Setting up 2-of-3 multisig with Sparrow

Coordinator Sparrow Wallet · cosigners BitBox02 (USB) + Trezor Safe 3 (USB) + SeedSigner (QR) · Native Segwit P2WSH · Self-Experiment 4 · belongs to Chapter 7/8

⚠ **The descriptor is part of the backup.** Three seeds without the build plan `wsh(sortedmulti(2,…))` are worthless – nobody knows how they belong together. It is saved in part C, **before** the first address ever sees coins, and later sits with every cosigner backup.

⚠ **Test amount first, then test spends, then the recovery trial.** Real amounts only after that. **No passphrases** on the three cosigners. **Fingerprints, xpubs and addresses only on this sheet** – not in chats, not on photos.

PREPARATION

- ☐ Sparrow is running, server icon green (own node), webcam connected. Trezor Suite and BitBoxApp are **closed** (they otherwise claim the device exclusively)
- ☐ BitBox02 and Trezor Safe 3 on USB, unlocked. SeedSigner booted, **microSD removed**, seed loaded (Seeds → Load a seed → Enter 24-word seed, word suggestions: right-hand buttons scroll up/down, middle one confirms), fingerprint matches the dice log
- ☐ Wallet name (neutral): _____ clock, pen, this sheet

A · CREATE THE WALLET B · IMPORT THE COSIGNERS

1. ☐ **File** → **New Wallet** → name → Create Wallet → Settings: **Policy Type "Multi Signature HD"**, **Script Type "Native Segwit (P2WSH)"**, Cosigners **2 of 3** → three keystore tabs. The derivation `m/48'/0'/0'/2'` only appears after the device import (steps 2–4) – then leave it alone.
2. ☐ **Keystore 1 – BitBox02:** "Connected Hardware Wallet" → "Scan…" → unlock the device → "Import Keystore" → label "BitBox02".
3. ☐ **Keystore 2 – Trezor:** same again, PIN on the device, Sparrow reads out the xpub → label "Trezor". Prompt on the Trezor: _____
4. ☐ **Keystore 3 – SeedSigner:** on the device **Seeds** → **Seed** → **Export Xpub** → **Multisig** → **Native Segwit** → **QR Format "Animated"** → warning → animated QR. Sparrow: "Airgapped Hardware Wallet" → SeedSigner → "Scan" → webcam onto the display. Compare the fingerprint on the SeedSigner with the Sparrow field.
5. ☐ Check: three different fingerprints, three xpubs, all derivations identical → **Apply** → wallet password (encrypts the file only, is **not** a passphrase and does **not** belong in the backup). Settings → the descriptor starts with `wsh(sortedmulti(2,`.

BitBox02 master fingerprint:	**Trezor Safe 3** master fingerprint:	**SeedSigner** master fingerprint:
_____	_____	_____

C · SAVE THE DESCRIPTOR D · CHECK THE ADDRESS ON ALL THREE DEVICES

6. ☐ **File** → **Export Wallet**: format "Sparrow" (wallet file) **and** "Output Descriptor" (text) → USB stick, later a copy with every cosigner backup. No private keys, but every address: private, not secret.
7. ☐ Settings → **Script Policy** → **QR icon** to the right of the descriptor → keep the QR ready (also "Save PDF…").
8. ☐ **BitBox02:** Receive → **"Display Address"** → the first time, the device **registers** the wallet (name, xpubs – do not click it away, compare it with the Settings page) → after that "2-of-3 Bitcoin Multisig" → wallet name → address: compare with Sparrow character by character.
9. ☐ **Trezor:** Receive → "Display Address" with the Trezor keystore → compare. The Trezor only confirms that ITS key is in there, it does not register the cosigners (issue #7118) – which is why steps 8 and 10 are the real protection.
10. ☐ **SeedSigner: Tools** → **Address Explorer** → **Scan wallet descriptor** → QR from step 7 → address list → compare address #0 with Sparrow. 0.8.7: "Descriptor Loaded" (Policy 2 of 3, Signing Keys) → "Receive Address".
11. ☐ The comparison is **Sparrow's Receive tab against the three device displays** (not BitBoxApp/Trezor Suite). Only once **all three** match: label it "Test #0".

E · TEST AMOUNT + TEST SPENDS F · RECOVERY TRIAL

12. ☐ Small amount to address #0, wait for the confirmation from your own node.
13. ☐ **Spend 1 – BitBox + SeedSigner:** Send → "Pay to" (target), Label, Amount → Create Transaction → Finalize Transaction for Signing → Sign → Scan: BitBox over USB (BitBox: wallet, amount/address, "Locktime on block" = block height, normal, RBF, Total, Fee) → "Show QR" → SeedSigner **Scan** → Select Signer → Review Transaction → at **"Your Change" choose "Verify multisig change"**, scan the descriptor QR from step 7, not "Skip" → Sign → QR back → Sparrow "Scan QR" → 2/2 → **Broadcast Transaction**.
14. ☐ **Spend 2 – Trezor + BitBox** (Trezor: "Recipient #1", "Amount #1", "Locktime set to blockheight" = normal, "Total amount", "Sending from"). Every pair once, otherwise the third key stays unproven. Third pair in step 15.
15. ☐ **Recovery trial:** File → **Close Tab** (**delete nothing**) → Import Wallet → "Output Descriptor" → scan the QR from the paper printout (or the file from step 6) → **a new name** (e.g. "Check") → balance, address #0 and fingerprints identical. Then play "BitBox lost": spend with the Trezor only (recognized over USB) + SeedSigner. **Any two seeds plus the descriptor are the wallet, three seeds without the descriptor are a riddle.**
16. ☐ **Fingerprint check from the steel** (as soon as the backups are stamped, then yearly): SeedSigner, microSD removed, **Seeds** → **Load a seed** → **Enter 24-word seed**, words read **from the steel** → compare the fingerprint with the three fields above. Applies to all three seeds, no device has to be wiped. Then cut the power. Steel 1 ☐ Steel 2 ☐ Steel 3 ☐

TIME

Start: _____
Wallet ready (B): _____
Addresses checked (D): _____
First spend / date: _____

STUMBLING BLOCKS / LABELS THAT DIFFERED (FOR THE LOGBOOK)

_____

_____

RESULT

- ☐ **Passed:** all three devices show the same address, all three pairs have spent, the descriptor on its own restores the view of the wallet, three steel fingerprints match. Only now may the wallet carry real amounts – descriptor copies sit with all three backups.
- ☐ **Deviation:** if a device shows a different address or refuses the registration: **deposit nothing**, check fingerprints and xpubs against the Settings page (usually the wrong keystore tab or the wrong derivation), create the wallet again.