Menhir: Peer-to-Peer Electronic Cash on a Single Cryptographic Assumption

Draft 0.1 · ticker MNH

2026-11-24

> Status. This paper describes a design. No Menhir network exists: a regtest prototype (phase 1) is being built on Bitcoin Core v32.1. Figures marked “estimate” come from the design and have not been measured yet. This is a technical description, not an offer of any asset. The normative specification is `doc/menhir/DESIGN.md`.

Abstract

Bitcoin’s security rests on two cryptographic bets: the SHA-256 hash function, used for mining, block linking, Merkle trees and addresses, and the elliptic-curve discrete logarithm, used for signatures. A large quantum computer running Shor’s algorithm breaks the second and leaves the first standing. The post-quantum ledgers proposed so far replace elliptic curves with lattice signatures, a second bet that is younger than the one it replaces. Menhir removes the second bet. Coins are spent with stateless hash-based signatures (SLH-DSA, FIPS 205), so that proof-of-work, commitments and signatures all rely on one family of hash functions, SHA-2. Coins can only be held in outputs spendable by such signatures, from the first block, by consensus rule. Everything else is kept from Bitcoin: proof-of-work with SHA-256d, the UTXO model, the script system and the incentive of block rewards. The price is signature size, and the paper states it plainly and describes how it is paid down.

1. Introduction

Nakamoto’s design needs two things from cryptography. A hash function makes blocks expensive to produce and cheap to check, and binds each block to its history. A signature scheme lets the owner of a coin, and nobody else, authorize its transfer. Bitcoin chose SHA-256 for the first and ECDSA over secp256k1 for the second.

The two choices do not age alike. Against a hash function, the best known quantum attack is Grover’s search, a quadratic speed-up that halves the security exponent and parallelizes poorly. Against the elliptic-curve discrete logarithm, Shor’s algorithm is polynomial: with a large enough fault-tolerant machine the private key follows from the public key. Resource estimates for that machine have fallen steadily; in March 2026 Google Quantum AI put secp256k1 under half a million physical qubits.

A ledger that wants to outlive that machine must change its signatures. The question this paper answers is what to change them to, and what else has to change with them.

2. Why not lattices

The signature standardized by NIST for general use is ML-DSA (FIPS 204), built on the hardness of problems over module lattices. Its signatures are small for a post-quantum scheme (about 2.4 kB) and fast, and it is the choice of the post-quantum ledgers announced or launched in 2026 that we could find.

ML-DSA is considered sound. It is also a structured mathematical assumption with roughly fifteen years of concentrated cryptanalysis, and structure is what attacks exploit: two candidates of the same NIST process, SIKE and Rainbow, were broken with classical computers in 2022, late in the evaluation. A ledger secures value for decades and cannot rotate keys on behalf of its users. For that use, the age and simplicity of the assumption matter more than the size of the signature.

Hash-based signatures need no assumption beyond the hash function. The idea dates from Lamport and Merkle (1979), and SHA-2 has been attacked in public since 2001. A ledger that mines with SHA-256 already depends on it completely: if SHA-256 fails, proof-of-work fails, whatever the signature scheme. Using the same family for signatures adds nothing to what must hold.

3. Design principle

One assumption. Every security property of Menhir reduces to properties of SHA-2:

Function in the system	Primitive
Proof-of-work, block linking	SHA-256d
Transaction and script Merkle trees, addresses	SHA-256 (tagged hashes)
Transaction signatures	SLH-DSA-SHA2-192s: SHA-256 and SHA-512

One nuance, stated rather than hidden: at the security level chosen, FIPS 205 uses SHA-512 next to SHA-256 inside the signature scheme. Both are members of SHA-2 with the same construction, but they are two functions. The claim is “one family”, not “one function”.

Three rules follow from the principle:

1. No elliptic curve is reachable from the ledger. Not as a legacy output type, not as a script opcode path.
2. The signature algorithm is named in every public key, so the scheme can be extended without a new address format.
3. Consensus code takes no cryptographic dependency beyond the hash implementations already in Bitcoin Core.

