The Race to Replace Bitcoin's Signatures

Quantum, AI, and Hash Functions

Jonas Nick, Blockstream Research · Plan B Forum, October 24, 2026



The gap

THE CRYPTOGRAPHY

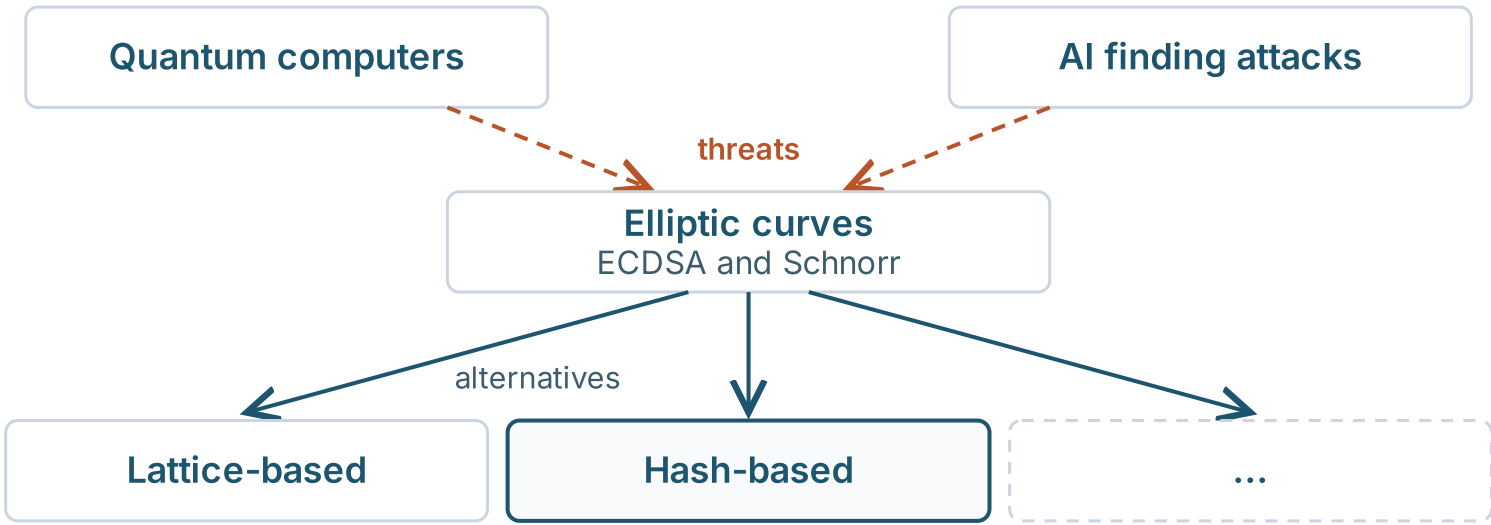
moving fast

THE CONCRETE PLAN FOR BITCOIN

catching up, more to do

The race in the title is the cryptography moving fast. It is not a call for a quick soft fork.

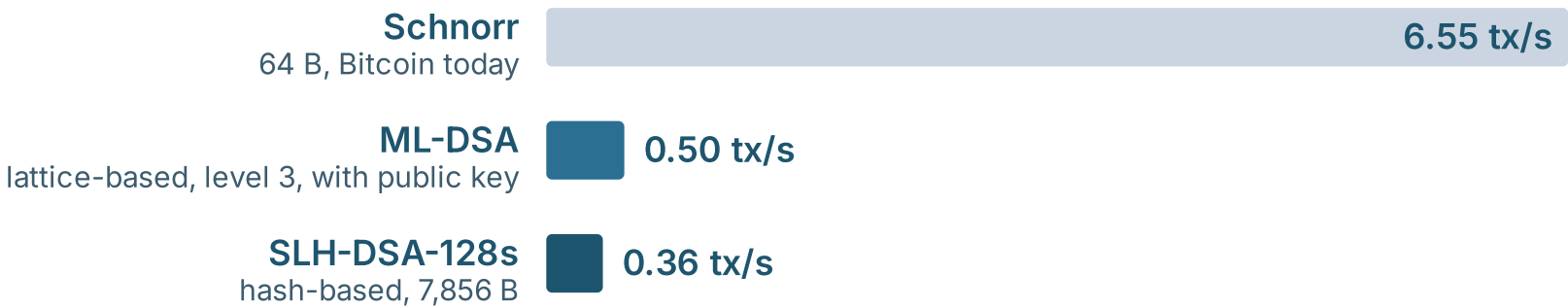
Why hash-based signatures



only hash functions, which Bitcoin already uses

Note on AI attacks: cryptographers believe some problems are genuinely hard for any classical algorithm, and that breaking Bitcoin's elliptic curve is one of them. No faster attack in 30 years. Even an attack 99% faster than today's would leave 121 of 128 bits.

