

The Second Key: How PQ Wallet 1.5.0 Moves Coins If BTX Ever Switches Off ML-DSA

BTX locks every coin behind two signature schemes so that a break in one cannot freeze the money. That promise only holds if the wallets people actually use can sign with the second scheme. PQ Wallet 1.5.0 now can. This article explains the mechanism from the address up, measures what it costs, says plainly what it protects against and what it does not, and records the first recovery-key spend on mainnet.

OCTOBER 5, 2026 · 18 MIN READ · [EASYBTX.COM/RESEARCH/PQ-WALLET-SECOND-KEY](https://easybtx.com/research/pq-wallet-second-key)

ABSTRACT

Every BTX coin is locked behind two post-quantum keys, but until October 2026 PQ Wallet could only sign with the first one. This is how the recovery key works, why BTX can switch its everyday signature scheme off, what that switch really is in the node's code, what a recovery spend costs and reveals, how we proved it on regtest and then on mainnet, and what is still open. Every number was measured.

Written on 5 October 2026, the morning after the first recovery-key spend on the BTX mainnet. easyBTX builds PQ Wallet and the qID library it signs with; we are not the BTX core team. Every figure below was measured during the PQ Wallet 1.5.0 release round against the BTX reference node v0.34.12, on a private test network (regtest) and on mainnet, and the transaction ids are given so you can check them yourself.

The one-page version

Two keys behind every coin. A BTX output is not locked to one public key. It is locked to the root of a small Merkle tree whose leaves are spending conditions. PQ Wallet builds every address with two leaves: an everyday leaf checked with **ML-DSA-44** (FIPS 204) and a recovery leaf checked with **SLH-DSA-SHAKE-128s** (FIPS 205). Either leaf alone spends the coin.

Until now PQ Wallet could not use the second leaf. The recovery leaf has been in every PQ Wallet address since the first version, but the wallet could only sign with the everyday key. (The BTX reference node's own wallet has SLH-DSA signing code; most people do not run it.) If the everyday scheme ever had to be switched off, a PQ Wallet user's coins would have been safe and stuck. PQ Wallet 1.5.0 closes that gap.

Why two schemes. ML-DSA is small and fast and rests on lattice problems. SLH-DSA is large and slow and rests only on the security of a hash function, the most conservative assumption in cryptography. They fail independently. If lattices fell, hash-based signatures would still stand.

The switch. BTX's consensus code has a parameter, `nMLDSADisableHeight`, that can make ML-DSA signatures invalid from a chosen block. On mainnet it is set to "never". Changing it takes a new node release and the network's adoption. It exists so that a broken scheme can be turned off; the recovery leaf exists so that turning it off does not freeze anyone's money.

What it costs. A recovery spend is about twice the size of a normal one and is priced at exactly 10,000 vB per coin: 0.0001 BTX per coin at the minimum fee.

Proven. On regtest: moves of 5, 100 and 150 coins confirmed, and a full rehearsal of the switch-off. On mainnet, 5 October 2026: one coin moved with the recovery key, transaction `50d246dd...fe90d29`.

Not yet. No outside audit of the new signing code. And the recovery key is no defence against a stolen master key, because both keys come from it.

1. What a BTX address really is

Bitcoin addresses commit to a public key, or to a script, or (since Taproot) to a key plus a tree of scripts. BTX removed elliptic-curve signatures from consensus and kept only one output type for ordinary coins, **P2MR** (pay to Merkle root): witness version 2, followed by a 32-byte Merkle root. Consensus rejects any non-data output that is not P2MR, so there is no classical address type on the chain at all.

