RSW-0014 — Message Encryption

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-0014 — Message Encryption

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 Privacy Model

Post-MVP proposal, dependent on RSW-0008/0009/0015 and independent cryptographic review. It does not change cartridge Ed25519 and is not encryption of the PNG or ZIP. Defines a candidate for individual, asynchronous private messages, without groups, attachments, or guaranteed forward secrecy. Do not activate in production before vectors and an audit. The node still observes sender, recipient, times, size, and IP. A compromised endpoint can read messages; local deletion does not eliminate external copies.

02. Separate Encryption Key

Each Ed25519 identity may publish an independent X25519 key via a key-binding event per RSW-0015. Generate the secret with a CSPRNG and an established library; never automatically convert an Ed25519 seed, reuse a signing key, or derive from a password. Identifier x25519: + 52 uppercase RFC 4648 Base32 without padding, canonical decoding of 32 bytes. The destination key needs a verified, active binding accepted locally, checked through an independent channel when necessary. An unexpected change requires confirmation. Do not use a revoked or unknown key; do not fall back to plaintext.

03. Candidate Suite

Use HPKE RFC 9180 Base mode (0x00), DHKEM(X25519, HKDF-SHA256) 0x0020, HKDF-SHA256 0x0001, ChaCha20Poly1305 0x0003. Use an established HPKE library, validate inputs, and reject an all-zero shared secret. One encapsulation and one Seal per message; a new context and new randomness, without reuse. Do not invent own nonces, DH, or AEAD. Sender authentication comes from the Ed25519 signature of the RSW-0008 envelope, not from HPKE Base. No forward secrecy against later compromise of the recipient's static key.

04. Event and Bytes

Type encrypted-message, RSW-0008 envelope. Exact body {to, binding, enc, ciphertext}; to is the recipient's Ed25519 public_key; binding is the SHA-256 id of the key-binding event; enc and ciphertext are canonical RFC 4648 Base64url without padding, decoded and re-encoded for checking. enc has exactly 32 bytes. The internal text is UTF-8 of 1..4096 bytes, without controls except LF/TAB; plaintext never appears in external fields. For the candidate version:

```
info = ASCII("RSW/1:private-message") || NUL
aad = JCS({protocol,stream,type,public_key,created_at,nonce,to,binding})
```

The first six fields come from the external payload; to and binding come from the body. Do not include enc, ciphertext, id, or signature in the aad. HPKE SetupBaseS(pkR, info) generates enc and the context; Seal(aad, plaintext) generates the ciphertext. Insert enc/ciphertext into the body, compute the id, and sign the whole payload per RSW-0008. Maximum decoded ciphertext 4112 bytes (text + 16-byte tag). Do not append a plaintext hash. Do not log the key, plaintext, or internal HPKE state.

05. Reception and Replay

Validate limits, schema, id, and the envelope's strict signature first. Check that to matches the local identity, locate the private key by the authenticated binding, reconstruct info and aad exactly, and use SetupBaseR and Open. Failures return a generic decryption message; do not expose a remote oracle. Verify UTF-8 and limits after Open; display inert text. Deduplicate by id; do not produce automatic receipts. Store the secret in protected storage, separate local state from public events. Key rotation affects future messages; keeping an old key offline may be necessary for history, with explicit risk. Revocation does not erase already-encrypted messages and does not protect old ciphertext from a compromised key.

06. Tests and Blockers

RFC 9180 vectors and RSW's own vectors of info, aad, enc, ciphertext, and envelope; compare two implementations. Test wrong signature, tampered recipient/binding/nonce, wrong key, invalid tag, truncated ciphertext, replay, all-zero DH, revoked key, missing binding, and limits. Pending: library selection, storage and backup policy, metadata, moderation, contact protection, forward secrecy or a future session protocol. Forbid claiming full security without review. References: RFC 9180 https://www.rfc-editor.org/rfc/rfc9180.html ; RFC 7748 https://www.rfc-editor.org/rfc/rfc7748.html ; RFC 4648 https://www.rfc-editor.org/rfc/rfc4648.html ; RSW-0003/0008/0009/0015.

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-0014

Author's Address

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

Small files. Independent worlds.