

bsecret: a vault whose hosts never hold a key

bsecret keeps an organization's secrets (passwords, tokens, keys, small files) on hosts that hold ciphertext and never a key. Readers are the members of a borg group; when one is removed, the vault owes a re-key, and once it acknowledges the removal every host refuses anything else until the re-key is done. Every version, grant and rotation is a mined fact, so an auditor verifies the whole trail against block headers alone, with no cooperation from the hosts or the owner. BSV Layered Multicast delivers each version's record once to every host.

Author: Jeff Harris · Lightweb Inc. · jeff@lightweb.net

Pattern 004 · Version 01

Audience: application developers, founders and business leaders. No Bitcoin or networking background assumed; every term is explained on first use.

We = Lightweb Inc., designer of the bstack patterns and operator of the 1BSV network; **you** = a builder or decision maker evaluating them.

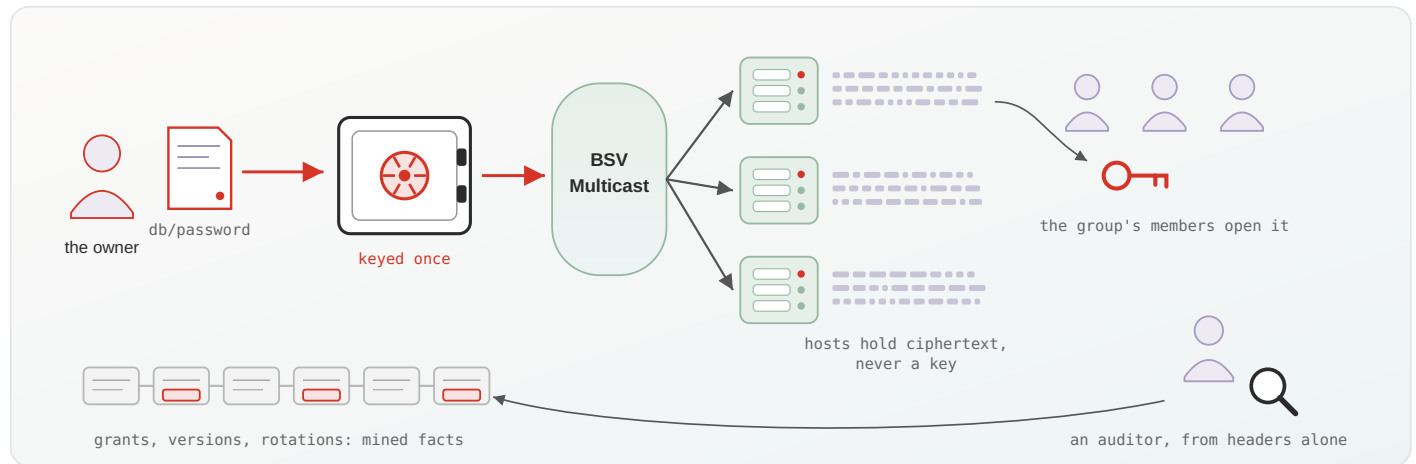


Figure 1: keyed once, held as noise, opened only by the group, audited by anyone.

The series, and where this fits

The bstack pattern series rebuilds the building blocks nearly every product needs so that the data defends itself: state proves itself on the BSV ledger, BSV Layered Multicast delivers each publication once to every subscriber, and readers verify instead of trusting whoever served them.