The Merkle root commits to a set of **leaves**, and each leaf is a short script with its own public key and signature check. PQ Wallet, through the qID library, builds every address from exactly two leaves:

Leaf	Signature scheme	Public key	Signature	Hardness assumption
Everyday ("login")	ML-DSA-44, FIPS 204	1,312 bytes	2,420 bytes	Module lattice problems (Module-LWE / Module-SIS)
Recovery	SLH-DSA-SHAKE-128s, FIPS 205	32 bytes	7,856 bytes	Second-preimage and related properties of SHAKE-256

To spend, the wallet reveals **one** leaf, a signature that satisfies it, and a 33-byte control block proving the leaf is in the tree. The other leaf is never revealed: only its hash appears, inside the control block. That is why a recovery spend says nothing about the everyday key, and a normal spend says nothing about the recovery key.

You can see the two shapes in any transaction's witness. A normal PQ Wallet spend carries three items of **2,420, 1,316 and 33 bytes**: the ML-DSA signature, the everyday leaf (a 1,312-byte public key plus four bytes of script), and the control block. The first mainnet recovery spend carries **7,856, 34 and 33 bytes**: the SLH-DSA signature, the recovery leaf (a 32-byte public key plus two bytes of script), and the control block.

Where the keys come from

Both keys are derived from the wallet's 64-character master key, through two separate derivations (`deriveLoginKey` and `deriveRecoveryKey` in qID) at the same index. They are different key pairs

for different algorithms, and neither can be computed from the other. But the master key produces both. This has a consequence people get wrong, so it is worth saying in its own sentence: **the recovery key is a defence against a broken algorithm, not against a stolen master key.**

2. Why BTX can switch its everyday scheme off

Two foundations

ML-DSA and SLH-DSA are both NIST post-quantum standards (finalized 13 August 2024), and neither is known to be broken, classically or by a quantum computer. They differ in what they rest on.

ML-DSA rests on the hardness of structured lattice problems. The research behind it is serious and decades old, and NIST chose it as its primary signature standard. But lattice cryptography is younger and more structured than hash functions, and structure is where attacks tend to come from. History has examples: other post-quantum candidates that reached late rounds of standardization, such as Rainbow and SIKE, were broken outright during the process.

SLH-DSA rests only on the security of a hash function. If SHAKE-256 failed badly enough to break SLH-DSA, a great deal of other cryptography would fail with it. It is the most conservative signature scheme standardized today. It pays for that with size and speed: a signature is more than three times larger, and signing is much slower.

BTX uses ML-DSA for everyday spending because it is cheap, and keeps SLH-DSA in every output as an insurance policy that does not depend on the same mathematics.

The switch, in the node's own code

The reference node (v0.34.12, the current release) has a consensus parameter for this:

- `nMLDSADisableHeight` in the consensus parameters. From that block height on, ML-DSA signature checks fail, so no ML-DSA spend can be mined.
- On mainnet it is set to the largest possible 32-bit value, which means **never** (`src/kernel/chainparams.cpp`).
- Nodes stop relaying ML-DSA transactions **960 blocks before** the switch-off height (`MLDSA_EMERGENCY_RELAY_WINDOW_BLOCKS` in `src/validation.cpp`), about one day at the 90-second block target. That gives the network a quiet period in which nothing that is about to become invalid is accepted into mempools.
- The command-line option `-mldsadisableheight` exists, but the node's own help text marks it **regtest-only**. It cannot be used to switch ML-DSA off on mainnet.

So there is no administrator, no key and no button. Switching ML-DSA off on mainnet would mean a developer setting a height in the code, a node release carrying it, and miners and node operators installing that release. Technically it is a soft fork: the rules get stricter, and old nodes would follow along without being able to tell. It would be public, discussed, and announced with lead time, which is exactly what wallets need to move coins.

What the switch is for

If ML-DSA were ever broken in practice, an attacker could forge spends of any coin whose everyday public key they could see. Every address that has ever been spent from has revealed its everyday leaf, and any coins still sitting on that address are locked to the same key. Switching the scheme off would stop such forgeries. Without a working recovery path, it would also stop every honest owner. The recovery leaf is what makes the switch usable as a safety measure rather than a disaster.

