SeedQR: Hand-Drawn QR Backups for Seed Phrases

October 29, 2026 · Document #412 · bip39-phrase.com



Photo by nuvra on Unsplash

Learn how SeedQR encodes BIP39 seeds, how to draw and verify a backup, and why a camera photograph can expose the entire phrase.

Contents

How Standard SeedQR and CompactSeedQR work 1

Drawing and checking a SeedQR backup 2

A camera can copy the entire secret 3

FAQ 3

Sources 4

BIP39 format examples 4

Reading a BIP39 phrase 4

See also 5

A SeedQR code contains the information needed to reconstruct a BIP39 seed phrase. Drawing it on paper can make loading a wallet into SeedSigner faster, but the drawing must be protected like the written words: anyone who photographs and decodes a readable copy can recover the phrase.

How Standard SeedQR and CompactSeedQR work

SeedSigner supports two SeedQR encodings. Both represent BIP39 mnemonic information without encrypting it. The distinction is the data placed inside the QR code.

Standard SeedQR encodes each word as its four-digit, zero-based position in the BIP39 English word list. The indexes run from `0000` through `2047`. The code concatenates these indexes without

spaces: a 12-word phrase produces 48 decimal digits, while a 24-word phrase produces 96.

The zero-based numbering matters. A printed word list numbered from 1 through 2048 requires conversion before its numbers can be used. The first word, `abandon`, has SeedQR index `0000`. Manually using one-based indexes would encode different words or produce invalid data; a checksum check is not guaranteed to catch the mistake.

Standard SeedQR preserves all the mnemonic's word indexes, including the checksum bits carried by the words. A decoder divides the numeric string into groups of four, looks up the words, and can check the BIP39 checksum. That checksum detects most accidental changes, not all.

CompactSeedQR instead stores the original entropy as raw binary bytes. A 12-word BIP39 phrase represents 128 bits of entropy plus a 4-bit checksum; CompactSeedQR stores the 16 entropy bytes. A 24-word phrase represents 256 bits of entropy plus an 8-bit checksum; its compact encoding stores 32 bytes. These are actual bytes, rather than hexadecimal text.

During decoding, software calculates SHA-256 of the entropy, takes the required checksum bits, appends them, and divides the result into 11-bit word indexes. Using the BIP39 English word list recreates the English phrase. Because the checksum is regenerated, it cannot independently detect a change to the stored entropy.

For SeedSigner's usual 12- and 24-word layouts, Standard SeedQR uses a `25 × 25` or `29 × 29` module grid respectively. CompactSeedQR reduces those sizes to `21 × 21` and `25 × 25`. A module is one black or white square. These dimensions describe the QR symbol itself, excluding its surrounding blank margin.

Neither format includes an optional BIP39 passphrase. Restoring a passphrase-protected wallet requires the same passphrase separately, together with compatible wallet settings.

Drawing and checking a SeedQR backup

Use SeedSigner's SeedQR export workflow and a matching blank grid template. Menu labels can change; the essential process is to display the generated QR modules, copy them, and verify the finished drawing with the device.

1. Generate or load the intended seed on your trusted SeedSigner. Keep phones, webcams, security cameras, and other recording devices away while the words or QR code are visible.
2. Choose Standard SeedQR or CompactSeedQR and the corresponding template. Confirm the grid dimensions against the displayed code. Do not use a template intended for a different symbol size.
3. Copy the displayed pattern onto the grid, using the device's enlarged views where available. Count rows and columns carefully and keep the template's orientation consistent. Fill black modules fully without bleeding into neighboring white squares.
4. Preserve the QR code's quiet zone: a clear white border at least four modules wide on every side. Keep notes, borders, and labels outside that area. Use dark, durable ink with strong contrast against the paper.

5 Scan the completed drawing with SeedSigner and use its verification workflow to confirm that it represents the intended seed. Successful scanning alone only proves that the symbol is readable; the decoded seed must also match.

6 Test recovery before relying on the drawing as your backup. Confirm the expected wallet using its existing fingerprint and receiving-address information, including the same passphrase, script type, derivation path, and account where applicable. A matching fingerprint alone is not conclusive.

QR error correction can tolerate some damage, but its capacity is limited and depends on the encoding settings and damage pattern. It does not replace careful transcription. Faded ink, folds, stains, or a missing margin can make a small hand-drawn code unreadable.

Keep a verified written-word backup as well if your recovery plan needs compatibility with wallets that cannot import SeedQR. Store each copy with the same care: every complete copy creates another opportunity for theft.