4. Signatures

Menhir uses SLH-DSA-SHA2-192s as specified in FIPS 205, without modification.

How it works, in short. A one-time signature (WOTS+) signs one message by revealing points along hash chains. A Merkle tree over many one-time keys (XMSS) turns them into a many-time key. A tree of such trees (the hypertree) makes the number of usable one-time keys astronomically large, and a few-time scheme at the leaves (FORS) lets the signer pick a leaf at random instead of remembering which ones were used. The result is **stateless**: a wallet keeps no counter, and restoring a backup cannot cause key reuse. This is the property that stateful schemes such as XMSS lack, and its absence is fatal when a backup is restored.

Parameters.

Quantity	Value
NIST security category	3 (at least as hard to break as key search on AES-192)

Quantity	Value
n, h, d, a, k, w	24 bytes, 63, 7, 14, 17, 16
Public key	48 bytes
Secret key	96 bytes (derived from a 72-byte seed)
Signature	16,224 bytes: 24 randomizer + 6,120 FORS + 10,080 hypertree
Signatures per key before the security bound degrades	2^64
Hash functions	F, PRF: SHA-256 · H, T_l, H_msg: SHA-512 · PRF_msg: HMAC-SHA-512

Signing is hedged: each signature mixes fresh randomness with the deterministic nonce, which protects against fault attacks without depending on the quality of the randomness.

Why category 3. Category 1 (128s) gives signatures of 7,856 bytes and is the level NIST considers sufficient. Menhir takes the next level for margin against improvements in generic quantum search: a Grover attack on category 3 needs on the order of 2^96 sequential evaluations instead of 2^64. The cost is a signature twice as large.

Implementation. The scheme is written directly on Bitcoin Core's SHA-256, SHA-512 and HMAC code, validated against every NIST ACVP test vector of the parameter set, and cross-checked against an independent implementation.

5. Outputs and addresses

Coins are held in **pay-to-Merkle-root (P2MR)** outputs, following the structure proposed for Bitcoin in BIP-360: a witness version 2 program holding the 32-byte root of a tree of scripts.

- The output commits to a Merkle tree of leaf scripts, built with the tagged hashes of BIP-341. There is no key path: Taproot's internal key, an elliptic-curve point, does not exist here.
- To spend, the witness provides the inputs to one leaf script, the script itself and the Merkle path to the root.
- Leaf scripts run under the tapscript rules of BIP-342, with the signature opcodes redefined. A public key is one **algorithm identifier** byte followed by the key. Identifier `0x01` is SLH-DSA-SHA2-192s. A key in the BIP-340 form makes the script fail. Unknown identifiers are reserved for future schemes and succeed by consensus, as unknown key types do in BIP-342, while standard relay policy rejects them.
- The signed message is the BIP-341 signature hash with the BIP-342 extension, so it commits to the amounts and scripts of all inputs, to the leaf and to the chosen hash type.
- The stack element limit inside P2MR is 16,384 bytes instead of 520.

The common case is a single key: a 51-byte leaf script (the 49-byte key and `OP_CHECKSIG`), whose leaf hash is the address. Addresses are bech32m with the prefix `mnh`.

Because a public key appears on the ledger only when its output is spent, and the output commits to a hash of the script, an unspent output reveals nothing an attacker could work on except a 256-bit hash.

Agility. The identifier byte and the tree give two upgrade paths. A new scheme, for example a more compact hash-based parameter set, takes a new identifier in a soft fork. And one address can commit to several leaves with keys of different schemes, so a holder can prepare a fallback without moving coins later.

6. A ledger with nothing to migrate

On Menhir, every output of every transaction, the block reward included, must be a P2MR output or a provably unspendable data output. Any other output makes the transaction and the block invalid. The rule holds from the genesis block, whose own reward is unspendable.

Bitcoin's hardest post-quantum problem is not the new signature but the old coins: outputs with exposed public keys, owners who never migrate, and the choice between letting a quantum attacker take them or freezing them by fiat. A ledger that never accepts a vulnerable output never faces that choice.

