BASTION

A Post-Quantum Secure Channel Protocol

Version 1.0 · June 2026 · bastionprotocol.org

A Note on Methodology

Bastion was designed and implemented gate by gate. Each security property was specified in a gate document before a line of implementation code was written. The gate document locked the wire format, the key derivation chain, the test cases, and the acceptance criteria. Implementation proceeded only when the design was locked. Tests were written before the implementation they verified. Each gate was independently verified — by security scanner, by static analysis, by test suite, and by manual cryptographic review — before the next gate was opened.

The result is a protocol where every claimed property in this document has a traceable lineage: a gate document that specifies it, an implementation that delivers it, a test suite that verifies it, and a gate check that confirms no regressions. 340 tests. 42 of 42 security gate checks pass. Zero critical or high findings.

This document is a co-founder recruitment vehicle as much as it is a technical specification. Building the production core of this protocol — Rust crypto core, TEE/HSM key isolation, FROST threshold signatures, formal verification — is the work of Phase 2. If you are a systems engineer with cryptography depth reading this and the construction makes sense to you, reach out via bastionprotocol.org.

1. Abstract

Bastion is a post-quantum secure channel protocol. It delivers a formally composable set of cryptographic guarantees — quantum-resistant confidentiality, authenticated sender identity, per-message forward secrecy, per-epoch post-compromise security, replay protection, and tamper-evident audit — over any transport layer.

Messaging is the reference implementation. It is not the product.

The product is the channel primitive: a construction any system can rely on to communicate with certainty that no third party can read the content, no quantum computer can retroactively break the encryption, no attacker who briefly obtains session state can maintain access, and every action taken on the channel is cryptographically authenticated and permanently auditable.

That primitive has immediate applications in regulated-industry compliance infrastructure. DORA, MiFID II, and HIPAA require tamper-evident communication records attributable to authenticated parties — requirements no existing protocol satisfies simultaneously with end-to-end confidentiality. Bastion provides the cryptographic record and non-repudiation layer; the regulated entity's existing identity attestation infrastructure provides the legal identity binding. Together, they close the compliance gap that current enterprise messaging platforms cannot close.

The channel primitive also enables autonomous agent communication at scale, where the economic and audit properties become load-bearing: every agent action is cryptographically authenticated, on-chain-auditable, and economically accountable.

Bastion assumes breach at every layer and designs accordingly.

2. The Problem

Secure communication has two unsolved problems. Existing protocols treat them as separate concerns. They are not. Bastion treats them as one design problem with one construction.

The Quantum Threat

ECDH, RSA, and related algorithms are broken by Shor's algorithm on a sufficiently capable quantum computer. In 2024, NIST finalised post-quantum cryptographic standards: ML-KEM (FIPS 203) for key encapsulation and ML-DSA (FIPS 204) for digital signatures. Nation-state adversaries are already executing harvest-now-decrypt-later operations — collecting classical ciphertext today, to decrypt when quantum capability matures.

This is not a theoretical risk. Any communication that must remain confidential for ten or more years must be quantum-resistant today. No widely deployed communication protocol — Signal, WhatsApp, Telegram, iMessage — provides quantum resistance. Their ciphertext is being collected now.

The Sovereignty Problem

Every major communication platform routes messages through infrastructure controlled by a third party. Even platforms marketed as private operate under jurisdictions that can compel disclosure, on infrastructure that can be seized, under terms that can change. Every mainstream platform requires an external identity anchor — a phone number, an email address — that can be revoked, surveilled, or coerced independently of the user.

Sovereignty means: identity is a keypair generated locally, owned entirely by the user. Messages are encrypted before they leave the device. No server holds plaintext. No intermediary can be compelled to produce communications it does not have.

