Hash-based signatures on Bitcoin Core: measurements from a P2MR and SLH-DSA prototype

Menhir technical report 1 · phase 1 (regtest prototype)

2026-11-25

> Status. No Menhir network exists. This report measures a prototype that runs on a private regression-test chain on one machine. The source code is not public yet, so the figures cannot be reproduced by a third party today; the measurement scripts are listed in section 9. Nothing here is an offer of any asset.

1. Summary

Bitcoin is discussing how to move to post-quantum signatures (BIP-360, BIP-361). The stateless hash-based option, SLH-DSA (FIPS 205), needs no new cryptographic assumption beyond the hash function Bitcoin already depends on, and it is usually dismissed in one line: the signatures are too large. This report replaces the one line with numbers, measured on a working implementation inside Bitcoin Core v32.1.

Question	Measured answer
How large is a payment?	16,832 WU (16,421 bytes, 4,208 vB) for one input and two outputs
How many fit in a 4,000,000 WU block?	237 (0.395 per second at one block per 600 s)
How fast is verification?	253.59 µs per signature on one thread
How long does a full block take to validate?	11.0 to 18.0 ms on 15 threads, on a loaded machine
How fast does the chain grow?	At most 204.57 GB per year with every block full
Is the implementation correct?	66 of 66 NIST ACVP cases; 100 of 100 signatures verified by OpenSSL 4.1, and 100 of 100 OpenSSL signatures accepted by the node

Three findings follow from the numbers:

1. Size is the whole cost. The signature is 98.8 % of the bytes of a payment and witnesses are 99.16 % of the bytes of a full block.
2. Verification is not the problem. A full block of honest payments validates in under 20 ms.
3. The tapscript signature-operation budget does not price a hash-based check. A block built to abuse it took 5.75 s to validate. This must be repriced before any public network.

2. What was built

A fork of Bitcoin Core v32.1 (commit `f0de87f040`). Consensus and policy changes are gated by chain parameters that only the prototype chain `mnhregtest` sets; the upstream chains keep upstream behaviour. The diff against v32.1 in `src/`, `test/` and `contrib/` is 53 files and 9,435 added lines.

- **Signature scheme.** SLH-DSA-SHA2-192s (FIPS 205, NIST security category 3): public key 48 bytes, secret key 96 bytes, signature 16,224 bytes. It is written directly on Bitcoin Core's own SHA-256, SHA-512 and HMAC code, with no new dependency.
- **Output type.** P2MR, in the style of BIP-360: witness version 2, `OP_2 <32-byte root>`. The root is a BIP-341 script tree with no key path, so no elliptic-curve key is committed anywhere.
- **Spending rules.** BIP-342 tapscript with three changes: a public key is one algorithm identifier byte followed by the key (`0x01` = SLH-DSA-SHA2-192s; a 32-byte BIP-340 key makes the script fail); the signature is 16,224 bytes, or 16,225 with a hash-type byte; the stack element limit is 16,384 bytes instead of 520. The signed message is the unchanged BIP-341/342 signature hash.
- **PQ-only outputs.** On the prototype chain every output must be P2MR or `OP_RETURN`, from the genesis block, by consensus. No output that a quantum computer could spend can exist.
- **Proof-of-work.** SHA-256d, unchanged.

The standard single-key leaf script is `31 01 <48-byte key> ac` (51 bytes), and its spend witness is `[signature] [leaf script] [control block]` with a one-byte control block.

3. Method and environment

- Commit measured: `a94c192393`. Release build (`CMAKE_BUILD_TYPE=Release`), Apple clang 21.0.0.
- Machine: Apple M5 Max, 18 cores, macOS 27.0.1. OpenSSL 4.1.0 (23 Oct 2026) as the independent implementation.
- The machine was not idle. It was shared with other jobs during every run, with 1-minute load averages between 14 and 51. Benchmarks are reported as measured and are not corrected for load. Treat every timing as an upper bound for this hardware.
- Transaction sizes, block capacity and chain growth are properties of the serialization and do not depend on load.
- Transactions were real: signed by the node's RPC (or by OpenSSL where stated), mined into blocks on `mnhregtest` and read back with `getrawtransaction`.

4. Results

4.1 Signature primitive

Benchmark `bench_bitcoin -filter='SLHDSA.*' -min-time=10000`, one thread.

Operation	Mean	Operations per second
Key generation	27.32 ms	36.61
Sign	278.83 ms	3.59
Verify	253.59 µs	3,943

Signing costs about 1,100 times a verification. The asymmetry suits a ledger: a signature is produced once and verified by every node. The key generation run was flagged unstable by the benchmark harness (error 6.6 %).

4.2 Transaction sizes

Shape	Size (bytes)	Virtual size (vB)	Weight (WU)
(a) 1 input, 2 outputs, single-key leaf	16,421	4,208	16,832
(b) 2 inputs, 2 outputs, single-key leaves	32,744	8,320	33,278
(c) 1 input spending a 2-of-2 `OP_CHECKSIGADD` leaf, 2 outputs	32,701	8,278	33,112

Shape (a) had been estimated at 16,830 WU from the design; the measured value is 2 WU higher. A signature produced by OpenSSL gives the same weight, 16,832 WU.

For scale, the same one-input, two-output payment with a Taproot key-path spend weighs 616 WU (154 vB). That figure is computed from the BIP-141 serialization rules, not measured here. The hash-based payment is 27.3 times heavier.

4.3 Block capacity

A block was filled with shape (a) payments until the next one was rejected.

- Block at height 117: 238 transactions (237 payments and the coinbase), weight 3,990,116 WU, size 3,892,037 bytes, stripped size 32,693 bytes.
- The 238th payment was rejected with `bad-blk-weight`.
- Overhead for header, transaction count and coinbase: 3,990,116 - 237 × 16,832 = 932 WU.

