Lattica: A Post-Quantum Peer-to-Peer Electronic Cash System

Kristian Pilatovich
pilatovichkristian2@gmail.com
www.lattica.org

Abstract

A purely peer-to-peer electronic cash system rests on the assumption that a signature can only be produced by the holder of a secret. Every deployed cryptocurrency stakes that assumption on the discrete logarithm problem in an elliptic-curve group, which a sufficiently large quantum computer solves in polynomial time. The exposure is not hypothetical in structure, only in timing: a spent output publishes the public key that guards every other output paying to the same key, and any observer may record the chain today and attack it later. We propose a chain whose signatures are lattice-based from genesis, using ML-DSA-44 in place of ECDSA, whose peer-to-peer transport agrees keys with ML-KEM-768 in place of an elliptic-curve handshake, and whose hashing and proof-of-work are built on SHA3-256 throughout. The substitution is not free: a signature grows from 71 bytes to 2420, and a public key from 33 to 1313. We show that the honest consequence of that arithmetic is to abandon the witness discount and to size blocks against the cryptography rather than against tradition, and we derive the resulting parameters.

1. Introduction

Commerce on the Internet has come to rely on a small number of trusted third parties, and the peer-to-peer chain removed the need for them by making signatures and proof-of-work do the work that trust used to do. That construction is sound. It has one dependency that its authors could not have been expected to weigh differently in 2008: the signature scheme.

Elliptic-curve signatures rest on the hardness of computing discrete logarithms. Shor's algorithm reduces that problem to polynomial time on a quantum computer of sufficient size and error correction. No such machine is known to exist. The question is therefore not whether coins are at risk today, but whether a system intended to hold value for decades should depend on a problem that is known to be easy for a machine that is being actively built.

We take the view that it should not, and that the substitution is better made at genesis than by consensus change on a chain already holding value. A chain that migrates its signature scheme must move every unspent output, and outputs whose keys are lost cannot be moved at all.

2. The exposure

The threat is usually stated as "quantum computers break Bitcoin." Stated that way it is too coarse to design against. The exposure has a specific shape.

A public key is not published when coins are received. It is published when they are spent: the spending transaction reveals the key in order to prove the signature. Two consequences follow.

First, an output that has never been spent from is protected by the hash of its public key rather than the key itself, and hashes are not weakened by Shor's algorithm. Second, the moment coins are spent, the key becomes public, and every other output paying to that same key — a common pattern in practice — is exposed from that moment on.

The window is therefore not "when the machine is built." It is "the interval between a key becoming public and the coins under it being moved to safety," and it is bounded below by the time already elapsed. Chains are public and permanent; an adversary may record them now and compute later. Nothing spent before a migration can be un-exposed afterwards.

Hash functions face a different and much weaker attack. Grover's algorithm gives a quadratic speedup on preimage search, halving the effective security of an n-bit hash. For SHA3-256 this leaves 128 bits, which is sufficient. Proof-of-work is likewise not the vulnerable component: a quadratic speedup against a target that adjusts every 2016 blocks is absorbed by difficulty, not by failure.

The signature scheme is the part that must change. The rest follows from what that change costs.

3. Signatures

We use ML-DSA-44, standardised as FIPS 204 and derived from CRYSTALS-Dilithium. Its security rests on the hardness of finding short vectors in module lattices, a problem for which no efficient quantum algorithm is known.

The sizes are the design constraint of this entire system:

	ECDSA (secp256k1)	**ML-DSA-44**
public key	33 bytes	1312 bytes
signature	~71 bytes	2420 bytes
secret key	32 bytes	2560 bytes

A signature is 34 times larger and a public key 40 times larger. Any design that treats this as an implementation detail will produce a chain that either cannot carry ordinary traffic or cannot be validated cheaply. We treat it as the premise.

There is no elliptic-curve code in the system. Recovery of a public key from a signature, on which some compact transaction formats depend, does not exist for ML-DSA, so keys are carried explicitly.

4. Transactions

A transaction is a chain of digital signatures over an unspent-output ledger, unchanged in structure from the original design. What changes is where the signature is carried.

Signatures and public keys are placed in the witness rather than in the input script. This is not only a matter of transaction malleability. A legacy script may not push a stack element larger than 520 bytes — a limit inherited from the original design and retained here — and neither a 2420-byte signature nor a 1313-byte serialised public key fits within it. Witness items are permitted up to 3600 bytes, which accommodates both, and script-hash spends whose scripts embed public keys directly.

The consequence is that spends are witness-only by construction rather than by policy. It is not possible to write a legacy input script that pushes a valid ML-DSA public key, and an invalid key is rejected before any sighash is computed. This closes, as a side effect, the quadratic-hashing behaviour of legacy signature validation: the expensive step is unreachable.

An ordinary payment — one input, two outputs — measures 3856 bytes on this chain, against roughly 140 for its elliptic-curve equivalent.

5. Block space

Bitcoin separates transaction data into two classes and charges the witness class one quarter of the weight of the rest. The discount is a deliberate incentive: it prices the data that a validating node may eventually discard below the data it must keep forever.

That reasoning does not transfer. Under ML-DSA the witness is not a minor appendix to a transaction; it is the overwhelming majority of it. A discount would mean charging almost nothing for almost the entire chain, and the limit would stop describing any real resource.