Signatures are not the easy part

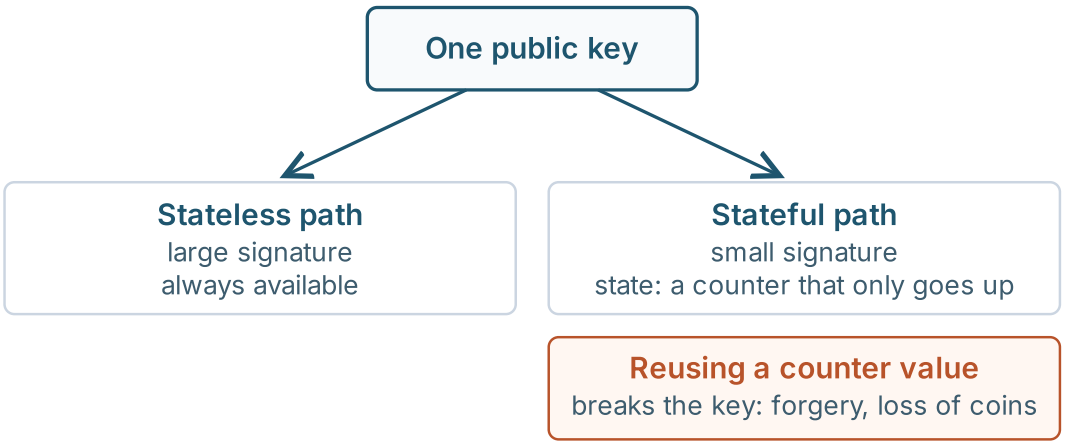


Less efficient signatures mean fewer people can use Bitcoin.

Post-quantum signatures also cost features: unhardened BIP 32 derivation and threshold signing (MuSig, FROST) are either unavailable or harder to achieve.

Throughput for the average transaction shape, 2.27 inputs and 2.64 outputs, at the current block weight limit.

SHRINCS


One public key
Stateless path
large signature
always available
Stateful path
small signature
state: a counter that only goes up
Reusing a counter value
breaks the key: forgery, loss of coins



BIP SHRINCS

548 B

Stateful path

minimum signature size

5,777 B

Stateless path

26% below SLH-DSA-128s

about 3 tx/s

Best case

if everyone signs statefully; 0.36 for SLH-DSA

The first post-quantum signature proposal designed for Bitcoin. It specifies exact algorithms and comes with a reference implementation, so it can be analyzed, criticized, and improved.

With Mike Casey, conduition, Ethan Heilman, Mikhail Kudinov, Oleksandr Kurbatov, Boris Nagaev, remix7531.

github.com/SHRINCS/shrincs-bip

Design considerations

	SLH-DSA-128s	BIP SHRINCS	alternative
stateless signature	7,856 B	5,777 B	4,256 B, 26% smaller
key generation	1×	1×	16×
signing	1×	0.77×	9×
verification per byte	1×	0.97×	8× (1.1× Schnorr, average)

Rule for the BIP: every cost at most SLH-DSA's, then the smallest signature. Signature budget 2^{40} instead of 2^{64}.



SHRINCS parameter explorer

Parameter search and explorer by Blockstream Research.

Public review of BIP SHRINCS

Project 11 (paraphrased)	Reply
Security responsibility shifts from the protocol to wallets; a stale counter fails silently.	• Agreed: the stateful path depends on wallet infrastructure the protocol does not prescribe • Wallets can start stateless and enable stateful signing later, without another soft fork
Fees are low and block space is abundant.	• Today's fees reflect 64-byte signatures; stateless SHRINCS is about 90× larger • Store-of-value holders eventually need to spend
Custodians will not use the stateful path; the "fallback" becomes the default.	• Agreed: replicas cannot share stateful slots, and custodians may simply sign statelessly • A designated signer can sign statefully while replicas sign statelessly • Calling the stateful path an optimization is right; the BIP drops "primary" and "fallback"
Consider a stateless-only scheme with a larger block size.	• Every node validates every block; bigger blocks cannot fully solve this • SHRINCS does not depend on a block-size increase • Modified algorithms: stateless signatures of about 3,300 bytes, 50× Schnorr instead of 90×. Worth leaving SLH-DSA?

