

Mnemonic Seed Formats That Differ From BIP39

October 21, 2026 · Document #404 · [bip39-phrase.com](#)



Photo by [valentin ciccarone](#) on Unsplash

Technical reference detailing non-BIP39 mnemonic phrase formats including Electrum, aezeed, Polyseed, and legacy Monero seeds.

Contents

Electrum Version 2.0+ Seeds	2
Lightning Network Daemon aezeed	2
Polyseed	2
Legacy Monero Mnemonics	3
FAQ	3
Sources	4
Example BIP39 word sequences	4
Reading a BIP39 phrase	5
See also	5

Entering a recovery phrase into wallet software frequently yields an "invalid checksum" error or generates an empty wallet with zero transaction history. This issue occurs because several major cryptocurrency projects intentionally reject or replace the BIP39 standard with proprietary mnemonic specifications. While these phrases use sequences of English words that closely resemble BIP39 mnemonics, their underlying wordlists, checksum algorithms, and derivation mechanisms are incompatible.

Electrum Version 2.0+ Seeds

Electrum discarded BIP39 support in 2014, starting with version 2.0. The developers cited deficiencies in BIP39, notably the absence of internal version numbers and derivation path metadata.

- **Word count:** Typically 12 words, though 24 words can occur.
- **Wordlist:** A 2,048-word English list largely overlapping with BIP39, but containing distinct word positions.
- **Checksum and encoding:** Electrum encodes wallet type and version directly into the phrase itself. It computes an HMAC-SHA512 hash where the key is the string `Seed version` and the message is the UTF-8 normalized phrase (using NFKD normalization without trailing spaces).
- **Verification logic:** The system inspects the prefix of the resulting hexadecimal digest. If the prefix matches `01`, it denotes a standard P2PKH wallet. A prefix of `100` indicates a Native SegWit (P2WPKH) wallet, and `101` indicates a two-factor authentication wallet. If the hash does not match a recognized prefix, Electrum rejects the seed as invalid.

A BIP39 wallet cannot reconstruct addresses from an Electrum phrase because BIP39 derives the master key via PBKDF2 with salt `mnemonic + passphrase`, whereas Electrum uses `electrum + passphrase` and applies different derivation branches.

Lightning Network Daemon aezeed

The Lightning Network Daemon (lnd) introduced the **aezeed** scheme to resolve backup challenges specific to off-chain channels, such as recording the precise wallet creation date (birthday) to speed up chain synchronization.

- **Word count:** Exactly 24 words.
- **Wordlist:** The standard BIP39 2,048-word English list.
- **Structure and checksum:** An aezeed phrase encodes an enciphered payload rather than raw entropy. The data layout contains an internal 5-bit version identifier, a 16-bit birthday timestamp (days elapsed since 2018-03-01), 16 bytes of true entropy, and a cryptographic integrity tag.
- **Verification logic:** The internal payload is encrypted using ChaCha20-Poly1305 with a key stretched through scrypt. If an optional user passphrase was set, it acts as the scrypt salt. Integrity validation occurs during decryption: if the Poly1305 MAC check fails, the phrase or passphrase is incorrect.

Because an aezeed mnemonic uses words from the BIP39 list, a BIP39-only wallet may accept the phrase if the 24 words happen to satisfy the BIP39 checksum equation. However, the derived Bitcoin addresses will not match lnd, and channel states cannot be restored.

Polyseed

Polyseed is a modern mnemonic backup format adopted by privacy coins, including Monero-compatible tools. It replaces older polynomial models with optimized Galois field arithmetic.

- **Word count:** Exactly 16 words.
- **Wordlist:** Supports standard 2,048-word lists, including BIP39 English.

- **Structure and checksum:** Polyseed packs 150 bits of total data into 16 words (each representing 11 bits, totaling 176 bits, leaving 26 redundant bits). The payload stores 128 bits of entropy, a 5-bit coin/feature identifier, a 19-bit birthday timestamp (weeks elapsed since 2021-11-01), and an 11-bit polynomial checksum calculated over GF(2048).
- **Verification logic:** The polynomial code functions similarly to Reed-Solomon encoding. It evaluates the 16 word indices as coefficients of a polynomial. Polyseed can detect up to two substituted words or one erased word without failing silently.

BIP39 does not define a 16-word standard (only 12, 15, 18, 21, and 24 words exist in BIP39). Any 16-word phrase is immediately recognizable as non-BIP39.

