

Watch-Only Wallets: Tracking Funds Without Private Keys

October 18, 2026 · Document #355 · bip39-phrasedotcom



Photo by [Kyle Williamson](#) on [Unsplash](#)

A technical explanation of how watch-only wallets derive addresses, prepare PSBTs using extended public keys, and manage privacy risks.

Contents

Derivation Mechanics with Extended Public Keys	2
Extended Key Types and Descriptors	2
Transaction Construction and PSBTs	3
Information Exposure and Privacy Risks	3
FAQ	3
Sources	4
Sample sequences of BIP39 words	4
Reading a BIP39 phrase	5
See also	6

A **watch-only wallet** reconstructs an account's complete address list and transaction history using public data alone, operating with no access to the private keys required to spend funds. In Hierarchical Deterministic (HD) wallets governed by BIP32, this functionality is driven by an **extended public key** (such as an xpub, ypub, or zpub) or output script descriptors. By decoupling balance verification and transaction creation from signing authority, watch-only setups allow network-connected devices to monitor balances safely while physical private keys remain isolated on cold storage or air-gapped hardware.

Derivation Mechanics with Extended Public Keys

Under BIP32, an extended key consists of a standard public or private key combined with a 32-byte entropy value known as a chain code. The chain code prevents derivation of child keys using only the public key point, requiring mathematical interaction with elliptic curve points.

When a child key uses non-hardened derivation (indices 0 through $2^{31} - 1$), child public keys can be derived directly from the parent public key and parent chain code without needing the parent private key. The process calculates an HMAC-SHA512 hash using the parent chain code as the key and the serialized parent public key concatenated with the child index as the payload. The leftmost 256 bits of the hash result are interpreted as a scalar integer, multiplied by the elliptic curve base point G , and added to the parent public key point on the secp256k1 curve. In the rare event that the scalar exceeds the curve order or the point at infinity is produced, BIP32 specifies that the index is invalid and derivation proceeds to the next index. The rightmost 256 bits become the child chain code.

A watch-only software wallet imports this root extended public key at a specific account level—such as `m/84'/0'/0'` for native SegWit. From this point, the software calculates child keys along two branches:

- External chain (`/0/*`): Generates sequential receiving addresses shown to counterparties.
- Internal chain (`/1/*`): Generates sequential change addresses returned to the wallet when constructing outgoing payments.

To detect incoming funds, the watch-only wallet queries a node or indexing server across sequentially derived addresses until it encounters a run of unused addresses reaching the **gap limit** (standardized at 20 consecutive unused addresses in BIP44).

Extended Key Types and Descriptors

Extended public keys are represented either through serialized Base58Check prefixes denoting target script types or through modern output script descriptors defined in BIP380 through BIP389.

Format Prefix	BIP Standard	Script Type	Address Format Example
xpub	BIP44	Legacy P2PKH	Base58 starting with 1
ypub	BIP49	Nested SegWit P2SH-P2WPKH	Base58 starting with 3
zpub	BIP84	Native SegWit P2WPKH	Bech32 starting with bc1q
Descriptors	BIP380+	Explicit Script (e.g., <code>wpkh()</code> , <code>tr()</code>)	Context-dependent (e.g., Taproot <code>bc1p</code>)

Descriptors supersede raw extended public keys by explicitly defining the derivation path, origin fingerprint, and script wrapper in a single string, such as `wpkh([d34db33f/84'/0'/0']xpub.../0/*)`. This prevents software from misinterpreting the intended spending script.

Transaction Construction and PSBTs

Although a watch-only wallet cannot authorize a withdrawal, it possesses complete visibility into the unspent transaction output (UTXO) set belonging to its addresses. This allows the watch-only instance to prepare spending transactions up to the signing stage using BIP174 or BIP370, known as a **Partially Signed Bitcoin Transaction** (PSBT).

- 1 UTXO Selection: The watch-only client selects available UTXOs to cover the output amount and estimated network fee.
- 2 Transaction Assembly: The client builds the transaction skeleton, defining the inputs, destination outputs, and newly derived internal change address.
- 3 PSBT Export: The unsigned transaction, containing input derivation paths, previous transaction hashes, output indices, and UTXO script data, is exported via a file, QR code, or microSD card.
- 4 External Authorization: The PSBT is transferred to an isolated signing device holding the seed phrase or private keys. The hardware verifies the details and generates signatures.
- 5 Broadcast: The fully signed PSBT is loaded back into the watch-only wallet, which finalizes the raw network transaction and broadcasts it to the peer-to-peer network.