Why Existing Protocols Do Not Solve This

Protocol	Gap
Signal	No quantum resistance. Phone-number identity. US-jurisdiction central servers. Traffic metadata visible to Signal Inc.
Nostr	Sovereign keypair identity. No quantum resistance. No economic spam resistance. No ordered delivery guarantee.
Session	Decentralised relay nodes. No quantum resistance. 14-day message retention. No economic layer.
Matrix	Federated infrastructure. No quantum resistance. Cross-server metadata exposure. Homeserver dependency.
Bastion	Quantum-resistant hybrid encryption. Sovereign keypair identity. Post-compromise security. Economic spam resistance. Tamper-evident on-chain audit. No central dependency.

3. Channel Properties

A Bastion channel delivers the following security properties. Each property is implemented in the current reference implementation, covered by a test suite that verifies the construction, and derivable from the cryptographic design in Section 4.

Channel Property	Status	Cryptographic Construction
Quantum-resistant confidentiality	**Delivered**	*ML-KEM-1024 + ECDH → HKDF-SHA256*
TOFU key pinning *Protocol property*	**Delivered**	*ML-DSA-87 first-contact pin; mismatch = hard reject*
Out-of-band key verification *User practice, not a protocol guarantee*	**Rec. Practice**	*QR code, voice fingerprint, or trust-path cert*
Per-message forward secrecy	**Delivered**	*Symmetric double ratchet, HKDF chain*
Post-compromise security	**Delivered**	*KEM ratchet injection per reply epoch*
Replay and ordering protection	**Delivered**	*Wire-level message index, hard drop on mismatch*
Tamper-evident state integrity	**Delivered**	*HMAC-SHA256 keyed to device ML-DSA identity*

Authenticated on-chain audit trail	**Delivered**	*ML-DSA-signed actions, BSV OP_RETURN anchoring*
Ephemeral key storage isolation	**In Progress**	*TEE/HSM — Phase 2 priority*
Threshold key operations	**Planned**	*FROST (RFC 9591) — Phase 2*
Network-level metadata protection	**Planned**	*Onion routing — Phase 3*

Named constraint: one-directional communication streams do not benefit from KEM ratchet post-compromise security. The KEM ratchet requires a reply to inject fresh entropy. Applications requiring PCS on one-directional channels must use a separate out-of-band re-keying mechanism.

4. The Construction

Bastion builds its channel properties from five composable security layers. Each layer closes a specific attack surface. The layers were implemented sequentially, each gate-verified before the next was opened.

Cryptographic Primitives

Primitive	Role in the construction
ML-KEM-1024 (NIST FIPS 203)	Post-quantum key encapsulation for session init and KEM
ML-DSA-87 (NIST FIPS 204)	Post-quantum digital signatures for sender authentication
ECDH (secp256k1)	Classical key agreement, combined with ML-KEM for hybrid security
HKDF-SHA256	Key derivation throughout: session init, ratchet chain, KEM injection, state MAC
AES-256-GCM	Authenticated encryption of message content
HMAC-SHA256	Per-message chain authentication and state file integrity

The hybrid construction (ML-KEM-1024 + ECDH) requires an attacker to break both the post-quantum lattice-based hardness assumption and the classical discrete logarithm assumption simultaneously. Neither advance alone compromises the session.

Layer 1 — Session Establishment

First contact uses hybrid key encapsulation. The sender encapsulates to the recipient's static ML-KEM-1024 public key, producing a KEM shared secret and ciphertext. The sender simultaneously derives an ECDH shared secret with the recipient's static identity key. Both secrets are combined via HKDF-SHA256 with a non-zero salt derived from the KEM ciphertext, providing domain separation between sessions:

Computation	Output
HKDF(sha256, ecdh_secret ‖ kem_secret, salt=SHA256(kem_ct), 'bastion-session-v1', 64)	ratchet_seed (64 bytes)
ratchet_seed[0:32]	root_key — parent of all derived chain keys
ratchet_seed[32:64]	Initial symmetric chain material

Session establishment requires only the recipient's on-chain ML-KEM public key. No handshake, no round-trip, no registration server.