3. What PQ Wallet 1.5.0 does

Move coins with the recovery key

Settings > Backup > **Move coins with the recovery key** takes every coin in the wallet and sends it, signed with the recovery key, to another wallet of yours, or to an address you type and confirm a second time. The review shows what you will receive, the network fee, how many coins move and how long signing will take. Nothing is sent until every coin is signed and checked.

Signing runs off the main window on up to four Web Workers at once, with a progress bar and a Cancel button. If the network answers that it already has the transaction, that counts as success. If a send has to be retried, the wallet resends **the same signed bytes** it kept, never a fresh signature. That matters more than it sounds: ML-DSA and SLH-DSA signatures are randomized in this implementation, so signing the same transaction twice gives the same transaction id with a different witness, which a node that already holds the first copy refuses (`txn-same-nonwitness-data-in-mempool`). We hit this in testing before shipping and built the wallet around it.

One move signs at most 100 coins. A wallet with more is moved in several rounds, and coins worth less than their own move fee are left where they are and listed, never silently dropped.

Send knows when the everyday key is off

If the network ever refuses a normal send because ML-DSA is switched off, the Send tab does not show a raw node error. It says: *"The BTX network has switched off this wallet's everyday key, so this send cannot go through. Your coins are safe. Use Settings > Backup > Move coins with the recovery key instead,"* with a link that takes you there.

Why it is on the Backup page

The recovery key is not a second account and not something to use day to day. It is the plan for the day the everyday key cannot be used. It lives next to the master-key backup because both are about the same question: how your coins stay yours when something goes wrong.

4. What a recovery spend costs

A BTX node prices a transaction by its **virtual size**: the larger of its byte size and its signature-operation cost multiplied by 20. The reference source gives SLH-DSA a weight of 500 signature operations per signature (`VALIDATION_WEIGHT_PER_SLHDSA_SIGOP` in `src/script/script.h`), against 50 for ML-DSA. For SLH-DSA that is **$500 \times 20 = 10,000$ vB per input**, more than its actual size

of about 8,000 bytes. For ML-DSA the sigop cost (1,000 vB) is far below its real size, so a normal spend is priced by its bytes.

	Normal spend (ML-DSA)	Recovery spend (SLH-DSA)
One coin in, one out, bytes	3,873	8,025
Priced at	its size	10,000 vB per coin
Fee at 1 sat/vB	about 0.00004 BTX	0.0001 BTX
Signing one coin, measured	well under a second	about 3 seconds (3.13 s, measured on an Apple M5)

Measured on regtest against the node: a 150-coin recovery spend came to 1,195,555 bytes, just under the 1,200,000-byte standard transaction limit, and the node priced it at exactly 1,500,000 vB, the same number qID's estimator predicts. qID refuses 151 coins before signing anything, because the result could not be relayed. The wallet caps a move at 100 coins to leave room.

The first mainnet recovery spend paid **10,000 sat, exactly 0.0001 BTX, for one coin**: the price this section derives.

5. How it was proven

On regtest

Regtest is a private BTX network that we run on one machine, with the same reference node and the same consensus code, where we can mine blocks on demand and move test coins without touching real money. We ran the **actual PQ Wallet application**, not a test harness, against it:

- **Recovery moves of 5 and 100 coins**, both confirmed. With the window hidden, the 5-coin move signed in 42 seconds and the 100-coin move in 8 minutes 50 seconds.
- **The switch-off rehearsal**. We restarted the node with ML-DSA disabled a few blocks ahead. The wallet's ordinary send was refused and showed the switch-off message above. The recovery move from the same wallet then went through and confirmed.
- **qID's own suites** against the node: a recovery spend mined past the ML-DSA disable height, a 100-coin spend, a 150-coin spend at the size limit, and the 151-coin refusal.