Information Exposure and Privacy Risks

Deploying a watch-only wallet eliminates the risk of private key theft directly from that host, but it does not completely eliminate transaction-tampering risks. A compromised host could present modified recipient addresses or fraudulent change destinations in generated PSBTs, making on-device verification at the signing stage essential. Extended key usage also introduces distinct privacy and security tradeoffs:

- Total Balance and History Visibility: Anyone with access to an extended public key can derive every current, past, and future address within that account path. They can audit all transaction amounts, counterparty patterns, and account balances indefinitely.
- Chain Code Vulnerability: If non-hardened derivation is used and an attacker obtains both the parent xpub and a single child non-hardened private key (for instance, exported from an isolated system to recover an individual token), the attacker can calculate the parent private key mathematically. This compromises the entire wallet branch.
- Network Leakage: If a watch-only wallet queries third-party public electrum servers or hosted API nodes without routing through a personal full node or encrypted proxy, all derived addresses are linked to the user's IP address on external servers.

FAQ

Can a watch-only wallet spend funds directly?

No. A watch-only wallet contains only public keys and chain codes, lacking the private key material necessary to generate valid cryptographic signatures for transaction inputs.

Does sharing an extended public key compromise a seed phrase?

An extended public key does not expose the master private key or seed phrase directly, but if any unhardened child private key derived from it is compromised, the parent private key can be derived mathematically.

What happens if transactions occur past the gap limit?

A watch-only wallet typically fails to detect incoming transactions if more consecutive unused addresses than the gap limit (standardized at 20 in BIP44) are skipped, requiring manual adjustment of the scan window or address generation in the wallet software.

Sources

- [BIP 32: Hierarchical Deterministic Wallets \(github.com\)](#)
- [BIP 174: Partially Signed Bitcoin Transaction Format \(github.com\)](#)
- [BIP 380: Output Script Descriptors General Operation \(github.com\)](#)

Continue reading on bip39-phrases.com: [xpub, ypub and zpub: Extended Public Keys Explained](#)

More on this topic: [Wallets](#)

Sample sequences of BIP39 words

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 train	2 planet	3 dial	4 soap
5 quarter	6 tent	7 shift	8 float
9 grid	10 weird	11 exile	12 magnet

Example 2			
1 bulk	2 grunt	3 festival	4 clap
5 stock	6 false	7 nominee	8 echo
9 supply	10 large	11 claim	12 item

Example 3			
1 razor	2 blast	3 rocket	4 what
5 master	6 fancy	7 risk	8 helmet
9 toe	10 point	11 evidence	12 polar

Example 4			
1 uphold	2 vanish	3 wood	4 firm
5 tribe	6 unable	7 improve	8 slice
9 excuse	10 flip	11 help	12 mobile

Example 5

1	portion	2	kit	3	snow	4	once
5	display	6	heart	7	shove	8	aim
9	all	10	raven	11	connect	12	oak

Example 6

1	open	2	near	3	rain	4	include
5	setup	6	enemy	7	change	8	expand
9	decorate	10	hunt	11	fresh	12	local

Example 7

1	include	2	inject	3	diagram	4	ripple
5	latin	6	swap	7	blind	8	trust
9	proud	10	unknown	11	mystery	12	chief

Example 8

1	guitar	2	short	3	paper	4	inject
5	sound	6	believe	7	already	8	select
9	easily	10	title	11	ladder	12	rely

Example 9

1	danger	2	banner	3	hole	4	tilt
5	burst	6	bulb	7	sail	8	goose
9	until	10	circle	11	tattoo	12	sea

Example 10

1	cause	2	notice	3	pen	4	demise
5	jump	6	swarm	7	elbow	8	stable
9	over	10	question	11	record	12	lock

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	train	1848	1849	11100111000
2	planet	1327	1328	10100101111
3	dial	488	489	001111101000
4	soap	1645	1646	11001101101
5	quarter	1403	1404	10101111011

#	Word	Index (0-2047)	Word no.	11 bits
6	tent	1786	1787	11011111010
7	shift	1582	1583	11000101110
8	float	713	714	01011001001
9	grid	819	820	01100110011
10	weird	1993	1994	11111001001
11	exile	637	638	01001111101
12	magnet	1071	1072	10000101111

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): e714bcf466daf7beb172c9667f253ec2

This sequence ends in the checksum bits 1111, 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

- [Mnemonic Keys to the Kingdom: Why BIP-39 Still Matters in 2025](#) · matters-in_2025-june
- [Building a Crypto Inheritance Plan: Safeguarding Your Digital Assets](#) · June 14, 2026
- [Cryptocurrency Security Best Practices: Protecting Your Digital Assets](#) · April 19, 2026
- [Why You Need a Hardware Wallet to Protect Your Crypto](#) · April 22, 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/](#)