Layer 2 — Sender Authentication

Every message is signed by the sender's ML-DSA-87 keypair over the complete signed wire region, which covers every security-relevant field: version, flags, message index, sender and recipient keys, timestamp, conditional KEM ciphertext, next ephemeral public key, IV, auth tag, ciphertext length, and ciphertext. No field in the meaningful payload is outside the signed region.

First contact pins the sender's ML-DSA public key via Trust-On-First-Use. Subsequent messages are hard-rejected if the ML-DSA key does not match. There is no downgrade path. For high-security contexts, out-of-band fingerprint verification before relying on the TOFU pin is the recommended practice: QR code exchange, voice confirmation of a short fingerprint string, or trust path through an existing trusted peer. This is a user practice; the protocol delivers TOFU pinning and hard rejection on mismatch.

Layer 3 — Per-Message Forward Secrecy

The symmetric double ratchet maintains two independent HKDF chain keys: one for sending, one for receiving. Each message derives a unique message key from the current chain key, then advances the chain:

Operation	Computation

Derive message key	message_key = HKDF(sha256, chain_key, salt=chain_key, 'bastion-ratchet-v1-msg', 32)
Advance chain key	next_chain = HKDF(sha256, chain_key, salt=chain_key, 'bastion-ratchet-v1-chain', 32)
Encrypt content	AES-256-GCM(plaintext, message_key, AAD)
Authenticate	HMAC-SHA256(message_key, plaintext) — second authentication layer over plaintext

Compromise of any message key at time T exposes only that message. Past and future message keys are not derivable from a compromised message key. Forward secrecy holds at per-message granularity.

Layer 4 — Post-Compromise Security

The symmetric ratchet alone does not provide post-compromise security: an attacker with the current chain key can derive all future message keys. The KEM ratchet closes this gap.

Every outbound message embeds the sender's current ephemeral ML-KEM-1024 public key as the next_epk field. On reply, the receiver encapsulates fresh randomness to this key, injecting a new KEM shared secret into the root key via HKDF:

Operation	Computation
KEM encapsulation	{ kem_ct, kem_secret } = ML-KEM-1024.Encapsulate(next_epk)
Root injection	new_root ‖ new_chain = HKDF(sha256, kem_secret, salt=root_key, 'bastion-kem-ratchet-v1', 64)
Key rotation	their_epk ← null after single use; fresh (our_epk, our_esk) generated on each send
Security effect	Attacker with full session state at time T cannot predict channel state after next KEM step

The session heals automatically at the next reply epoch. The window of compromise closes without any explicit re-keying action by either party.

Layer 5 — Replay Protection and State Integrity

A 4-byte message index is embedded in the signed wire region and enforced by the receiver against stored ratchet state. Any index mismatch results in a hard drop. The ratchet chain does not advance on a rejected message.

Ratchet state is protected by an HMAC keyed to the device's ML-DSA secret key via HKDF with domain separator 'bastion-state-mac-v1'. The MAC covers the state serialised per RFC 8785 (JSON Canonicalization Scheme), ensuring consistent serialisation across implementations. Any field modification invalidates the MAC and causes the state to be treated as absent. This protects against a write-capable adversary who does not also hold the ML-DSA secret key file.

5. Wire Format

The Bastion v5 wire format is self-describing. Parser behaviour is determined entirely by the wire — not by receiver state inference. Every security-relevant field is in the ML-DSA signed region. A tampered flags byte breaks the signature before the parser acts on it.