7. Proof-of-work and incentives

Menhir keeps Nakamoto consensus unchanged: blocks are found by SHA-256d proof-of-work, the longest valid chain by accumulated work wins, and the miner of a block collects the block subsidy and the fees of its transactions. That reward is the incentive to secure the network, exactly as in Bitcoin. As in Bitcoin, a node that only validates and relays earns nothing; it is run by those who want to verify their own coins.

The mining puzzle is brute-force search for a hash below a target. It has no algebraic structure for Shor's algorithm. Grover's algorithm gives a quadratic speed-up in theory; it lets nobody spend another's coins, and its practical value against specialized hardware is doubtful given the overhead of error correction and its poor parallelism. It is a question of future miner concentration, not of ledger safety.

Because the puzzle is the same as Bitcoin's, existing SHA-256 mining hardware of any generation mines Menhir unmodified. That is also the main risk of a young chain: hashrate is abundant and rentable, and a small SHA-256 network can be out-mined. Two measures are planned for the public network and are not yet decided in detail: a per-block difficulty adjustment, so the chain stays live under large hashrate swings, and a decision between independent mining and merged mining with Bitcoin, which would let Bitcoin's miners secure Menhir at no extra energy cost.

8. Monetary policy

- Block interval: 10 minutes.
- Initial subsidy: 50 MNH per block, halving every 210,000 blocks.
- **Floor:** after six halvings, at block 1,260,000 (about 24 years), the subsidy stays at 0.78125 MNH per block forever.
- Supply at the floor: 20,671,875 MNH. After it, about 41,062 MNH a year, an inflation of about 0.2 % that falls every year as supply grows.
- No premine and no allocation to anyone: the genesis reward is unspendable and every coin is mined.

The floor answers Bitcoin's open question of who pays for security once the subsidy is gone. Large post-quantum signatures make block space scarcer, so fees alone are an even weaker bet here; a small perpetual subsidy keeps a minimum security budget that does not depend on the fee market.

9. Capacity and cost

Hash-based signatures are large. That is the cost of the design.

	Bitcoin (P2WPKH)	Menhir (single key)
Signature	about 72 bytes	16,224 bytes
Public key	33 bytes	48 bytes (+1 identifier)
One-input, two-output payment	about 561 weight units	about 16,830 weight units (estimate)
Payments per 4M-weight block	about 7,100	about 237 (estimate)
Payments per second	about 12	about 0.4 (estimate)

Signing takes on the order of a second on a desktop processor (estimate); verification takes milliseconds.

Menhir does not answer this by multiplying the block size before measuring. The plan has four steps:

1. **Measure.** The prototype reports signing and verification times and the validation time of a full block. Block weight, the discount applied to signature data and the block interval are decided from those numbers.
2. **Compact parameter sets.** A wallet key signs far fewer than 2^64 messages. Hash-based parameter sets that cap the number of signatures per key cut signatures to roughly 3 to 4 kB at category 1 (Kudinov and Nick, 2025). They would enter under a new algorithm identifier once their parameters and security level are settled.
3. **Witness pruning.** Signatures buried under enough work can be dropped by nodes that do not need to re-verify history, which bounds the growth of storage.
4. **Aggregation (research).** A hash-based proof system (STARK) can replace the signatures of a block with one proof that they were all valid. It needs no assumption beyond hashes, so it fits the principle. It is a research track.

10. Security

Property	Attack	Classical cost	Best known quantum cost
Forging a signature	Break SLH-DSA-SHA2-192s	2^192	about 2^96 sequential Grover iterations
Redirecting an output	Second preimage on the 256-bit script root	2^256	about 2^128
Two scripts, one address	Collision on the script root	2^128	about 2^85, with very large quantum memory
Rewriting history	Out-mining the honest chain	majority of hashrate	majority of hashrate; Grover's speed-up is quadratic at best

Stated limits:

- **No proof.** As for every deployed scheme, security is the absence of known attacks, here on SHA-2. A structural break of SHA-2 would break Menhir, and Bitcoin's mining with it.

