A Store of Order

Electronic cash everyone can verify

NEWSATOSHI · newsatoshi.org · November 2026 · draft for review

Abstract. Bitcoin made the order of payments a matter of proof instead of trust. Their validity it left to re-execution: anyone who wants to know that no coin was spent twice must check every payment since 2009, at a cost that grows with every block and that most participants now hand to someone else. We propose electronic cash in which validity is proven too. Every payment carries a zero-knowledge proof that it is valid, and the proofs are folded into one, so that a single proof of a few hundred kilobytes, checked in milliseconds, establishes the validity of the entire history. Any participant, person or machine, can then hold the whole agreement on who owns what, at a cost that never grows. Blocks are ordered by SHA-256 proof of work that can be merged with Bitcoin's, so the chain need add no new energy. Every rule rests on SHA-256 alone, so nothing depends on a problem a quantum computer can solve. Coins are private, the supply follows Bitcoin's schedule to a permanent floor, and a coin can be locked to any condition a proof can state, so new uses never require new rules. A reference implementation runs the payments and their proofs on a test network; what remains to be built is listed at the end.

1. Introduction

Satoshi Nakamoto asked for *"an electronic payment system based on cryptographic proof instead of trust"* [1]. Bitcoin's proof covers the order of payments: to rewrite history an attacker must redo the work. Their validity is recomputed rather than proven. Anyone who wants to know that no coin has been spent twice and none created from nothing must download every block since 2009 and run every check again. By mid-2026 that is about 760 gigabytes, growing by 60 to 70 a year, and it takes between half a day and five days on consumer hardware [2].

Satoshi expected that most users would not do this. Simplified payment verification lets a client assume that the chain with the most work is valid, which holds as long as honest nodes control the network [1, §8]. Today most holders go further and rely on exchanges, custodians and other people's nodes. A money is an agreement about who owns what, and Bitcoin's is held in full by a shrinking share of the people who rely on it. Order that only a few can check is order held in trust.

Three further pressures will force Bitcoin's rules to change [3]. Its signatures fall to a quantum computer running Shor's algorithm, and millions of coins sit behind keys already exposed on chain. Its subsidy halves towards nothing, after which fees alone must pay for its security; in the autumn of 2026 they were under one percent of miner revenue. And its record is public, so coins are graded by their history and some are worth less than others.

What is needed is a payment system in which validity is a matter of proof as well as order: checked by everyone at a cost that does not grow, resting on nothing a quantum computer can break, private, and with rules that no foreseeable pressure can force to change. We propose such a system. We call it a store of order, because what it stores is the agreement itself, as a proof anyone can check.

2. Notes

A coin is a note: a value v, the owner's address a, and 160 random bits ρ. A note is published only as its commitment

$$cm = C(\text{“nsCM”}, a, v, \rho)$$

where C is one SHA-256 compression and each four-byte tag keeps one use of C apart from the others. The address is $a = \mathrm{C}(\text{“nsAD”}, k)$, the hash of the owner's 256-bit secret key k. There is no public key and no signature. Commitments are appended to a Merkle tree of depth 32 whose nodes are also C, and nodes keep every root the tree has had. Spending a note publishes its nullifier

$$nf = C(\text{“nsNF”}, k, position, \rho)$$

which only the owner can compute and nobody else can link to the note. A nullifier seen once is never accepted again. This is the structure Zcash introduced [4], with every primitive in it reduced to the one hash.

3. Payments

A payment spends up to two notes and creates two. Its public part is small: the tree root it proves against, two nullifiers, two new commitments and a fee. With it goes a STARK [5] proving that each spent note is in the tree under that root, that the spender knows its key, that the nullifiers are correctly derived, that the new notes are well formed, and that the values balance, with the fee leaving the private pool in public. The circuit computes SHA-256 itself, one compression round per row of its trace, and the STARK's own commitments and Fiat–Shamir challenges are SHA-256 as well, so the proof assumes nothing beyond the hash. At the test network's parameters the proof has 128 bits of proven soundness, is about 249 kilobytes, takes about two seconds to make on two processor cores and about ten milliseconds to check [6].

The proof reveals nothing about which notes were spent, by whom or for how much. For a STARK that requires randomising both the trace and the quotient polynomials the prover commits to; we use the construction of Haböck and Al Kindi [7]. Because the proof binds every public part of the payment, it is also the signature: nobody without the key can make a valid payment, and nobody can alter one. Nodes accept a proof only in its single canonical encoding.