We therefore set the witness scale factor to one. A block is bounded by its serialised size and nothing else, and the bound is 32,000,000 bytes. The figure is derived rather than chosen. At 3856 bytes for an ordinary payment a block admits roughly 8,300 of them; a 4-million-weight-unit block admits roughly 7,100 of the elliptic-curve equivalent. The two are within twenty percent of each other, and that is the whole intent of the number: the change of signature scheme should cost the user bandwidth and disk, which it plainly does, but not throughput.

The sigop budget is 640,000 per block, which preserves Bitcoin's density of one sigop per fifty bytes exactly. Holding that ratio while the block grows is deliberate. It is also not sufficient on its own, because an ML-DSA verification costs more than an elliptic-curve one; what bounds the real work is that a signature cannot be presented without carrying 3733 bytes of signature and key with it, so the data limit binds before the sigop limit does.

Storage follows directly. At 32 MB every ten minutes a saturated chain grows by at most 4.6 GB per day, and about 170 GB per year at ten percent occupancy. This is the price of the substitution and we state it plainly rather than discounting it away.

6. Proof-of-work

The proof-of-work is double SHA3-256 over the standard 80-byte header. The retarget is unchanged: 2016 blocks, a two-week timespan, a ten-minute target.

SHA3 is a sponge construction and is not subject to length extension, so the double application is not, as in the original design, a defence against that property. It is retained for a different reason: it doubles the work per candidate header without doubling the verification cost of a block, since a block header is hashed once by a validator and billions of times by a miner.

Choosing SHA3-256 over SHA-256 is not a security claim about the latter. It follows from using one primitive throughout — transaction identifiers, Merkle trees, address hashes and proof-of-work all reduce to SHA3-256 — and from the practical consequence that the existing SHA-256 mining industry does not transfer. A new chain secured by the same function as the largest chain is secured at the pleasure of that chain's spare capacity. A commodity GPU is the intended miner.

Addresses use the first 160 bits of SHA3-256 rather than RIPEMD160 of SHA-256, for the same reason of using one primitive.

7. Network

Transactions and blocks are broadcast on a best-effort basis and nodes accept the longest chain, as in the original design.

The transport itself is encrypted. Key agreement uses ML-KEM-768 (FIPS 203): the initiator sends an ephemeral encapsulation key, the responder encapsulates to it, and both derive session keys from the shared secret. The derivation binds the full handshake transcript and the network's magic bytes, so a tampered or cross-chain handshake yields non-matching keys rather than a working session. The packet stream above it is the same length-prefixed authenticated framing used by BIP324, with a four-byte length descriptor in place of three, because a 32 MB block does not fit in a 24-bit length.

We state its limit precisely. The handshake is unauthenticated and opportunistic. It provides confidentiality and forward secrecy against an observer who records traffic — including one who records it now and decrypts later, which an elliptic-curve handshake does not. It does not provide protection against an active man-in-the-middle. Mutual authentication with an ML-DSA identity key is a possible extension and is not claimed today.

8. Incentive

The first transaction in a block creates a new coin owned by the miner. The subsidy begins at 50 coins and halves every 210,000 blocks, and the supply is bounded at 21,000,000 coins. Once the subsidy ends the incentive is transaction fees alone.

There was no premine and no allocation to the author. The genesis block carries the headline

> *Citi 03/Jan/2026 The trillion-dollar quantum security race is on*

which fixes the date below and, as in the original design, comments on the reason for the chain's existence. Its output is unspendable. Every coin in circulation was produced by a miner.

9. Calculations

Consider an attacker attempting to alter a past block. The analysis is unchanged from the original design: the attacker must redo the proof-of-work of the block and all blocks after it, and catch up with the honest chain. The probability of success falls exponentially with the number of confirmations, and the presence of a quantum adversary does not change this materially, since Grover's quadratic speedup applies equally to the honest majority and is absorbed by the difficulty adjustment.

The calculation that does change is the cost of validation. A block of 32 MB consisting of ordinary payments contains on the order of 8,300 inputs and therefore that many ML-DSA verifications. At the per-verification cost of the reference implementation this is a fraction of a second on one core and is parallel across inputs, so it is not the binding constraint. The binding constraint is bandwidth and storage, at up to 4.6 GB per day when blocks are full.

The 96-bit truncation used in the pooled search tooling, and other engineering choices of that kind, are outside the consensus rules and are described in the repository rather than here.

10. Conclusion

We have proposed a chain that does not depend on the discrete logarithm problem. Signatures are ML-DSA-44, key agreement on the wire is ML-KEM-768, and hashing and proof-of-work are SHA3-256 throughout. The ledger, the script system, the difficulty adjustment and the incentive structure are those of the original design, because those parts are not what a quantum computer threatens.

The substitution costs a factor of roughly twenty-seven in transaction size, and we have shown what follows from paying that cost honestly rather than hiding it: no witness discount, a block bound set against the cryptography, a sigop budget tightened per byte, and a growth rate stated in the open rather than buried. The system is live and its rules are fixed from genesis, because a signature scheme is not something a chain holding value can change later without abandoning the coins whose keys are lost.

References

1. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
2. National Institute of Standards and Technology, "Module-Lattice-Based Digital Signature Standard," FIPS 204, 2024.
3. National Institute of Standards and Technology, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," FIPS 203, 2024.
4. National Institute of Standards and Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," FIPS 202, 2015.
5. P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," SIAM J. Comput. 26(5), 1997.
6. L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," STOC, 1996.
7. P. Wuille, D. Wuille, et al., "BIP324: Version 2 P2P Encrypted Transport Protocol," 2023.