proofs

6.1. Merkle Tree Construction

Aggregator constructs binary Merkle tree from attestations:

1. Collect m attestations: A_1, A_2, ..., A_m

2. Compute leaf hashes: L_j = SHA256(A_j)

3. Build tree bottom-up: H_parent = SHA256(H_left || H_right)

4. Compute root: R

Merkle proof for attestation A_j:

* Path: Sibling hashes from leaf to root

* Length: ceil(log2(m)) hashes

* For m=10,000: 14 hashes * 32 bytes = 448 bytes

6.2. zkSNARK Proof Generation

Aggregator generates proof π for statement S:

```
Statement S:
  "I know m attestations {A_1, ..., A_m} such that:
   1. MerkleRoot({SHA256(A_j)}) = R
   2. Each A_j contains valid Ed25519 signature
   3. Aggregate loss rate < threshold (e.g., 0.05)
   4. Median(packet_max) > threshold (e.g., 0.95 * N)"
```

Public inputs: {R, m, aggregate_stats, thresholds}

Witness: {A_1, ..., A_m, Merkle_paths, signatures}

Proof size: Approximately 200 bytes (constant regardless of m)

6.3. On-Chain Verification

Smart contract verifies zkSNARK proof:

1. Extract public inputs: {R, m, statistics, π }
2. Verify proof: valid = Verify(vk, public_inputs, π)
3. If valid: Accept coverage claim for m users
4. If invalid: Reject and slash aggregator stake

Verification cost: ~100,000 gas (constant regardless of m)

6.4. Challenge Protocol

Any party may challenge aggregator by requesting Merkle proof for specific user:

1. Challenger submits challenge: "Prove user_j is in tree R"
2. Aggregator MUST respond within T_challenge (recommended: 24 hours)
3. Aggregator provides: {A_j, merkle_path}
4. Challenger verifies: MerkleVerify(A_j, path, R) and signature validity

5. If verification fails: Aggregator stake slashed, challenger rewarded

6. If verification succeeds: Challenge bond returned to challenger

7. Test Period Optimization

7.1. Unicast Replication Detection

A malicious relay might receive content via unicast forwarding and claim multicast reception. Detection relies on bandwidth constraints:

For unicast replication to B recipients:

Required bandwidth: B * R_content
If B * R_content > R_relay, relay must lag
Lag accumulates at rate: (B * R_content - R_relay)

Minimum test period for detection:

P_min = L / (alpha - 1)
where alpha = (B * R_content) / R_relay
L = acceptable loss rate

Example: B=1M recipients, R_content=25 Mbps, R_relay=10 Gbps, L=0.05:

alpha = (1,000,000 * 25) / 10,000 = 2,500
P_min = 0.05 / (2,500 - 1) 0.00002 seconds

Conclusion: For typical broadcast scenarios, any test period P > 1 second provides overwhelming detection certainty. Optimal P is determined by cost-benefit tradeoff, not detection requirements.

7.2. Recommended Test Periods

Test period selection based on use case:

Use Case	Test Period P	Tests/Hour	Cost/Hour*	Detection Latency
Live Events	300s (5 min)	12	$12	<5 min
Prime Time TV	900s (15 min)	4	$4	<15 min
Off-Peak Content	3600s (1 hour)	1	$1	<1 hour
ISP SLA Reporting	7200s (2 hours)	0.5	$0.50	<2 hours

Table 1

*Assumes m=10,000 users, $0.0001 per transaction

Maximum recommended P: 7200 seconds (2 hours)

Rationale: Beyond 2 hours, network state staleness reduces actionable value of coverage data.

8. Security Considerations

8.1. Threat Model

SONAR must resist the following adversarial behaviors:

Sybil Attacks: Attacker creates multiple fake receiver identities to inflate coverage claims. Mitigated by stake requirements and VRF-based random sampling.

Replay Attacks: Adversary captures authenticated content and replays to different receivers. Mitigated by sequence numbers, timesonars, and hash chaining.

Man-in-the-Middle: Intermediate node modifies content while maintaining valid authentication. Prevented by ALTA MAC chains and periodic signatures.

DoS Attacks: Attacker floods network with fake packets to exhaust receiver buffers. Mitigated by instant ALTA authentication enabling immediate rejection.

Sample Manipulation: Adversary attempts to influence which users are selected for sampling. Prevented by VRF unpredictability.

8.2. Economic Security

Cryptographic security is complemented by economic incentives:

* Stake requirements: $10-$100K deposits create accountability
* Slashing penalties: Fraudulent attestations result in stake loss
* Fraud detection rewards: 5-10x multiplier encourages honest reporting
* Rational behavior: Cost of fraud exceeds expected benefit

Game-theoretic analysis shows honest participation is Nash equilibrium when fraud detection probability exceeds 0.001% (easily achieved through random spot checks).