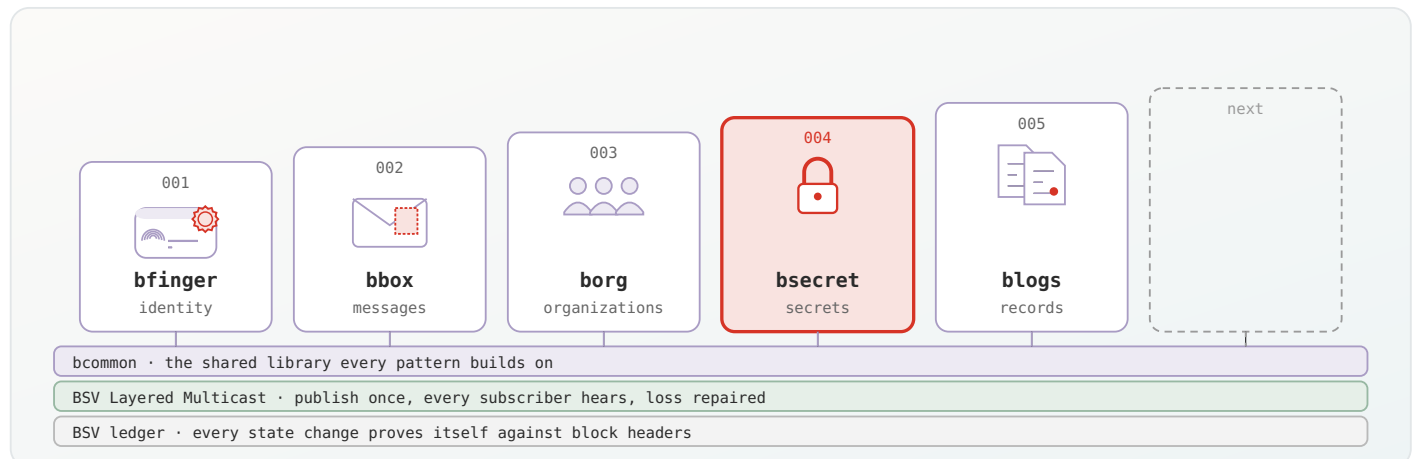


Figure 2: the series as one build. bsecret, in red, is the fourth pattern and the first built on another.

So far. bfinger (001) proved identity, bbox (002) delivered privately to one party with payment inside, and borg (003) gave teams a membership roll and a group key that changes when anyone leaves.

This one. bsecret is the first pattern that stands on another. It adds almost nothing new to the substrate: it takes borg's groups and epochs as its access control, and spends its design on the question every security team is asked: who could read this secret, under which key, and from when to when?

The problem

Every company runs on secrets: database passwords, API tokens, signing keys, certificates. They live in vault services, cloud key managers and team password managers, and each of these asks for the same two kinds of trust. First, trust in the operator: the service that stores the secret can, in principle, read it, or be made to. Second, trust in the log: the record of who was granted what, and when access was removed, is the operator's own audit log, which an auditor must take on faith.

And removal is usually softer than it sounds. Taking someone out of a vault stops the server serving them; it does not change the key that protected what they already had, unless someone remembers to rotate.

What bsecret does

An owner creates a vault bound to a borg group. Putting a secret encrypts it once and publishes the ciphertext; any member of the group reads it; removing a member forces a re-key; anyone can audit.

```
$ bsecret put db/password < new-password.txt
$ bsecret get db/password | psql ...
$ bsecret rotate -check -set db/password=new.txt -same-all
$ borg org remove <bob> -group engineering
$ bsecret rotate -set db/password=new.txt -same-all
$ bsecret -org <owner key> -vault <vault id> audit -free # no home, no key, no coin
```

Figure 3: put, get, remove and re-key, audit. A value is never taken from an argument, so it is in no process list.

Every read reports where the vault stands: **settled** when it owes nothing; **in debt** while some secrets are still under an old epoch; **EXPOSED** for a value keyed under an epoch a removed member held. A script that must not act on such a value asks for strict mode and is refused it.

How it works

1. **A secret is a keyed object.** Each version is encrypted under its own random content key, committed to before anyone holds it (BRC-369). The ciphertext is public; hosts hold it and cannot read it.
2. **Access is group membership.** The content key is wrapped to the owner and, once, under the current epoch key of the vault's borg group. Adding a reader is adding a group member; nothing changes in the vault. A version costs the same for three readers or a thousand.
3. **Revocation forces rotation, and every host checks it.** A vault is a chain of mined tokens. Once it acknowledges a new epoch, hosts admit nothing onto that chain but the re-keying or retirement of what it held, until nothing is left under an earlier epoch. No new secret, no routine update, while one is stale.
4. **The chain is the audit trail.** Which versions the owner listed, in what order, under which epoch, and how long each stayed exposed to a removed member's key: all of it is checked against the auditor's own block headers.