- **Implementation.** Verification is simple; signing handles secrets and must resist side channels and fault injection. The code has not been audited. An external audit is a condition of any public launch.
- **Key use.** The bound of 2^64 signatures per key is far beyond any wallet's use; compact parameter sets would introduce a real limit, to be enforced by wallets.
- **Small network.** The exposure to majority-hashrate attacks described in section 7 is real until the network's hashrate, or merged mining, makes it expensive.

11. Related work

Approach	Signature assumption	Legacy outputs	Consensus
Bitcoin today	Elliptic curves (broken by Shor)	all	Proof-of-work
Bitcoin with BIP-360 (draft)	Output type without key path; post-quantum signatures deferred	remain, migration debated (BIP-361)	Proof-of-work
Bitcoin forks and new chains with ML-DSA (Bitcoin Quantum, Naoris, Splendor, QRL 2.0)	Lattices	vary	Proof-of-work or proof-of-stake
QRL 1.x	Hash-based, stateful (XMSS)	none	Proof-of-work, later proof-of-stake
Menhir	Hash-based, stateless (SLH-DSA)	none, by consensus	Proof-of-work, SHA-256d

The table reflects public information as of November 2026 and was not verified beyond it. To our knowledge no mineable ledger uses stateless hash-based signatures as its only scheme.

Menhir builds directly on prior work: Bitcoin and Bitcoin Core (the code base, under the MIT license), BIP-341 and BIP-342 (script trees and tapscript), BIP-360 (the P2MR output), FIPS 205 and the SPHINCS+ line of research behind it.

12. Limitations and open problems

- Throughput is low until the capacity work of section 9 is done.
- Signing is slow for constrained devices; hardware-wallet feasibility has to be measured.
- Hierarchical wallets lose public derivation: there is no equivalent of an extended public key, so watch-only address generation needs another design.
- Multi-signature scripts multiply the signature size; there is no hash-based threshold or aggregate signature of practical size today.
- Dust and fee policy for 16 kB inputs is undefined.
- The legal treatment of a mined asset without an issuer has not been reviewed.

13. Roadmap

Phase	Content	Done when
0	Design	Done (2026-11-24)
1	Regtest prototype: the signature primitive, the chain and the PQ-only rule, P2MR spending, measurements	A coin is mined to a P2MR address and spent with a real signature; every invalid spend is rejected
2	Public testnet: network parameters, difficulty adjustment, emission floor, wallet, capacity decision, mining stack	External nodes connected; weeks without a consensus split
3	Compact parameter sets, witness pruning, fuzzing, external audit	No open critical finding
4	Mainnet: fair launch, reproducible builds	Genesis on an announced date

14. Conclusion

A quantum computer breaks one of Bitcoin's two cryptographic bets. Replacing the broken bet with a new one is the obvious repair and the one most projects have chosen. Menhir takes the other road: it keeps the bet that survives and builds everything on it. Proof-of-work, commitments and signatures rest on SHA-2; no vulnerable output can ever exist; the incentive structure is Bitcoin's. The design pays for this in bytes, says so, and treats signature size as the engineering problem to solve in the open, with measurements, rather than as a detail to hide behind larger blocks.

References

1. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008.
2. NIST, FIPS 205, "Stateless Hash-Based Digital Signature Standard", 2024.
3. NIST, FIPS 204, "Module-Lattice-Based Digital Signature Standard", 2024.
4. P. Wuille, J. Nick, A. Towns, BIP-341 "Taproot: SegWit version 1 spending rules" and BIP-342 "Validation of Taproot Scripts".
5. BIP-360, "Pay to Merkle Root" (draft), `bitcoin/bips`.
6. D. J. Bernstein et al., "The SPHINCS+ Signature Framework", ACM CCS 2019.
7. M. Kudinov, J. Nick, "Hash-based Signature Schemes for Bitcoin", IACR ePrint 2025/2203.
8. P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", FOCS 1994.
9. L. Grover, "A fast quantum mechanical algorithm for database search", STOC 1996.
10. Google Quantum AI, resource estimate for the elliptic-curve discrete logarithm on secp256k1, March 2026.
11. L. Lamport, "Constructing digital signatures from a one way function", 1979; R. Merkle, "Secrecy, authentication, and public key systems", 1979.