Field	Size and Notes
Magic bytes	7 bytes — ASCII 'BASTION'
Version	1 byte — 0x05
Flags	1 byte — bit 0 = kem_ct_present; bits 1–7 reserved ●
Message index	4 bytes uint32 big-endian ● signed
Sender public key	33 bytes ● signed
Recipient public key	33 bytes ● signed
Timestamp	8 bytes unix milliseconds ● signed
KEM ciphertext (conditional)	1568 bytes — present when flags bit 0 = 1 ● signed
Next ephemeral public key	1568 bytes — always present ● signed
IV	12 bytes ● signed
AES-GCM auth tag	16 bytes ● signed
Ciphertext length	4 bytes uint32 ● signed
Ciphertext	n bytes ● signed
ML-DSA-87 signature	4627 bytes — signs all ● fields above
HMAC-SHA256 tag	32 bytes — HMAC(message_key, plaintext)

● denotes fields in the ML-DSA signed region. Fixed header without conditional KEM ciphertext: 1687 bytes. Trailer: 4659 bytes. Minimum wire size per message: 6346

bytes before content — appropriate for high-security channel use; constrained IoT environments should evaluate this overhead against their requirements. Versioned HKDF domain separation strings ('bastion-session-v1', 'bastion-ratchet-v1-', 'bastion-kem-ratchet-v1') provide an explicit negotiated upgrade path for future wire versions without breaking existing sessions.*

6. Identity, Trust, and the Infrastructure Layer

Sovereign Identity

A Bastion identity is a BSV keypair generated locally. No registration, no phone number, no external anchor. The keypair is the identity. It cannot be revoked by a third party, cannot be compelled from an intermediary, and cannot be transferred without the holder's private key.

Identity anchoring follows the W3C Decentralised Identifier standard (DID, did:bsv method). A DID Document published on-chain contains the identity's ML-DSA verification key, ML-KEM public key for session establishment, and inbox service endpoint. Resolution is permissionless: any party can retrieve the DID Document from BSV without authentication or prior relationship.

In-Protocol Trust

Level	Meaning
Unknown	No prior contact. Maximum scrutiny. No trust extension possible from this
Known	At least one valid ML-DSA-authenticated message received. TOFU pin established. Cannot self-promote.
Trusted	Explicit operator designation or Guardian certificate from an existing trusted

A trusted peer issues INTRO: <pubkey> [TTL:<seconds>] to promote a target identity. Only trusted peers can issue introductions. Expiry demotes to known, not unknown. Trust propagates through explicit peer endorsement only.

Why BSV

The BSV infrastructure layer is a design requirement. Three properties make it load-bearing in this construction and are not replicable by a transparency log, a permissioned ledger, or a competing public blockchain.

First: high-throughput unbounded OP_RETURN anchoring at low cost. The Bastion audit trail requires every agent action to be permanently anchored on-

chain. Bitcoin SV's TeraNode infrastructure demonstrates one million transactions per second across six global regions (AWS case study, March 2026). OP_RETURN data payloads are unbounded. Ethereum's gas economics make per-action anchoring prohibitively expensive at scale; Arweave's storage economics differ but lack the transaction throughput required for real-time action anchoring; other chains impose payload size limits or lack the throughput for protocol-scale use.

Second: economic spam resistance as a protocol property. Every Bastion message or agent action requires a micro-transaction. This is not an application-layer rate limit — it is a protocol-level constraint. A well-funded adversary can treat transaction fees as operational costs, but the cost scales linearly with attack volume in a way no application-layer control can enforce.

Third: proof-of-work immutability of the audit trail. BSV's fixed protocol and proof-of-work consensus means the anchored record of every action cannot be retroactively modified by any party, including the operator of the anchoring infrastructure. A permissioned ledger can be silently amended; a PoW chain cannot. This is the property that makes the audit trail trustworthy to external parties — regulators, counterparties, auditors — rather than merely present.

The centralisation risk of TeraNode's current AWS deployment and BSV's current hashrate are acknowledged in Section 8. The mitigation path is progressive decentralisation in Phase 3.

7. Applications of the Channel Primitive

A Bastion channel is a primitive. The following applications are ordered by the degree to which the channel's properties become genuinely load-bearing.

