Single-Address Wallets Miss HD Change

November 24, 2026 · Document #438 · bip39-phrase.com



Photo by Scott Rodgerson on Unsplash

A single-address restore can miss HD wallet change. Learn why and how to scan the correct receive and internal address chains.

Contents

Why change uses another address 1

How a single-address restore loses the view 2

Restore with software that discovers both chains 2

Check discovery before concluding funds are gone 3

FAQ 3

Sources 3

Example sequences from the BIP39 wordlist 4

Reading a BIP39 phrase 5

See also 5

A wallet that restores only one address for a coin can show an incomplete balance even when the seed phrase is correct. The missing funds are often on change addresses that an HD wallet generated automatically after earlier transactions.

Why change uses another address

Hierarchical deterministic (HD) wallets derive many addresses from one seed. For Bitcoin-style accounts, the receiving chain normally uses branch `0`, while the internal change chain uses branch `1`. A simplified BIP32 path looks like this:

- Receive addresses: `m/44'/0'/0'/0/index`

- Change addresses: `m/44'/0'/0'/1/index`

The exact purpose, coin type, account, and address format can differ. For example, SegWit accounts may use BIP49 or BIP84 paths instead of BIP44. The important detail is the final branch: `0` is normally external or receiving addresses, and `1` is normally internal or change addresses.

When you spend coins, a transaction may need to return value to your wallet. The wallet creates a fresh internal address for that remainder. It signs the transaction with keys derived from the same seed, but the change address is different from the address you gave to the sender. A wallet can therefore hold value at several receive addresses and several change addresses at the same time.

Using a new change address improves privacy and makes address reuse less likely. It also means that a recovery program must derive and scan more than one address sequence.

How a single-address restore loses the view

Some simple wallets track one address per coin. They may import a seed, derive the first receiving address, and display transactions found there. That process can succeed cryptographically while still missing funds held elsewhere.

For example, suppose an HD wallet received coins at a receiving address, spent part of them, and sent the remainder to a fresh internal address. A single-address wallet may find the first incoming transaction but never inspect the internal address that received the change. Its displayed balance can then be lower than the spendable balance controlled by the seed.

The seed has not changed, and the funds have not moved to a different wallet. The restored software is following a narrower discovery rule. It may also miss later receiving addresses if it checks only the first address rather than deriving a sequence.

A different derivation path can create the same symptom. A wallet restoring a BIP44 account may not find funds created under BIP84, even though both accounts came from the same BIP39 seed. Address format is a clue: legacy Bitcoin addresses usually begin with `1`, wrapped SegWit addresses with `3`, and native SegWit addresses with `bc1q`. These prefixes help identify an account type, but they do not prove the correct derivation path.

Restore with software that discovers both chains

Use wallet software that supports the relevant HD standard and scans both external and internal chains. The recovery process should derive candidate accounts, inspect receiving branch `0` and change branch `1`, and continue through unused addresses according to its discovery or gap-limit rules.

1. Identify the original wallet’s coin, address format, account number, and derivation path if possible. Check its documented recovery procedure or account export information.
2. Install recovery software from its official distribution and verify that it supports BIP39 or the seed format used by the original wallet.
3. Restore the seed in an offline or carefully controlled environment. Never type a recovery phrase into a website, chat, form, or untrusted application.
4. Select the matching account type and derivation path. If the original path is unknown, inspect the standard paths used by that wallet and compare derived addresses with addresses from old records.

5 Let the software scan both receiving and change chains. It should continue scanning when it finds activity and use an appropriate gap limit for runs of unused addresses.

6 Compare the derived addresses with transaction history from a reliable block explorer or your own node. Confirm that incoming payments, spending transactions, and change outputs match.

7 After recovery, create a small test transaction and confirm that the wallet signs it and selects the expected funds. Move funds only after confirming that the restored wallet controls the relevant addresses.

A watch-only restore can help with discovery. Importing an account-level extended public key, such as an `xpub`, `ypub`, or `zpub` where supported, lets software scan addresses without exposing private keys. It cannot spend funds. To recover control, the wallet must use the seed or the corresponding private key material.