8.3. Privacy Considerations

SONAR reveals the following information:

* Which users are registered for coverage verification (public keys on-chain)
* Which users were selected for sampling (challenge message)
* Aggregate statistics about selected users (loss rates, packet counts)

SONAR does NOT reveal:

* Content of multicast stream (encrypted separately if needed)
* Individual user consumption patterns (when zkSNARK aggregation used)
* Non-selected users’ reception status

Privacy Attack Vector for Small Communities:

Without zkSNARK aggregation, direct on-chain attestations enable correlation attacks. For small communities (N < 100,000), attackers can:

* Link public keys to known wallet addresses

* Correlate viewing times with other on-chain activity

* Cross-reference with geographic or demographic data

Privacy decreases inversely with community size. For N=5,000, individual identification probability exceeds 75% through cross-referencing. For N=10,000,000, crowd anonymity provides natural protection.

RECOMMENDATION: zkSNARK aggregation MUST be used when sample size m > 1,000, SHOULD be used when m > 100. This protects small community viewers from de-anonymization while maintaining cryptographic coverage proof.

9. IANA Considerations

This document requests IANA to create a new registry for SONAR message types:

Registry Name: SONAR Message Types

Registration Procedure: IETF Review

Reference: This document

Initial allocations:

Value	Description	Reference
0x01	Sample Challenge	Section 5.1.2
0x02	User Attestation	Section 5.2.1
0x03	zkSNARK Aggregated Proof	Section 6.3

Table 2

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[I-D.krose-mboned-alta] Rose, K. and J. Holland, "Asymmetric Loss-Tolerant Authentication", July 2019, <https://datatracker.ietf.org/doc/html/draft-krose-mboned-alta-01>.

10.2. Informative References

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J. D., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, DOI 10.17487/RFC4082, June 2005, <https://www.rfc-editor.org/info/rfc4082>.

[I-D.ietf-mboned-ambi] Holland, J., Rose, K., and M. Franke, "Asymmetric Manifest Based Integrity", March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-mboned-ambi-04>.

Appendix A. Example Deployment

A.1. NYC Television Station Scenario

This appendix provides a concrete deployment example for a New York City television station broadcasting to 10 million concurrent viewers.

A.1.1. Network Configuration

* Content Provider: NBC New York
* Content: Live sports broadcast
* Bitrate: 25 Mbps H.265 video
* Target Coverage: 10,000,000 concurrent viewers
* Geographic Area: NYC metropolitan area

A.1.2. SONAR Configuration

Content Authentication (ALTA):

* MAC algorithm: HMAC-SHA256 (128-bit)
* Signature: Ed25519 every 50th packet
* Overhead: 6.75% (1.69 Mbps)

Statistical Sampling:

* Sample size: m = 10,000 users (0.1%)
* Test period: P = 900 seconds (15 minutes)
* Confidence: 95%
* Margin of error: ±0.3%

Broadcast Overhead:

* ALTA: 1.69 Mbps
* Challenge message: 320 KB / 900s = 2.8 Kbps
* Total: 1.69 Mbps (6.76%)

A.1.3. Cost-Benefit Analysis

Per-Hour Costs:

* User sampling: 10,000 users × 4 tests/hour × $0.0001 = $4.00
* zkSNARK aggregation: 4 tests/hour × $0.0001 = $0.0004
* Total: $4.00 per hour

Revenue Impact:

* Traditional CPM (unverified): $10 per 1000 impressions
* Verified CPM: $15 per 1000 impressions (50% premium)
* Traditional revenue: 10M × $10/1000 = $100,000/hour
* Verified revenue: 10M × $15/1000 = $150,000/hour
* Additional revenue: $50,000/hour
* Net benefit: $50,000 - $4 = $49,996/hour

* ROI: 1,249,900%

A.1.4. Performance Metrics

* Broadcast bandwidth: 25 Mbps + 1.69 Mbps = 26.69 Mbps
* Overhead: 6.76%
* Authentication latency: 1-10ms (ALTA)
* Detection latency: <15 minutes
* Coverage confidence: 95% (9,458,000 - 9,542,000 users)
* Blockchain TPS: 0.011 (well below capacity)

Acknowledgments

The author thanks Jake Holland and Kyle Rose (Akamai) for the ALTA protocol specification and insights on multicast authentication. A special thanks to Lenny Giuliano (Juniper Networks), Chris Lenart (Verizon), Neil Chatterjee (DAWN Internet) for real-world deployment experience with decentralized multicast networks and feedback on earlier revisions of this work.

Author’s Address

Omar Ramadan
Blockcast Inc
Berkeley, CA
United States of America
Email: omar@blockcast.network
URI: https://blockcast.network