Regulated Industry Compliance Infrastructure

Financial services, healthcare, legal, and defence communications face regulatory requirements that existing enterprise messaging platforms cannot simultaneously satisfy. DORA Article 11 requires demonstrable ICT incident communication integrity with tamper-evident records not under unilateral operator control. MiFID II Article 25 requires immutable transaction-related communication records. HIPAA requires demonstrable end-to-end confidentiality of protected health information without reliance on a third-party audit function. These requirements share a common structure: authenticated sender, immutable timestamped record, demonstrable confidentiality, and the ability to prove compliance without disclosing content.

Bastion provides the cryptographic record and non-repudiation layer for this compliance structure. ML-DSA-87 signatures provide cryptographic non-repudiation — proof that a specific keypair signed a specific message. BSV OP_RETURN anchoring provides the immutable timestamped record that no operator, including the Bastion infrastructure operator, can modify. End-to-end encryption provides demonstrable confidentiality without trusting a server-side record keeper. The regulated entity's existing KYC and identity

attestation infrastructure provides the legal identity binding — the mapping from cryptographic keypair to natural or legal person — that DORA, MiFID II, and HIPAA require for attributing records to accountable parties. Bastion and the regulated entity's identity infrastructure together close the compliance gap; neither closes it alone.

The specific gap in current enterprise messaging is this: Bloomberg Terminal, Symphony, and Slack satisfy the record-keeping requirement but the records are under operator control, which creates a single point of regulatory and competitive risk. A regulator or counterparty cannot independently verify that the records have not been altered. Bastion's on-chain anchoring removes that dependency — the record is on a public PoW chain that no operator controls.

Phase 3 zk-STARK delivery receipts extend this: zero-knowledge proofs of delivery enable compliance attestation without content disclosure, the exact primitive needed for cross-institutional communication in regulated markets.

Autonomous Agent Communication

The Ruflo agent layer demonstrates a construction that does not exist elsewhere in this form: an autonomous agent whose every action is ML-DSA-signed by the authorising identity and permanently anchored on-chain. An agent cannot take an action that is not attributable to the identity that authorised it. A rogue or compromised agent cannot forge actions without the identity's ML-DSA secret key.

This primitive — cryptographically authenticated, economically accountable, on-chain-audited autonomous agent actions — is applicable to AI agent infrastructure where regulatory compliance, auditability, or contractual accountability is required. As autonomous AI agent deployment accelerates, the absence of a trustworthy attribution and audit layer becomes a structural infrastructure gap.

A well-resourced competitor can implement ML-KEM and a ratchet. They cannot replicate the sovereignty property: a proprietary audit trail on a controlled ledger cannot be trusted by a regulated entity or an independent counterparty precisely because the operator controls it. The audit trail is trustworthy only when no single party can modify it. That requires a public, permissionless blockchain. A Microsoft or Google-built version of this, on a proprietary ledger, forfeits the one property that makes external trust possible.

Sovereign Messaging

Messaging is the reference implementation — the application that proves every channel property works end to end. For journalists, lawyers, healthcare providers, financial professionals, and anyone whose communications have consequence, this is the channel that should have existed: quantum-resistant, identity-sovereign, forward-secret, post-compromise-recovering, with no server that can be compelled and no intermediary that can be coerced.

Machine-to-Machine Communication

Any system requiring authenticated origin verification for messages or actions, without trusting the transport layer, can use a Bastion channel as its

communication primitive. Note that the minimum wire overhead of 6346 bytes per message makes Bastion appropriate for high-value authenticated communication rather than high-frequency low-payload IoT telemetry. The economic layer makes identity spam irrational at scale; the ML-DSA authentication makes origin forgery computationally infeasible.

8. Security Considerations

A protocol that does not name its limitations is a protocol that cannot be trusted. The following are Bastion's known constraints, the attack surface they represent, and the current or planned mitigations.