Review by Project Eleven (Alex Pruden, Conor Deegan).

review and our reply

Dealing with state

State management guide

Practical rules for wallets: rollback-resistant storage, store-then-sign, fresh addresses, redundancy, and offloading counters to an untrusted host behind a commitment on the device.

Wallet IDs

A restored seed derives fresh stateful keys under a new, public ID, so an old counter is never needed again.

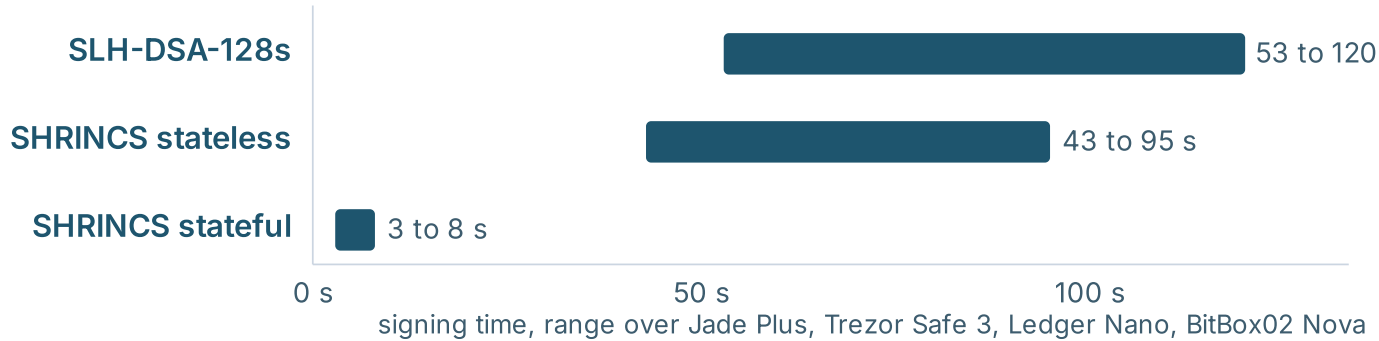
Atomic state migration

Move the state from one device to another over a two-way protocol, so it never exists in two places.

The design space for handling state is open and being explored.

State management guide by the BIP authors. Wallet IDs and atomic migration discussed in the BIP repository.

Can hardware wallets sign hash-based signatures?



Yes. Current devices sign the stateful path in seconds and the stateless path in about a minute.

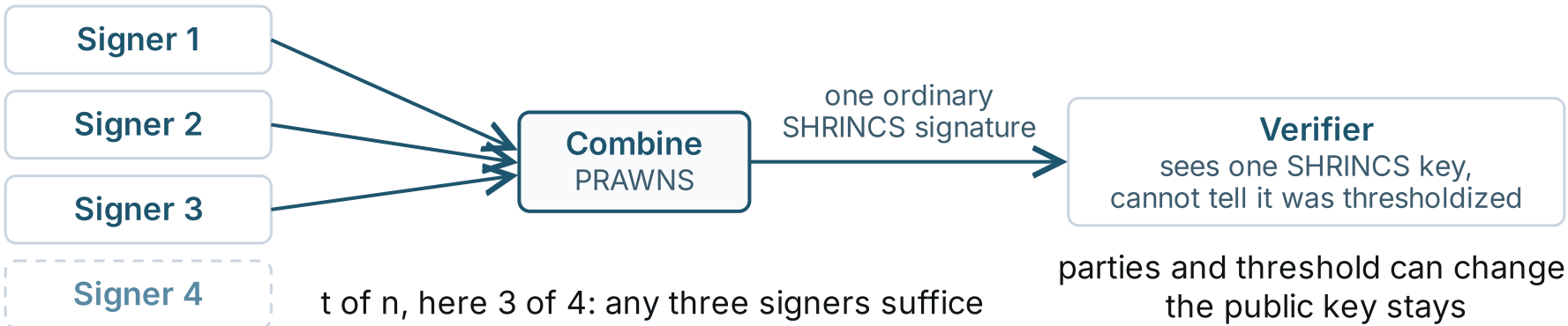
Hypertree pruning: trade signature budget for much faster key generation and signing.

Caching: much of the signing work is the same for every signature. A BIP PR specifies how to keep it; benchmarks are in progress.