A camera can copy the entire secret

SeedQR's convenience also makes copying fast. A phone camera can capture the whole backup in a single image, including from a distance if the modules remain clear enough.

Standard SeedQR may appear in an ordinary QR reader as a long number. CompactSeedQR contains binary data, so some camera applications may show nothing useful or refuse to display it. That behavior provides no protection: suitable decoding software can extract the bytes from a sufficiently clear photograph and reconstruct the words.

Do not scan your real backup with a phone to check it. Camera applications, photo libraries, cloud synchronization, and scanning services can retain or transmit the image. Verify it on the intended offline device instead.

If an unauthorized camera captures the code, treat the seed as potentially compromised. A separate BIP39 passphrase may still protect a wallet, but the mnemonic is no longer secret. Moving funds to a securely generated new seed removes reliance on keeping that photographed backup secret.

FAQ

Is SeedQR encrypted?

No. Both formats are reversible encodings of seed information. Possession of a readable copy can reveal the complete mnemonic.

Can every wallet import CompactSeedQR?

No. The recovery software must support interpreting its binary payload as BIP39 entropy. Check compatibility before depending on it.

Does SeedQR store my BIP39 passphrase?

No. Preserve the exact passphrase separately. A different passphrase after BIP39's Unicode NFKD normalization derives a different seed and wallet; a typo does not produce an incorrect-passphrase warning.

Sources

- SeedSigner official repository and documentation (github.com)
- BIP39: Mnemonic code for generating deterministic keys (github.com)

> Continue reading on bip39-phrase.com: Private Key QR Code Risks: Why Online Generators Are Unsafe
> More on this topic: Seed Phrases

BIP39 format examples

Randomly generated sequences of words from the BIP39 English wordlist. None of them passes the BIP39 checksum, so no wallet will accept them: they show the format only. Never store funds with a phrase you found in a document.

Example 1

1	version	2	retreat	3	share	4	output	5	cushion	6	spot
7	gasp	8	wear	9	cement	10	hip	11	inform	12	kind

Example 2

1	flip	2	charge	3	beef	4	help	5	physical	6	popular
7	online	8	reunion	9	merge	10	sleep	11	rice	12	stable

Example 3

1	neck	2	tongue	3	figure	4	clay	5	hair	6	custom
7	world	8	parent	9	vanish	10	victory	11	special	12	version

Example 4

1	wife	2	system	3	favorite	4	vocal	5	tent	6	depth
7	pond	8	mobile	9	sphere	10	tumble	11	long	12	sun

Example 5

1	online	2	shove	3	vague	4	bus	5	fresh	6	bid
7	balcony	8	spread	9	mistake	10	border	11	orient	12	quote

Example 6

1	hair	2	team	3	front	4	among	5	domain	6	upper
7	rookie	8	addict	9	camp	10	perfect	11	mansion	12	negative

Reading a BIP39 phrase

Every word stands for an 11-bit number: its position in the 2048-word list, counted from 0. Here is Example 1 decoded word by word. Each word links to its page on bip39-phrase.com, where the list is numbered from 1.

#	Word	Index (0-2047)	Word no.	11 bits
1	version	1942	1943	11110010110
2	retreat	1472	1473	10111000000
3	share	1577	1578	11000101001
4	output	1259	1260	10011101011
5	cushion	435	436	00110110011
6	spot	1686	1687	11010010110
7	gasp	769	770	01100000001
8	wear	1987	1988	11111000011
9	cement	297	298	00100101001
10	hip	863	864	01101011111
11	inform	924	925	01110011100
12	kind	980	981	01111010100

The 12 words carry 132 bits: 128 bits of entropy followed by a 4-bit checksum. The checksum must equal the first 4 bits of the SHA-256 hash of the entropy.

`Entropy (hex): f2d70314ceb367a5980fc3252d7dce3d`

This sequence ends in the checksum bits 0100, but SHA-256 of its entropy starts with 1101, so wallets reject it as an invalid phrase. That is the point of the checksum: about 15 of every 16 random or mistyped sequences fail it.

See also

- Your Wallet Vault Inside Phone Cloud Backups · December 22, 2025
- Connect Keystone 3 Pro to Helium Wallet: Step-by-Step · March 25, 2025
- Air-Gapped Signing with PSBT and QR Codes · June 2, 2026

Educational document. The example sequences are random and fail the BIP39 checksum: they are not wallets and must never hold funds. Security notes: bip39-phrase.com/security/