First-Contact Key Establishment

Trust-On-First-Use means first contact is the moment of maximum exposure. A network-position adversary capable of intercepting and modifying traffic between two parties making first contact could substitute a public key at the moment of TOFU pinning. The attacker would also need to register the substituted key in a DID Document on BSV before contact — raising the bar, but not eliminating the risk for adversaries capable of both.

The recommended mitigation for high-security contexts is out-of-band fingerprint verification: QR code exchange, voice confirmation of a short fingerprint string, or trust path through a mutually trusted peer who issues a Guardian certificate. This is a user practice. Phase 2 makes DID resolution a mandatory step in session establishment, tightening the protocol's contribution to first-contact security.

Metadata Visibility

Every Bastion session establishment, prekey rotation, and anchored agent action is a public on-chain event. The communication graph — which identities communicate with which, at what frequency and volume, and when — is observable by any chain watcher. Identities are pseudonymous but not unlinkable: a sufficiently motivated observer can build a communication graph from on-chain events without decrypting any content.

For the regulated industry compliance use case, this has a specific implication: financial services firms should evaluate whether their counterparty communication graph is sensitive before adopting Bastion for external communications. For internal enterprise communications, legal communications, or healthcare communications where the counterparty graph is not itself sensitive, metadata visibility is not a barrier. For external financial services communications with metadata-sensitive counterparties, Phase 3 onion routing is the mitigation. Until Phase 3 ships, users with metadata-sensitive threat models should treat the communication graph as observable.

Ephemeral Key Material at Rest

Ephemeral ML-KEM secret keys are stored on disk alongside ratchet chain state, protected by a state MAC. This protects against a write-capable adversary who does not also hold the ML-DSA secret key file (stored separately at 0600). An adversary with read access to both files has full session access. Full isolation requires TEE/HSM — the first Phase 2 deliverable and on the critical path before high-security deployment.

State-Save Desync

Ratchet state is saved before network send. A crash between save and send advances the sender's chain without the peer receiving the message. Recovery requires both peers to re-initialise the session. This is a documented operational trade-off, not a security vulnerability.

BSV Infrastructure and Hashrate

TeraNode's current deployment runs on AWS across six regions, creating an availability and potential compulsion dependency. BSV's current hashrate is a fraction of Bitcoin's, making a 51% attack a realistic rather than purely theoretical risk — this would not compromise message confidentiality or authentication, but could affect the ordering and finality of on-chain anchored records. Phase 3 introduces multiple independent node operators across jurisdictions. Users requiring the strongest finality guarantees for anchored records should await Phase 3.

Export Classification

ML-KEM-1024 and ML-DSA-87 may be subject to EAR/ECCN export classification under US export control regulations. Implementers and deployers should verify compliance with applicable regulations in their jurisdiction.

9. Roadmap

Phase 2 is sequenced by dependency and risk. The Rust crypto core must precede TEE/HSM integration. TEE/HSM must precede FROST threshold operations. Formal verification runs against a stable implementation. Co-founder recruitment is a present operational priority — it is the prerequisite for Phase 2 execution, not a line item within it.

Phase	Deliverables	Status

Phase 1 **Reference** **Implementation**	Complete post-quantum hybrid ratchet stack: ML-KEM-1024 session init, ML-DSA-87 sender auth, symmetric double ratchet, KEM ratchet per reply epoch, wire-level replay protection, device-bound state integrity MAC (RFC 8785 canonical serialisation). TypeScript reference implementation on BSV testnet. 340 tests. 42/42 security gate checks. Zero critical or high findings.	**Complete**
Phase 2 **Production** **Core**	1. Rust crypto core rewrite — constant-time implementations, aws-lc-rs for ML-KEM, formal API surface for TEE. 2. TEE/HSM ephemeral key isolation — Intel SGX, AMD SEV-SNP, ARM TrustZone with remote attestation anchored to DID. 3. FROST threshold signatures (RFC 9591) — 2-of-3 Guardian. 4. ProVerif formal verification of ratchet properties across unbounded sessions with explicit adversary state-observation model. Migration: Phase 1 sessions re-initialised on upgrade; state format not forward-compatible.	**Active**
Phase 3 **Network** **Infrastructure**	BSV-incentivised relay network with on-chain reputation. Onion routing for network-level metadata protection. Progressive Guardian decentralisation across independent TeraNode operators across jurisdictions. zk-STARK delivery receipts for DORA/MiFID II compliance attestation without content disclosure.	**Planned**
Phase 4 **Ecosystem**	Open application ecosystem on the channel primitive. Ruflo agent infrastructure as a general-purpose authenticated agent layer. Cross-protocol compatibility. Versioned protocol negotiation. Community governance.	**Planned**