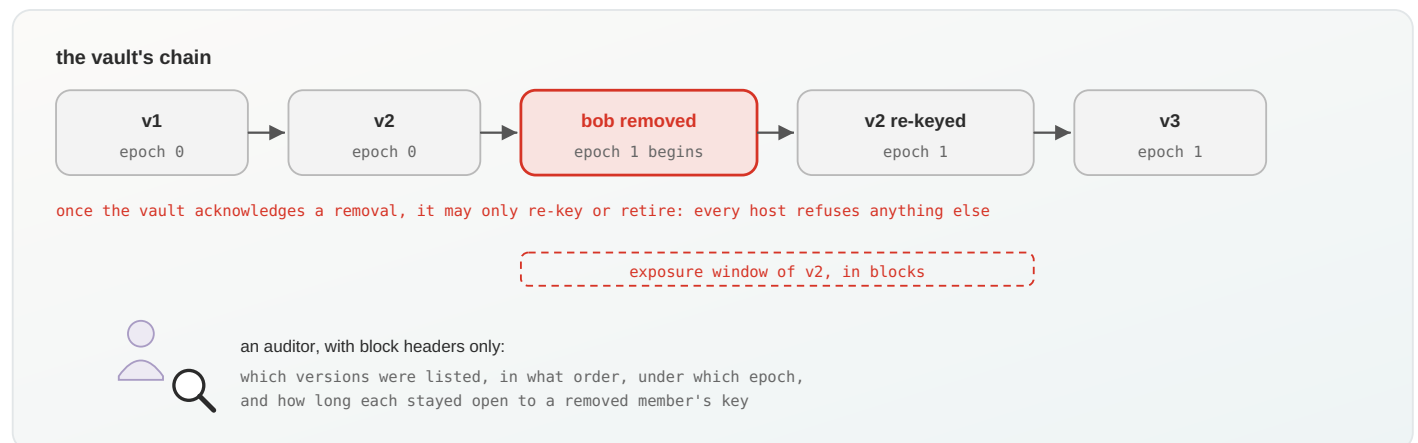


Figure 4: a vault's chain. Once it acknowledges a removal, the vault may only re-key or retire; the exposure window is on the record.

For credentials that machines rotate every few minutes, an **interval vault** publishes versions immediately and, at an interval of the owner's choosing (at least once per 64 versions), lists the latest of each secret in one anchor. The price is assurance: an unlisted version is not yet in the audit trail.

Words in this brief. *Ciphertext*: encrypted data, indistinguishable from noise without the key. *Content key*: the random key one version is encrypted under. *Epoch*: one generation of a borg group's key. *Re-key*: publish the same value under a fresh content key wrapped to the current epoch. *Exposure window*: the blocks during which a removed member's key could still open a version.

Why it is new

- **The host never holds a key.** Hosts store and serve ciphertext only: a host can withhold, but it cannot forge, alter or decrypt. Choosing a host is an availability decision, never a confidentiality one.
- **Removal that actually rotates, enforced by every replica.** Not a checklist item for an administrator: a vault that has acknowledged a removal cannot do anything else until it has re-keyed.
- **An audit that needs nobody's cooperation.** An auditor with block headers verifies grants (in borg), versions and removals, and the exposure window of every version, signed by the owner, needing no one's permission or key, no trust in the hosts that serve the trail, and no access to the secrets.
- **Composition, not reinvention.** bsecret defines no certificate and no role of its own; it reuses borg's groups and the series' shared foundation. The fourth pattern is small because the first three exist.

Built to scale: BSV Layered Multicast

Secrets are read by servers, pipelines and people across regions and clouds, so a vault worth running has hosts near all of them. Point to point, every version is uploaded once per host.