On mainnet

On 5 October 2026, using the PQ Wallet 1.5.0 build that is being released, one coin was moved with the recovery key:

Transaction	50d246dd6325e455070e79c7bc58393644db11fc9078b95c8683d0611fe90d29
Input	one coin, 0.00243078 BTX
Output	0.00233078 BTX to another wallet of the same owner
Fee	0.0001 BTX
Size	8,025 bytes

Witness	7,856 + 34 + 33 bytes: an SLH-DSA signature, the recovery leaf, the control block
---------	---

You can open it in any BTX explorer and check the witness sizes yourself. A 7,856-byte first witness item is an SLH-DSA-SHAKE-128s signature; an everyday spend would show 2,420.

6. The rest of 1.5.0, in brief

The same release fixed how wallets that receive mining payouts behave, because those are the wallets with the most coins.

- **Every coin is spendable.** A miner's wallet holds one coin per payout, often thousands. The wallet checks each coin's transaction before spending it, so a coin that carries a BTX artifact is never spent as plain BTX by accident. It used to check only about 600, and every coin past that was set aside. It now checks all of them, and remembers transactions proven to carry no artifact while the wallet is open.
- **Block rewards wait until they can move.** A coin from a block you mined yourself cannot be spent until it is 100 blocks old, and the network refuses a whole transaction that includes a younger one. Measured on regtest: a reward from block 224 was refused at height 322 and accepted at 323. The wallet now leaves such coins out and says how many blocks are left.
- **The artifact check no longer trusts a shortened answer.** Before spending a coin, the wallet checks that the server's description of that coin's output really pays this wallet its exact amount. A server that left an artifact's marker out of a transaction could otherwise make the artifact's coin look like plain BTX.
- **Several sends in a row.** Measured on mainnet with this build: six sends from one wallet with no block in between, all confirmed.
- **The cryptography libraries were updated** (@noble/post-quantum 0.7.1), and the new core was checked against the previous one: 1,990 comparisons of keys, addresses and signatures, all identical.

The easyBTX miner's built-in wallet received the same mining-wallet fixes in easyBTX 0.30.0, including splitting a very large "Max" send into several transactions of 300 coins.

7. What the recovery key does not do, and what is still open

It does not protect against theft. Both keys come from the master key. Anyone with the master key can sign with either. The single most effective protection for most users is unrelated to post-quantum cryptography: **put a passphrase on the wallet**, so the key is not stored in the clear on the device. PQ Wallet warns about every unencrypted wallet that holds coins, under the balance.

It is slower and dearer. Thousands of coins take a long time to sign and cost 0.0001 BTX each to move. The recovery move is an emergency exit, not a payment method.

The new signing code has no outside audit yet. About 200 lines were added to qID for recovery-leaf spending. They went through repeated internal adversarial reviews, a byte-for-byte

equivalence check against the previous core, and every test on a real BTX node, including mainnet. That is diligence, not independent assurance. The bonuz mobile wallet, which also uses qID, will not adopt the recovery feature until an outside audit is done.

The switch has never been used. Everything here was rehearsed on regtest with the reference node's real consensus code. A real switch-off on mainnet would be a coordinated network event, and its exact timing would depend on the release that schedules it.

Signing is slower with the window hidden on macOS. The review's time estimate assumes full speed; with the window in the background it reads low (about 3 minutes estimated, 8 minutes 50 seconds measured for 100 coins). Keeping the window in front helps.

8. How to check this yourself

- Open transaction `50d246dd6325e455070e79c7bc58393644db11fc9078b95c8683d0611fe90d29` in a BTX explorer and look at the witness of its input: 7,856, 34 and 33 bytes.
- Compare it with any ordinary BTX transaction: 2,420, 1,316 and 33.
- In the BTX reference source (v0.34.12): `nMLDSADisableHeight` in `src/kernel/chainparams.cpp` (mainnet: maximum value, never), `MLDSA_EMERGENCY_RELAY_WINDOW_BLOCKS` (960) in `src/validation.cpp`, and `VALIDATION_WEIGHT_PER_SLHDSA_SIGOP` (500) in `src/script/script.h`.
- In PQ Wallet 1.5.0: Settings > Backup > Move coins with the recovery key. The review shows the fee per coin before anything is signed.