Check discovery before concluding funds are gone

If the balance is unexpectedly low, record the addresses and paths already checked. Then test the likely account types and scan both branches. Look for an address receiving an output labelled as change in the original transaction history. Wallets may label change differently, so inspect the transaction outputs and address ownership rather than relying only on the interface.

A correct seed can restore multiple accounts. BIP44 account discovery normally starts with account `0` and should not skip an account that contains funds. Some software stops after a fixed number of unused addresses; this is the gap limit. A wallet that used many addresses without activity between them may require a larger scan range or a manual recovery procedure.

Do not “repair” the seed by changing words or adding a passphrase. A BIP39 passphrase creates a separate wallet, and even one character entered differently produces different keys. Check the word list, order, spelling, and passphrase exactly, then verify the derivation path and both address branches.

FAQ

Can a change address have a different private key from the seed?

Yes. It has its own derived private key, but that key is deterministically generated from the seed and derivation path.

Does scanning only receive addresses destroy the missing funds?

No. It only fails to discover them. Software that derives and scans the correct change chain can find the same wallet-controlled outputs.

What should I do if the original path is unknown?

Use the original wallet’s documented path first, then compare supported account types and scan both branches while matching derived addresses to known transactions.

Sources

- BIP-32: Hierarchical Deterministic Wallets (github.com)
- BIP-39: Mnemonic code for generating deterministic keys (github.com)
- BIP-44: Multi-Account Hierarchy for Deterministic Wallets (github.com)

> Continue reading on bip39-phrase.com: BIP39 vs BIP32 vs BIP44: How the Seed Standards Fit Together
> More on this topic: Recovery

Example sequences from the BIP39 wordlist

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	flip	2	firm	3	student	4	marine	5	gospel	6	solar
7	party	8	brother	9	mystery	10	maid	11	place	12	shed

Example 2

1	love	2	mercy	3	dice	4	donor	5	spend	6	wood
7	century	8	arctic	9	apple	10	tennis	11	pitch	12	tower

Example 3

1	umbrella	2	foot	3	mad	4	car	5	glimpse	6	race
7	bottom	8	firm	9	cattle	10	system	11	work	12	vanish

Example 4

1	purchase	2	champion	3	coffee	4	guitar	5	badge	6	decorate
7	bunker	8	praise	9	report	10	mutual	11	pumpkin	12	apple

Example 5

1	mountain	2	gold	3	exclude	4	swift	5	fine	6	box
7	camp	8	bike	9	celery	10	echo	11	gold	12	clarify

Example 6

1	congress	2	solar	3	coil	4	finish	5	genius	6	sword
7	slush	8	unusual	9	urban	10	park	11	logic	12	universe

Example 7

1	task	2	carpet	3	betray	4	forum	5	rebuild	6	autumn
7	goat	8	cotton	9	hen	10	eyebrow	11	total	12	razor

Example 8

1	light	2	ketchup	3	march	4	elephant	5	mixed	6	snap
7	renew	8	twelve	9	shock	10	love	11	lunch	12	thrive

Example 9												
1	orbit	2	inner	3	obscure	4	oppose	5	arctic	6	animal
7	beauty	8	broom	9	elephant	10	apology	11	must	12	subway

Example 10												
1	cricket	2	assist	3	organ	4	vicious	5	game	6	gown
7	bronze	8	toast	9	glance	10	marriage	11	meat	12	organ

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	flip	712	713	01011001000
2	firm	698	699	01010111010
3	student	1723	1724	11010111011
4	marine	1088	1089	10001000000
5	gospel	806	807	01100100110
6	solar	1651	1652	11001110011
7	party	1284	1285	10100000100
8	brother	231	232	00011100111
9	mystery	1171	1172	10010010011
10	maid	1072	1073	10000110000
11	place	1326	1327	10100101110
12	shed	1578	1579	11000101010

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): 590aeb5dc4064d9ce820e79270c29762
```

This sequence ends in the checksum bits 1010, 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

- How BIP44 Account Discovery Decides When to Stop · June 26, 2025
- Recovering Funds Sent to Contract Wallets on Other Chains · April 23, 2026
- Coins Sent to Mistyped or Wrongly Restored Wallets · October 15, 2026
- Funds in a Second or Third Account Did Not Return · November 21, 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/