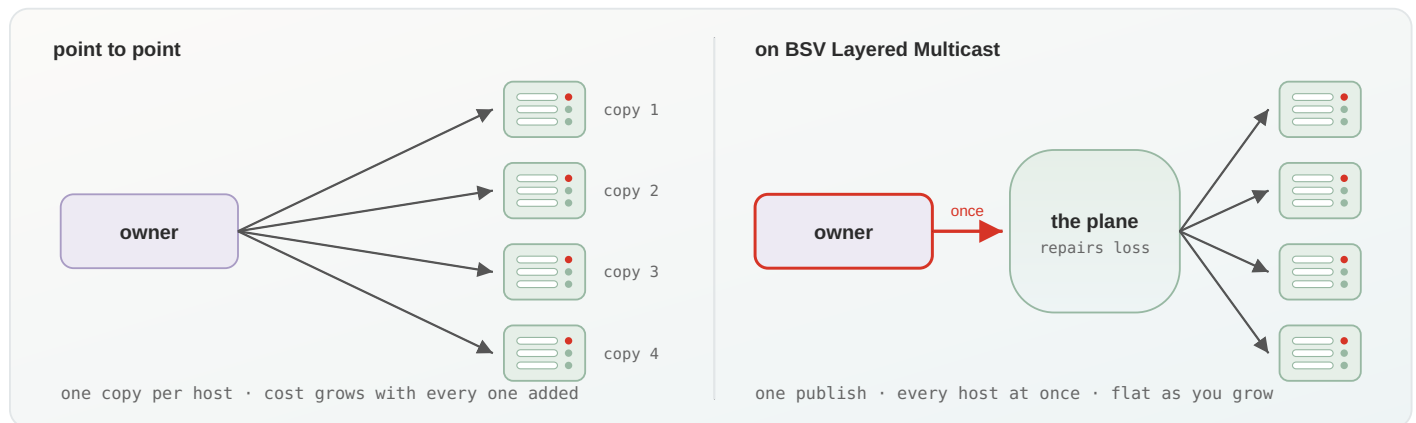


Figure 5: one publish per version record, whatever the number of hosts.

On BSV Layered Multicast each version's record, the ciphertext, is submitted once and every subscribed host receives it, repaired by the network; the small mined token that lists it is also handed to each host the owner names. A record does not grow with the group, so a thousand readers cost what three do. Each vault is its own chain, so vaults never slow each other and different vaults move in the same block. bsecret runs without the network too, sending to a few hosts directly; past a handful, the plane is what removes the cost.

What it means for the business

Who	Pays	Gets
The vault owner	miner fees per version, a fraction of a cent	secrets no host can read; removal that provably re-keys
A member	nothing to read	the current value, with its standing (settled, in debt, exposed)
An auditor	nothing for the base checks	the full trail, verified against its own headers
A host operator	its hosting and the delivery it elects	paid bulk history and audit pages, through a 402 the client pays

- **Compliance evidence that proves itself.** Who could read what, under which key, and for how long after a departure, is a verifiable fact, not an export from a vendor.
- **Vendor risk shrinks to availability.** A compromised or careless host leaks noise; the worst it can do is stay silent, and a second host answers.
- **Offboarding with a receipt.** The exposure window of every secret is on the record, signed by the owner, so a security review starts from facts.

Honest limits

- **A removed member keeps what they could read.** Removal protects what is written after the vault re-keys; a credential they saw must still be changed at the service that accepts it.
- **Reads leave no footprint.** The trail proves grants, versions and removals, never who read what, or that anyone did.
- **Metadata is public.** Names, sizes and timing are visible, including when the vault re-keyed, which says when the group removed someone.
- **Nothing forces the owner to acknowledge a new epoch;** readers and auditors see a vault that has not.
- **No forward secrecy, and the organization's key is everything.**
- **Retraction is not erasure.**

What comes next

bsecret answers who could read a secret, and when. Pattern 005, **blogs**, turns the series from state to volume: many producers writing signed, numbered batches to many archives at once, with one anchor per interval that lets any archive prove its copy complete and any auditor prove when a line was committed. The vault keeps the keys; the log keeps the record of everything that happened around them.

Run it on 1BSV

The 1BSV network is the commercial operation of BSV Layered Multicast, run by Lightweb Inc. For an application built on bsecret:

- **Owners ride free.** Publishing into the network is never billed, and neither are the network's repair retransmissions.
- **Hosts pay for what is delivered to them**, and for nothing else: overlay objects with their proofs on an application account, which also carries the live header feed to verify against. No subscriptions, no minimums, no commitments.
- **Early access.** The overlay lane is onboarding its first users. Onboarding is self-service with one approval step.

Go deeper

- *The bstack: publish once, prove everything, and let users ride free* (Lightweb Inc., Paper 009): <https://1bsv.net/papers/bstack-patterns.pdf>
- *The overlay object plane* (Lightweb Inc., Paper 006): <https://1bsv.net/papers/overlay-object-plane.pdf>
- Pattern 003, borg, which bsecret is built on: <https://1bsv.net/patterns/borg.pdf>
- The pattern series: <https://1bsv.net/patterns.html>

1BSV - BSV Layered Multicast - cash moves fast on 1BSV. · 1bsv.net

Lightweb Inc. · Jeff Harris · jeff@lightweb.net · github.com/lightwebinc

Unbounded Solutions for a Small World™ · © Lightweb Inc.