10. Conclusion

The protocols that exist today have made important contributions. Signal proved strong encryption can be deployed at scale. Nostr proved sovereign keypair identity is achievable. Session proved decentralised relay infrastructure can work without a central server. None provide quantum resistance. None provide post-compromise security at the per-epoch level. None provide a tamper-evident, cryptographically authenticated audit trail as a protocol property. None combine economic spam resistance with an authenticated agent layer.

Bastion delivers all of these properties in a single coherent construction. The channel primitive it provides — quantum-resistant, forward-secret, post-compromise-recovering, identity-authenticated, economically-accountable, on-chain-auditable — does not exist elsewhere in this combination.

The compliance infrastructure use case is the immediate commercial application: organisations that need tamper-evident communication records attributable to authenticated parties, where neither the record nor the attribution can be under unilateral operator control. The autonomous agent infrastructure use case is the structural long-term opportunity: a world of AI agents that need to act with cryptographic accountability, at scale, over channels that cannot be forged or repudiated.

A well-resourced competitor can implement ML-KEM and a ratchet. They cannot replicate the sovereignty property: a proprietary audit trail on a controlled ledger cannot be trusted by a regulated entity or an independent counterparty, because the operator controls it. The audit trail is only trustworthy when no single party can modify it. That requires a public, permissionless blockchain. That is why the infrastructure choice is not incidental — it is the moat.

The path from reference implementation to production protocol requires a Rust engineer with cryptography depth, a ProVerif formal security proof, and TEE/HSM infrastructure. That is Phase 2. The design is done. The construction is verified. The channel works.

Messaging is what we built to prove the channel works. The channel is what matters.

References

- S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
- NIST FIPS 203 — Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), August 2024.
- NIST FIPS 204 — Module-Lattice-Based Digital Signature Standard (ML-DSA), August 2024.
- D. Connolly, C. Komlo, I. Goldberg, C. A. Wood, “FROST: Flexible Round-Optimized Schnorr Threshold Signatures,” IETF RFC 9591, June 2024.
- T. Perrin, M. Marlinspike, “The Double Ratchet Algorithm,” Signal Foundation, November 2016.
- W3C Decentralised Identifiers (DIDs) v1.0, W3C Recommendation, July 2022.
- D. J. Bernstein, T. Lange, “Post-quantum cryptography,” Nature 549, 188–194, 2017.
- V. Shoup, “A Proposal for an ISO Standard for Public Key Encryption,” 2001. (IND-CCA2 security definition)

- M. Sporny, D. Longley, G. Kellogg, M. Jones, D. Chamberlin, D. Chadwick, “JSON Canonicalization Scheme (JCS),” IETF RFC 8785, June 2020.
- European Banking Authority — DORA (Digital Operational Resilience Act), Article 11: ICT incident management and reporting requirements.
- European Parliament — MiFID II, Article 25: Obligation to keep records of transactions and communications.
- BRC-18 — OP_RETURN data anchoring standard, BSV Request for Comments.
- BRC-31 — Authrite mutual authentication standard, BSV Request for Comments.
- AWS Case Study: TeraNode — 1 million TPS across six global regions, March 2026.