Legacy Monero Mnemonics

Legacy Monero (and platforms derived from early CryptoNote implementations) relies on a native format often called an Electrum-style Monero seed, separate from modern Polyseed implementations.

- **Word count:** 25 words (standard CLI/GUI backup) or 13 words (MyMonero format).
- **Wordlist:** A custom Monero wordlist consisting of 1,626 words. Unlike BIP39 words—which are uniquely identifiable by their first four characters—Monero words are uniquely determined by their first three characters.
- **Checksum logic:** In a 25-word seed, the first 24 words encode 256 bits of private spending key material (each group of 3 words maps to 4 bytes of integer data). The 25th word is an explicit checksum: the client runs a standard CRC32 checksum over the 3-letter prefixes of the first 24 words, takes the result modulo 24, and appends the word located at that zero-based index.

Scheme	Word Count	Wordlist Size	Checksum / Validation Algorithm
BIP39	12, 15, 18, 21, 24	2,048	SHA-256 hash prefix (4 to 8 bits)
Electrum 2.0+	12 (or 24)	2,048	HMAC-SHA512 version prefix (01, 100, 101)
aezeed	24	2,048	ChaCha20-Poly1305 MAC via scrypt
Polyseed	16	2,048	11-bit polynomial code over GF(2048)
Legacy Monero	25 (or 13)	1,626	25th word mirrors word at index (CRC32 % 24)

FAQ

How can I tell if a 12-word phrase is BIP39 or Electrum?

Check the checksum using an offline BIP39 tool. If the tool reports an invalid checksum, or if entering it into Electrum without checking the "BIP39" option successfully populates an address, it is an Electrum-native seed.

What happens if I import an aezeed phrase into a standard BIP39 wallet?

If the 24 words pass the BIP39 checksum by mathematical coincidence, the wallet will open but show zero balances and foreign addresses. If the words do not satisfy the BIP39 checksum, the wallet will reject the seed outright.

Can a 25-word Monero phrase be converted into BIP39?

No direct mathematical conversion exists between the two formats. To move assets from a legacy Monero seed to a BIP39-compatible device, generate a new seed on the target wallet and transfer the funds on-chain.

Sources

- [Electrum Seed Versioning Specification](https://electrum.readthedocs.io) (electrum.readthedocs.io)
- [LND aezeed Specification](https://github.com) (github.com)
- [Polyseed Storage Format](https://github.com) (github.com)

Continue reading on bip39-phrase.com: [Monero Seed Phrase: 25 Words, Polyseed and Private Keys](#)
More on this topic: [BIP Standards](#)

Example BIP39 word sequences

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	wire	2	off	3	live	4	fence
5	spider	6	avoid	7	transfer	8	bench
9	father	10	artist	11	garbage	12	lazy

Example 2

1	capital	2	process	3	dignity	4	allow
5	seed	6	share	7	guess	8	must
9	enroll	10	arrive	11	hire	12	fence

Example 3

1	endorse	2	carbon	3	tail	4	unknown
5	level	6	smooth	7	fame	8	captain
9	average	10	leave	11	belt	12	senior

Example 4

1	burden	2	grant	3	mandate	4	metal
5	teach	6	trade	7	surge	8	aunt
9	ethics	10	false	11	hawk	12	youth

Example 5			
1	vote	2	forum
3	since	4	amused
5	parade	6	robust
7	all	8	reveal
9	guess	10	eagle
11	excess	12	pupil

Example 6			
1	panther	2	resemble
3	improve	4	stove
5	industry	6	violin
7	trust	8	path
9	come	10	rather
11	parrot	12	solve

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	wire	2017	2018	11111100001
2	off	1226	1227	10011001010
3	live	1045	1046	10000010101
4	fence	680	681	01010101000
5	spider	1677	1678	11010001101
6	avoid	128	129	00010000000
7	transfer	1849	1850	11100111001
8	bench	168	169	00010101000
9	father	669	670	01010011101
10	artist	104	105	00001101000
11	garbage	764	765	01011111100
12	lazy	1011	1012	01111110011

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): `fc332a0aaa8d1a2039c8a853a1a17e3f`

This sequence ends in the checksum bits 0011, but SHA-256 of its entropy starts with 1011, 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

- [Compiling Policies with rust-miniscript](#) · October 20, 2026
- [Why There Is No BIP39 Version 2](#) · October 16, 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/