Glossary

ML-DSA-44. Module-Lattice-Based Digital Signature Algorithm, NIST FIPS 204, the smallest parameter set. BTX's everyday signature.

SLH-DSA-SHAKE-128s. Stateless Hash-Based Digital Signature Algorithm, NIST FIPS 205, small-signature parameter set built on SHAKE-256. BTX's backup signature.

P2MR. Pay to Merkle root. BTX's only output type for ordinary coins: a commitment to a tree of spending leaves.

Leaf. One spending condition in a P2MR tree. Spending reveals one leaf and proves it belongs to the tree.

Control block. The 33 bytes in a spend that prove the revealed leaf is part of the committed tree, containing the hash of the leaf that is not revealed.

Virtual size (vB). The size the node charges fees on: the larger of a transaction's bytes and its signature-operation cost times 20.

Soft fork. A consensus change that makes the rules stricter. Blocks valid under the new rules remain valid to old nodes.

Regtest. A private test network run by the reference node with the same consensus code, where blocks can be mined on demand.

Frequently asked questions

What is the recovery key?

Every address in PQ Wallet commits to two spending conditions. The everyday one is signed with ML-DSA-44 (FIPS 204, lattice-based) and is what every normal send uses. The second is signed with SLH-DSA-SHAKE-128s (FIPS 205, hash-based). Either one alone can spend the coin. PQ Wallet 1.5.0 can now sign with the second one, under Settings > Backup > Move coins with the recovery key.

Is it a different private key? Does using it leak anything?

It is a different key pair with a different algorithm, but both are derived from the same 64-character master key. Using the recovery key reveals the recovery public key and an SLH-DSA signature, nothing about the everyday key and nothing about the master key. It does not protect against theft: whoever has the master key has both keys.

Why would BTX ever switch ML-DSA off?

ML-DSA rests on lattice problems, a younger foundation than hash functions. Nobody expects a break, but if a practical attack or a serious implementation flaw appeared, continuing to accept ML-DSA signatures would let an attacker forge spends. Switching it off stops that, and the hash-based recovery key keeps every coin movable.

Who can flip the switch?

Nobody, on its own. The switch is a consensus parameter, `nMLDSADisableHeight`, which is set to 'never' for mainnet in the reference node (v0.34.12). The command-line override exists for regtest only. Switching ML-DSA off on mainnet would take a new node release that sets a height, adopted by the miners and nodes that enforce it: a public soft fork with lead time, not a button.

What does a recovery spend cost?

The node prices an SLH-DSA signature at 500 signature operations, which at 20 virtual bytes each is exactly 10,000 vB per coin. At the minimum fee rate of 1 sat/vB that is 0.0001 BTX per coin moved. The first mainnet recovery spend paid exactly that.

Has it been used on mainnet?

Yes. On 5 October 2026 a PQ Wallet 1.5.0 user moved a coin with the recovery key: transaction `50d246dd6325e455070e79c7bc58393644db11fc9078b95c8683d0611fe90d29`, 8,025 bytes, fee 0.0001 BTX, carrying a 7,856-byte SLH-DSA signature.

Has the new code been audited?

Internally and adversarially, repeatedly, and it passed every test on a real BTX 0.34.12 node. It has not been audited by an outside firm yet. That is the main open item, and the bonuz mobile wallet will not adopt the feature before it is done.

Who wrote this?

It was researched and written by Claude, an AI model, under our direction, from measurements made during the 1.5.0 release round and from the BTX reference source, and reviewed before publication. easyBTX builds PQ Wallet and qID, so this is a participant describing its own work.

Published by easyBTX, which builds software for BTX and is not the BTX core team. This paper is educational research, not financial advice, and it is not an official BTX document. Read the live version and check the sources at easybtx.com/research/pq-wallet-second-key.