Benchmarks and caching by Blockstream Research; hypertree pruning by conduition. Stateful bar: pre-BIP SHRINCS-L variant, comparable to the BIP's parameters.

benchmark post · hypertree pruning · cache management PR

~~"Hash-based signatures cannot do threshold signing"~~



PRAWNS: threshold signing for SHRINCS. The stateful path directly, the stateless path with hypertree pruning.

Caveats: no distributed key generation, so a trusted dealer. Any two signing groups must share an honest signer: 3 of 4 tolerates one corrupt signer, 2 of 3 tolerates none.

Implementing SHRINCS

What would a modern post-quantum cryptographic library for Bitcoin Core look like?

handwritten C proof: the code matches the specification ➔ **specification** proof: forging it breaks the ➔ **hash function assumptions**

- **libshrincs:** handwritten C, like libsecp256k1, so familiar to Bitcoin reviewers
- A proof of concept for the one-time signature in SHRINCS's stateful path, with both proofs machine-checked
- Until recently, both proofs were prohibitively expensive. In 2026, LLMs can write most of the proofs
- Humans review the definitions and the theorem statements

libshrincs by remix7531 (library, correctness proofs, Rocq model) and Jonas Nick (security proof).

Proving SHRINCS

1

Machine-checked proof of the complete SHRINCS scheme

In progress. A first version is done and under review.

→

2

Found on the way: a gap in the SPHINCS+ security proof that SLH-DSA relies on

The proof does not count the hash evaluations its reduction makes. In the most recent paper proof and in its machine-checked formalization. A gap in the proof, not an attack.

LESSON

Machine-checked proofs are no cure-all. The gap was missed because the existing formalizations do not model the reduction cost.

IN PARALLEL

A pen-and-paper proof of the complete SHRINCS scheme is also in the works, without the gap.

A stateful path that tolerates reuse

HG-WOTS is a one-time signature that keeps most of its security if a key is used twice.

80 bits

of security after a second use

a reused WOTS+C key has about 33 bits

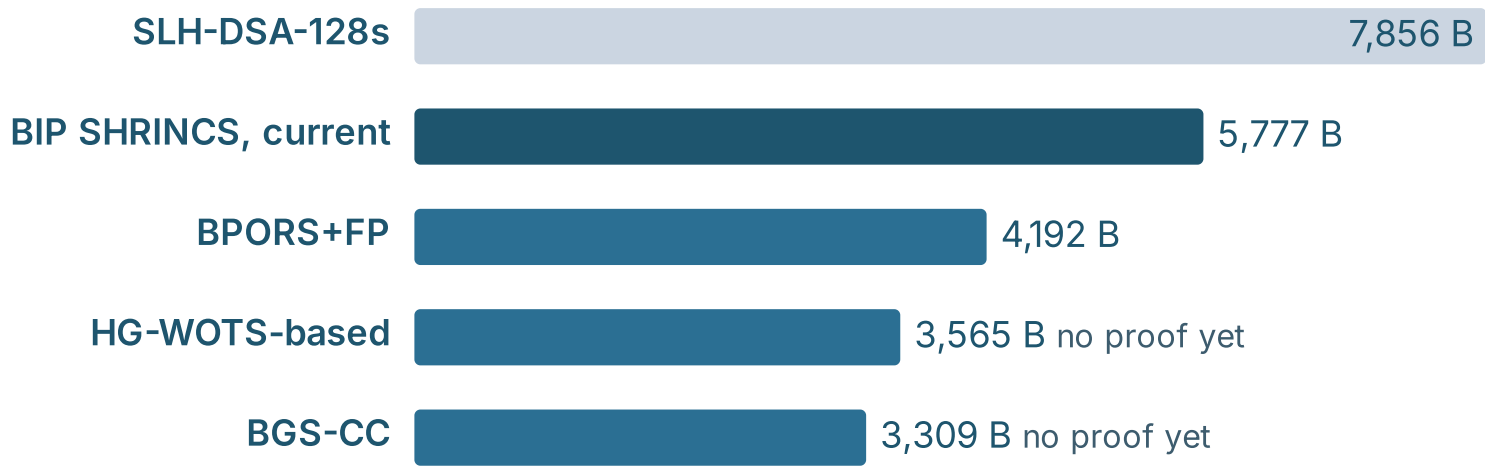
786 B

per one-time signature

against 530 B for WOTS+C in BIP SHRINCS

HG-WOTS was invented by AI, with little steering from us. There is no security proof yet.