Shape	Payments per block	Payments per second at 600 s
(a)	237 (measured)	0.395
(b)	120 (arithmetic from the measured weight)	0.200
(c)	120 (arithmetic from the measured weight)	0.200

The same block filled with Taproot key-path payments would hold about 6,491 (computed, not measured).

4.4 Chain growth

A full block of shape (a) is 3,892,037 bytes. At one block per 600 s that is 52,560 blocks and 204.57 GB per year if every block is full, which is an upper bound.

Of those bytes, 99.16 % are witness data. The stripped size of the measured block is 32,693 bytes, so the same year without witnesses is 1.72 GB (derived). A node that discards signatures buried under enough proof-of-work would store about one hundredth of the chain.

4.5 Block validation

Benchmark `P2MRBlockVerify`: script validation of 241 single-key spends on 15 script-check threads.

Run	Time per input	Error reported	Time per block	Load (1 min) at start and end
1	71.0 µs	10.1 %	17.1 ms	26.9 to 50.9
2	45.5 µs	29.0 %	11.0 ms	14.1 to 30.9
3	74.6 µs	8.7 %	18.0 ms	30.9 to 46.4

All three runs were flagged unstable by the harness. The range, 11.0 to 18.0 ms, is the result; no single run is. A single-thread block run was not measured, because the benchmark does not expose the thread count. The serial reference derived from section 4.1 is 241 × 253.59 µs = 61.1 ms.

4.6 Worst-case block

The code review of the spend path measured a block built to maximize validation work. Tapscript charges 50 units of validation weight for each executed signature check, a price set for a 64-byte Schnorr signature. One 16,224-byte signature pays for about 340 checks, and SLH-DSA verifications are not cached. A consensus-valid block that repeats one check per input as often as the budget allows validated in 5.75 s on 15 threads and 12 s on one thread.

That is 320 to 525 times the honest block of section 4.5. It is not fixed in the prototype. The capacity decision of the next phase sets the price of a hash-based check, and the 5.75 s figure is its baseline.

4.7 Correctness and interoperability

Check	Result
NIST ACVP vectors for SLH-DSA-SHA2-192s (key generation 10, signing 28, verification 28)	66 of 66
Signatures made by the node, verified by OpenSSL 4.1 (5 keys, all 7 hash types)	100 of 100
The same 100 transactions accepted by the node's mempool test	100 of 100
Signatures made by OpenSSL 4.1 with the same keys, accepted by the node	100 of 100
Negative cases (signature over another transaction; message with one bit flipped)	2 of 2 rejected
Signature hash: node against the independent Python implementation of the test framework	20 of 20 transactions, 140 of 140 vectors

The two implementations share no code: the node's is written on Bitcoin Core's hash functions, OpenSSL's is its own. The code review of the spend path ran 14 mutations of the consensus code; the test suite detected all 14.

5. What the numbers say to the BIP-360 discussion

- **Capacity falls by a factor of about 27 at an unchanged weight limit.** 237 payments per block against about 6,491. Any deployment of SLH-DSA at this parameter set is a decision about capacity: the witness discount, the weight limit, smaller parameter sets, or aggregation.
- **Validation time does not constrain the block size.** At 253.59 µs per verification, a block could carry many times more signatures before validation approached one second on this hardware. The binding limits are bandwidth and storage.
- **Storage is almost entirely signatures.** With 99.16 % of block bytes in witnesses, pruning buried witnesses changes the storage cost by two orders of magnitude. The trust model of doing so is an open design question.
- **Budgets inherited from elliptic-curve signatures need review.** The validation-weight charge is the example found here. Dust thresholds and fee policies sized for 100 vB spends are the next candidates.
- **The larger stack element limit worked without incident** in the unit and functional tests of the prototype. That is absence of failures in a prototype, not an audit.

6. Limitations

- One machine, one architecture (ARM64), under load. No idle-machine run was achieved.
- Every block benchmark was flagged unstable. Timings are ranges.
- A regtest chain on one host: block propagation, compact block relay and initial block download over a real network were not measured.
- SHA-256 and SHA-512 compression counts per verification were not measured.
- Shape (c) was signed by OpenSSL because the node signs single-key leaves only.
- The code has had internal review with mutation testing and no external audit.
- The source is not public, so the results are not independently reproducible yet.

7. What comes next

The measured numbers feed the capacity decision of phase 2: block weight, witness discount, block interval and the price of a signature check. Later phases cover compact limited-use hash parameter sets (roughly 3 to 4 kB per signature at category 1 in the published proposals), witness pruning, and an external audit.

8. References

1. NIST, FIPS 205, “Stateless Hash-Based Digital Signature Standard”, 2024.
2. BIP-341 “Taproot: SegWit version 1 spending rules” and BIP-342 “Validation of Taproot Scripts”.
3. BIP-360, “Pay to Merkle Root” (draft), bitcoin/bips.
4. BIP-141, “Segregated Witness (Consensus layer)”.
5. NIST ACVP-Server test vectors, commit `975de31eb83d87039ec88934fdc47d8c312b892d`.
6. M. Kudinov, J. Nick, “Hash-based Signature Schemes for Bitcoin”, IACR ePrint 2025/2203.
7. Menhir white paper, draft 0.1, `doc/menhir/WHITEPAPER.md`.

9. Source of every figure

All measured figures come from `doc/menhir/phase1-measurements.md` (commit `c8d9aa89af`), which holds the commands, the raw output and the measurement scripts (`measurements/*.py`). The worst-case block of section 4.6 comes from the code review of the spend path. Figures marked “computed” or “derived” are arithmetic on measured values or on the BIP-141 serialization rules and are labelled where they appear.