The new note travels on chain, sealed with ML-KEM to the payee's key where the proof binds it; the payee's wallet finds it by trying each envelope, and change is sealed to the payer's own key, so a key alone restores a wallet. Hashes alone cannot seal a message [8], so a break of ML-KEM would reveal who paid whom, but could never move a coin. A note can also travel as a string, which needs no assumption, or as a link with a fresh key: whoever opens it first moves the coin to a key of their own, so money can reach someone who has no wallet yet. Privacy is the default, not a wall: an owner can show any payment to anyone they choose, an auditor or a court, by revealing its note.

4. Ordering

Blocks are ordered by SHA-256d proof of work over an 80-byte header in Bitcoin's layout. Nodes follow the chain with the most work, and difficulty adjusts every block by ASERT [9]. Any SHA-

256 miner can work the chain directly.

It can also be merged-mined. A Bitcoin miner places the hash of a NEWSATOSHI block in its Bitcoin coinbase; if that Bitcoin block's hash also meets the NEWSATOSHI target, the Bitcoin header, with the path from its Merkle root to the commitment, is accepted as the work for the NEWSATOSHI block. Namecoin has been merged-mined with Bitcoin since 2011, Dogecoin with Litecoin since 2014, and in 2026 about 84% of Bitcoin's hash rate also merge-mines Rootstock [10]. The reference node speaks Namecoin's format and calls, so a pool that merge-mines Namecoin can add this chain as one more entry.

Merged mining matters for two reasons. A new chain mined alone can be overpowered by renting a sliver of the hash rate that already exists; a chain merged with Bitcoin's miners is as hard to rewrite as the share of their work committed to it. And it adds no new energy. Proof of work makes the order of history costly to rewrite by turning energy into heat that cannot be recovered. Bitcoin already does that, and each Bitcoin header can certify the order of other chains at no further cost. This chain uses that spare capacity instead of burning its own: more order from the same joules, and a second income for the miners who give it.

The price is that a large Bitcoin pool could try to rewrite this chain without giving up any Bitcoin revenue; what it would lose is the value of a coin it earns. While the chain is young, release checkpoints, Satoshi's 2010 method, refuse rewrites of old history and nothing else [6, §13]. The work orders; it does not validate. That is the job of the proofs.

5. Validity

Payments travel with their proofs to the miners and provers who include and prove them, and each checks every proof it receives, in about ten milliseconds. Blocks carry only the payments' public parts, about 1,360 bytes each, most of it the sealed note. Validity is carried by a chain proof: a recursive STARK whose statement is

> the chain proof for block n−1 verifies;
> every payment in block n has a valid proof;
> block n's nullifiers are new, and its state root is block n−1's with block n's payments applied;
> block n's header, or the Bitcoin header that merge-mines it, carries the stated work.

The public output is block n's hash, its height, the cumulative work of its chain and its state root. A valid chain proof for block n is therefore a proof that the whole history up to n is valid.

Making it is the expensive part. Checking a STARK inside a STARK means computing its SHA-256 commitments inside the circuit, 9,325 compressions for each payment proof absorbed, measured, and 16,317 at mainnet's parameters [6]. Payments are folded in a tree, in parallel, by anyone who wants to; what a chain proof states is fixed by the block, so provers compete only on price and speed, and a miner pays the prover whose chain proof it includes. Mining and proving are separate. Every block carries the chain proof for the block six below it, so a chain on which the proof for block n never appears cannot be extended past n+5, and miners abandon it for one without block n. A miner therefore includes only payments whose proofs it has checked, and only as many as the provers can fold in time; an empty block is cheap to prove, so the chain never stops for want of provers, it slows to what they can prove. A payment is final when its

block is proven: at most six blocks, about an hour, the wait Bitcoin users already accept for six confirmations.

6. Verification

A participant who wants the full security of the chain needs only the latest chain proof. Checking it takes milliseconds and a few hundred kilobytes however old the chain is, and from the proofs offered by several peers the participant keeps the one with the most work, as a Bitcoin full node keeps the chain with the most work. Nothing a full node would check is taken on trust: the work is attested by the proof and the validity is the proof. The remaining assumption is Satoshi's, that at least one honest peer can be reached. To spend, a wallet also needs the Merkle path to its own notes, which any node can supply and the wallet checks against the proven root.

A proof shows that a block is valid, not that its contents were published. A miner cannot prove the blocks that follow one whose contents it lacks, so it abandons such a block as it would an invalid one; withholding contents is then a form of censorship, which, as in Bitcoin, takes most of the work.

