SeedSigner – rolling a seed in dice mode

99 rolls → 24 words · menu path SeedSigner OS 0.8.7 · independent verification · Self-Experiment 1b · belongs with dice log sheet 1b

⚠ **The string of digits IS the seed.** SHA-256 turns the 99 digits into the 24 words – whoever knows the digits knows the wallet. The filled-in dice log sheet is as secret as the words.

⚠ **Always treat the first run as practice** with a throwaway seed that never holds real coins afterwards. Only once menu, sequence and verification are familiar do you roll the productive seed. ⚠ **No camera in the room:** phone away, webcam unplugged, mind windows and mirrors.

PREPARATION

- ☐ Dice log sheet 1b is ready (20 rounds of 5 digits = 99, last round 4)
- ☐ Dice setup: cup with a lid (an ice cream tub does the job), throwing surface with a rim (box lid), five dice of the same kind, casino quality
- ☐ SeedSigner on a power bank or mains adapter, around 45 seconds to boot. **Then pull the microSD** – the device runs entirely from RAM and says so itself: "You can remove the SD card now". With no card in it, nothing can be written to it. Permanent: Settings → Persistent settings → Enabled, then Advanced → MicroSD notification duration → Until SD removed.
- ☐ Paper for the fingerprint (8 characters) and for device labels that differ, a clock

MENU PATH ON THE DEVICE (SEEDSIGNER OS 0.8.7)

1 ☐ Main menu → **Tools**

2 ☐ There are **two** entries called "New seed": the first with a **camera icon** (image entropy), the second with a **dice icon**. **Choose the second one.** Easy to mix up – the camera variant is legitimate too, but it cannot be recomputed.

3 ☐ Screen "Mnemonic Length": choose **"24 words (99 rolls)"**, not "12 words (50 rolls)". Label on the device: _____

4 ☐ Screen **"Dice Roll 1/99"**: 3×3 pad with dice pips 1 to 6. For each roll select the number and press, the title counts up (2/99, 3/99 …). Mistyped? The delete key (⌫, bottom right of the pad) takes back the last digit – then reconcile counter and sheet again.

5 ☐ After the 99th digit the device moves on by itself – no confirmation needed.

6 ☐ Warning screen **"Caution – Classified Info!"** (keep seed words private, away from online devices) → **"I understand"**

7 ☐ **"Seed Words 1/6" through "6/6"**, four words per page, "Next" until "Done". With the practice seed write the words down for comparison – with the real seed only onto the backup.

8 ☐ Prompt **"Verify" / "Skip"** → **Verify**: the device asks for all 24 words one at a time, four options each. Play along, this is the first backup test.

9 ☐ The final screen shows the **fingerprint** (8 characters): _____ note it down → "Done". If "BIP-39 Passphrase" appears: do not use it for multisig cosigners.

10 ☐ The seed now lives in RAM only (menu "Seeds"). Power gone = seed gone. For the practice run that is the point.

How the rolling goes: Per round shake five dice, throw, write the digits onto the sheet in a **fixed reading order** (left to right), then tap those same five into the device, **reconcile the device counter with the sheet.** That check every five rolls is the error brake. No roll is repeated because the number is not to your liking. Dice that land tilted or stacked: repeat the whole round, note it on the sheet.

Why 99 rolls: Each roll carries $\log_2 6 \approx 2.58$ bits, 99 rolls around 256 bits – as much as 24 words hold (50 rolls = 12 words). The SeedSigner takes the 99 digits as text, forms SHA-256 over them and derives the words per BIP39. That is exactly why it can be recomputed.

VERIFICATION – NOT HAVING TO BELIEVE THE DEVICE

A ☐ **Your own check script** (in the workshop package: `verify_dice_seed.py`, Python standard library only plus the official BIP39 word list, no SeedSigner code): start `python3 verify_dice_seed.py`, type the 99 digits when asked (that keeps them out of the command history) → compare the 24 words one by one with the display. Run `--selftest` first (checks against the examples in the SeedSigner documentation). The script sends and stores nothing.

B ☐ **Second code base** (optional): Ian Coleman's `bip39-standalone.html` offline → "Show entropy details" → actively click **"Base 10"** (**never "Dice [1-6]"**: it turns a 6 into a 0) → set **"Mnemonic Length" to "24 Words"** (otherwise "Use Raw Entropy" stays active and nothing is hashed) → enter the digits → compare.

! ☐ **What is tested is the method, not the real seed:** if the device computes practice digits correctly, it computes correctly every time – the real seed never leaves the device. If you do recompute it anyway: **offline only** (Tails without network), then shut down. Terminal, history and clipboard see everything.

TIME

Start: _____

Rolling done: _____

Verification done: _____

Date: _____

SNAGS / LABELS THAT DIFFERED (FOR THE LOGBOOK)

_____

_____

_____

RESULT

☐ **Passed:** The 24 words from your own computation are identical with the display – the SeedSigner computed correctly, and the seed does not hang on a manufacturer's randomness.

☐ **Mismatch:** Discard the practice seed, check the digits on the sheet against what you entered (most common cause: a typo between sheet and device), repeat the run. A real seed is only rolled once the practice run goes smoothly.