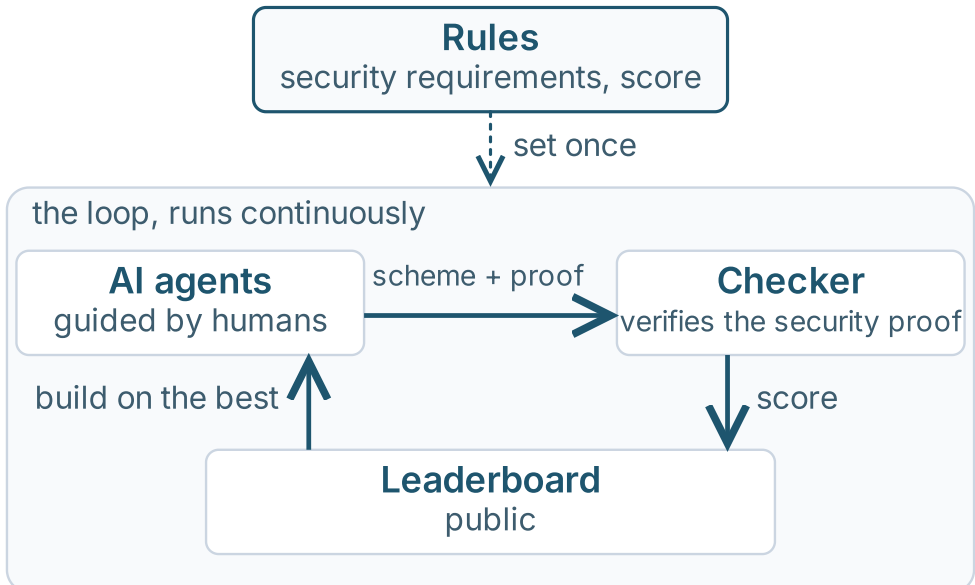
Optimizing the stateless path



All rows target 128-bit security with key generation, signing and verification per byte at most SLH-DSA-128s's. SLH-DSA-128s has a budget of 2^{64} signatures, the others 2^{40}.

BPORS+FP by Chongxu Ren, Kaiyi Zhang, Hongbo Yu. BGS-CC by Chongxu Ren, Hongbo Yu. HG-WOTS-based path by Blockstream Research with Claude Fable.

https://sig.golf: autoresearch



Autoresearch: AI agents propose, a checker verifies, results go on a public leaderboard, and the next agents build on the best.

- A public collaboration on hash-based signatures
- Running for three weeks
- The best scheme got about 50% smaller and faster to verify
- The agents invented techniques that were not in the literature

Score: signature bytes × verification cycles (zkVM cost model); Bitcoin would score differently.

sig.golf

Outlook

- **Smaller stateless signatures.** 43% below the BIP's, unproven, and autoresearch may find better ones. Whether BIP SHRINCS will break SLH-DSA compatibility is open.
- **Lattice-based signatures.** We expect improvements there too.
- **Parameters.** Is there a better choice for BIP SHRINCS?
- **FXMSS.** The BIP's stateful path is a flexible XMSS. Too much flexibility, or not enough?
- **Script integration.** The Script BIP is being worked on; the stack element size and sigop limits are the main sticking points.
- **Hardware wallets.** They can sign today; hypertree pruning and caching are being explored.
- **State management.** A guide exists; wallet IDs and atomic migration need more exploration, and eventually a standard.
- **Threshold signing.** PRAWNS is one design; alternatives with other trade-offs are worth watching for.
- **libshrincs.** A proof of concept for the one-time signature; next is completing it for BIP SHRINCS.
- **Proofs.** Machine-checked and pen-and-paper proofs of SHRINCS are in the works.
- **Public feedback.** We hope for more.



nickler.ninja/slides

@n1ckler · @blksresearch

Blockstream Research, hash-based signatures: Dmitrii Kurbatov, Jonas Nick, Mikhail Kudinov, Oleksandr Kurbatov, Viktor Mashtaliar, Yaroslava Chopa.

BIP SHRINCS
SHRINCS proposal, Delving Bitcoin
SHRINCS parameter explorer
Searching for SHRINCS parameters
Project 11 review and reply
State management guide
Wallet IDs, atomic state migration
Hardware wallet benchmarks
Hypertree pruning
Cache management PR
PRAWNS, eprint 2026/2142, blog post

libshrincs
Towards libshrincs
Choosing a verification toolchain
The gap in the SLH-DSA proof
HG-WOTS
BPORS+FP, eprint 2026/1328
BGS-CC, eprint 2026/2502
sig.golf
OP_CHECKSHRINCS talk, OP_NEXT 2026
FIPS 204, FIPS 205