This changes who holds the agreement. In Bitcoin the cost of holding it grows with history, so it concentrates. Here the cost is constant, so it can be held by every wallet, every phone and every machine that holds the coin. A money's rules are only as hard to change as the number of participants who would refuse a block that breaks them, and here that number can be everyone.

7. Conditions

Bitcoin has gained what it can express by changing its rules: pay-to-script-hash, segregated witness and Taproot were each a soft fork, and covenants have been argued over for years since. A money that is never to change must be complete from the start.

A note may therefore commit to a condition, the hash of a circuit's verifying key. The default condition is knowledge of the owner's key. To spend a note under any other condition, the payment proof verifies, inside itself, a proof that the condition is met. The chain learns that some condition was met, not which one, and such a payment costs more to prove, not to check. Several owners, time locks, payment channels, escrow: anything that can be stated as a circuit can lock a note, and a new use is a new circuit, never a new rule of the chain.

One condition joins this chain to Bitcoin's. A coin locked to the SHA-256 hash of a secret, or after a time back to its owner, is the lock Bitcoin's scripts express and every Lightning payment uses. Locked on both sides, it trades bitcoin for this coin with no exchange and no custodian: the secret that completes one payment completes the other, so any Lightning wallet can pay in or be paid without knowing this chain exists. The test network runs it as a built-in module, with a server that swaps it for Lightning payments, and runs conditions written as programs, public as Bitcoin's scripts are.

The other use already running there, also a built-in module that conditions will replace, is an escrow for verified computation [6, §6.6]. A buyer locks payment against a matrix product. A worker posts a bond and commits to the result. The buyer checks it privately with Freivalds' al-

gorithm [11], far faster than computing it, and a single wrong entry, shown on chain, forfeits the bond. Work whose result can be checked can be paid for without trust.

8. Issuance

The coin is LOVE, after Empedocles' name for the force that joins things. New coins follow Bitcoin's schedule, 50 a block halving every 210,000 blocks, until the subsidy reaches 0.1953125 LOVE, and then stay there. The schedule issues 20,917,968.75 LOVE before the floor; the floor then adds 10,265.625 LOVE a year, 0.049% of the supply and a falling share every year. A permanent subsidy means the security of the chain never rests on fees alone, a regime that is unstable when fees vary from block to block [12], while the supply stays as predictable as Bitcoin's.

Issuance is blind to work, as Bitcoin's is, and must be. A rule that paid new coins for useful work would have to measure the work's value, which buyers outside the protocol decide; any measure it can compute from inside can be gamed by a party that buys from itself. Useful work belongs in payments, where its result can be checked (§7). The coins pay for the one thing a protocol can measure without trust: work done on its own history.

9. Quantum computers

No coin is guarded by a problem a quantum computer can solve. There are no elliptic curves, discrete logarithms, factorisations or lattices anywhere a coin is guarded; ownership is knowledge of a preimage, proven inside a hash-based proof. The known quantum attacks on a hash function are generic. Grover's algorithm finds a preimage of a 256-bit hash in about 2^{128} steps, and the best quantum collision search needs about 2^{85} queries and as much quantum memory [13], which gains nothing over classical search once the cost of that memory is counted [14].

The proofs stay sound against a quantum adversary in the quantum random-oracle model, with a quadratic loss [15]. At the test network's parameters that leaves about 67 proven bits. Mainnet's parameters double the query soundness, after which the bound is set by the hash's 256-bit output, at about 85 proven bits against quantum query attacks: the limit for any proof committed with a 256-bit hash. Bitcoin will have to choose between freezing the coins whose keys are exposed and letting a quantum computer take them. This chain has nothing to migrate.

10. Incentive

Each participant is paid for what it adds. Miners, including Bitcoin's miners through merged mining, receive the subsidy and fees for ordering blocks. Provers are paid by miners for chain proofs. Workers are paid by buyers for checked computation. Holders pay nothing to verify. There is no premine, no allocation, no sale and no foundation. The genesis block pays no one, and its coinbase names a Bitcoin block of the day it was mined.

Why a new money and not a layer on Bitcoin? A layer can prove its own validity, but its money is bitcoin, and the pressures of §1 act on the money. A quantum computer that can find a coin's key on Bitcoin takes the coin wherever it is used, and the security budget and the public record are Bitcoin's. Only a new money can have rules free of them. Merged mining lets it share Bitcoin's work without sharing its rules.

