RSW-0015 — Key Rotation and Revocation

Revision 0.01 | Protocol rsw/1 | Format 1 | Experimental proposal

Developed by Pablo Murad — November 23, 2026 — pablo@pablomurad.com — https://rsw.works

Experimental Project Specification. This document is an experimental specification of the Really Small Web project. It is a work in progress and has not been published as an RFC or approved as an Internet Standard.

License: specifications CC0 1.0 (public domain); future reference implementation and official tools MIT.

RSW-0015 — Key Rotation and Revocation

Revision 0.01 | Protocol rsw/1 | Format 1

Post-MVP — experimental draft; not enabled in basic rsw/1

Developed by Pablo Murad

Pablo Murad — November 23, 2026

pablo@pablomurad.com

MUST and MUST NOT indicate requirements of the described profile; SHOULD indicates a recommendation; MAY indicates an option. Experimental document, not yet ratified or implemented. The rules of RSW-0001 through RSW-0003 remain applicable.

01. Status and Identities

Post-MVP key-management proposal; it does not redefine the stable identity (A,name) of RSW-0003. A new key means a new identity. There is no automatic transfer of domains, trust, permissions, followers, or historical signatures. Rotation is a proposed and consented proof of continuity, not a rewriting of the past. Losing a key without a recovery plan remains unrecoverable by this profile.

02. Encryption Binding Events

Use the RSW-0008 envelope. key-binding: exact body {purpose,encryption_key,sequence,previous}. purpose literal "messaging"; encryption_key the textual X25519 of RSW-0014; sequence integer 1..9007199254740991; previous null in sequence 1, otherwise the id of the previous binding. The external Ed25519 signature binds the key to the identity. It is not proof that someone possesses the X25519 secret; confirm operational capability before use. Persist the largest accepted sequence, require verifiable continuity or explicit gap approval; the same sequence with a distinct payload is a conflict. A specific revocation remains terminal for that binding. Do not automatically transfer trust from a previous encryption key to the next one; show the change and require approval.

03. Identity Rotation

key-rotation: payload.public_key is the old key. Exact body {new_key,proof}; new_key is a distinct, valid Ed25519 identifier. proof is 128 lowercase hex of a strict Ed25519 signature of the new key over ASCII("RSW/1:key-rotation-proof") || NUL || JCS({old_key,new_key,nonce,created_at}), where old_key=payload.public_key and nonce/created_at come from the envelope. The external signature of the old key also authenticates the proof. The receiver verifies both, rejects self-rotation, cycles, and divergent proposals; a proposal is not proof that the old key was not compromised. A new publication continues to be (new_key,name); optionally show a verified continuity link after user action. Updating the domain to the new key requires a binding confirmation per RSW-0005/0006; do not carry high-water marks between keys. Do not automatically follow rotation chains; limit analysis to 16 local links.

04. Revocation

key-revocation: exact body {scope,target}; scope enum identity or encryption-binding. If identity, target must be the envelope's own public_key. If encryption-binding, target is the id of the binding signed by the same identity. Verify authorship and association before applying. An accepted revocation record is terminal locally;

never erase it upon receiving an older announcement. To avoid ambiguity, do not use created_at as proof of the compromise moment. Mark historical signatures as cryptographically valid but known-revoked identity; do not automatically accept new publications from that key as updates. If there is an identity revocation, do not send new encrypted messages to its bindings. Do not treat absence of news as absence of revocation. Without mandatory infrastructure there is no guaranteed propagation; make events available through authorized sources and manual import.

05. Custody and Future Recovery

Keep the old seed in a protected offline backup; avoid copying secrets to hosting; prepare rotation while both keys are under control. In case of suspected compromise, stop use, communicate through independent channels, and do not trust only proof produced by the compromised key. A pre-compromise recovery key would be a distinct extension with an explicit policy for authorization, conflict, and revocation; not defined in this revision. There is no reset by email, password, or central authority. Third-party revocation events are hints, not authenticated decisions.

06. Tests and Pending Items

Test correct binding, another author, repeated sequence, missing new proof, wrong signature, rotation conflict, cycle, duplicate revocation, replay, and out-of-order receipt. Test state persistence after cache clearing and restart. Pending: pre-authorized recovery, trust policy under compromise, revocation distribution, interoperability, and cryptographic review. Do not activate an automatic rotation mechanism. References: RSW-0003/0005/0006/0008/0013/0014/0016; RFC 8032 https://www.rfc-editor.org/rfc/rfc8032.html .

License: CC0 1.0 (public domain dedication; see CC0-1.0.txt in the downloads area). Developed by Pablo Murad | pablo@pablomurad.com END — RSW-0015

Author's Address

Pablo Murad
Independent
pablo@pablomurad.com
https://pablomurad.com
Official project website: https://rsw.works

Small files. Independent worlds.