11. Calculations

The cost of holding the whole agreement, for a Bitcoin full node and for a participant here:

	Bitcoin full node	**This chain, with chain proofs**
Data checked	every block since 2009, about 760 GB in mid-2026 [2]	one chain proof, a few hundred KB
Time to check	10 hours to 5 days on consumer hardware [2]	tens of milliseconds
Growth	60 to 70 GB a year [2]	none

The chain-proof figures are expected from the measured payment proof, which is 249 KB and is checked in about 10 ms natively and 30 to 50 ms in a web browser [6]. An 8 MB block holds the public parts of about 5,800 payments with their sealed notes, some ten a second against Bitcoin's seven or so; what a block will hold is set by the cost of proving, which the test network will measure before mainnet's limits are fixed. Today, with every proof inside its block, a test-network block holds about 32 payments.

Provers, and the nodes that serve paths, keep the state: the note tree and the spent nullifiers, roughly 128 bytes a payment: about 40 GB a year if every block were full of delivered payments, 300 GB at most. Verifying needs none of it. The main network keeps every nullifier, in a sorted Merkle tree whose root the chain proof carries, so one path proves a nullifier new and a validator stores nothing. If provers' state ever binds, pruning [17] can come as a second kind of note without touching the first.

The supply reaches its floor at block 1,680,000, after about 32 years. The probability that an attacker with a share q of the work committed to this chain catches up from z blocks behind is as Satoshi calculated [1, §11], with q measured against all honest work, direct and merged.

12. Related work

Zcash introduced shielded notes [4]; its proofs and keys rest on elliptic curves, as will those of its planned Tachyon upgrade [17]. In June 2026 a four-year-old flaw in its Orchard circuit was found to allow counterfeit coins nobody could have detected; the pool that replaced it carries a machine-checked proof that its balance holds [19]. Private money is now held to that standard: this design's main network waits on the same proof, for a circuit of one hash function and 249 constraints, every cell of which is already tested for freedom [6]. Neptune Cash is the nearest relative, with proof of work, private notes and recursive STARKs free of elliptic curves [20]; it uses a new hash made for proofs, its own miners and a premine. Mina keeps its whole chain in one recursive proof [16], under proof of stake and with proofs over elliptic curves. Ethereum is moving block validity to proofs [6, §1], under proof of stake and rules that change by a regular process. Monero is private by default and has a permanent tail emission; its signatures rest on elliptic curves and its history must be checked in full.

Bitcoin's own history can be proven from outside its rules: in August 2026 a prototype ran Bitcoin Core's consensus code inside a zero-knowledge virtual machine, with a proof checked in 27 ms but an estimated 17 GPU-years to make from genesis [18]. Nothing in Bitcoin's rules makes anyone keep producing such a proof, and it leaves the pressures of §1 where they are.

Here no chain can grow without it. Each of these systems has part of this design; the parts each lacks are the ones a money cannot add later without changing its rules.

13. Conclusion

We have proposed electronic cash in which both the order and the validity of payments rest on proof. Blocks are ordered by proof of work that can be shared with Bitcoin. The validity of the whole history is one proof that anyone can check in milliseconds. Every rule rests on SHA-256 alone, coins are private, the supply has a floor, and coins can be locked to any condition a proof can state, so the rules need never change to add a use. Because the agreement costs nothing to hold, it can be held by everyone who uses the money, and the more widely it is held, the harder it is to change.

Running on the test network in the reference implementation (Rust, MIT, 99 tests): notes and payments with zero-knowledge proofs [7], delivered on chain (§§2–3), proof of work with ASERT and merged mining with Bitcoin (§4), the issuance schedule and floor (§8), and the escrow for matrix products and the Lightning lock (§7). newsatoshi.org checks a real proof in the browser and runs a wallet that proves payments there (§3); `newsatoshi --net testnet node` joins the public test network.

Specified here, not yet built, and required before a main network: the chain proof and the market for proving it (§5), the light clients it allows (§6), conditions inside notes (§7) and a machine-checked proof of the payment circuit (§12). A pre-release review found thirteen problems; all are fixed, and all are listed at newsatoshi.org.

References

1. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
2. Spark, “Running a Bitcoin Full Node in 2026,” October 2026, compiling J. Lopp's 2025 synchronisation tests.
3. NEWSATOSHI/2.0 specification, §2, with sources for each pressure. newsatoshi.org, 2026.
4. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, “Zerocash: Decentralized Anonymous Payments from Bitcoin,” IEEE Symposium on Security and Privacy, 2014.
5. E. Ben-Sasson, I. Bentov, Y. Horesh, M. Riabzev, “Scalable, transparent, and post-quantum secure computational integrity,” 2018.
6. NEWSATOSHI/2.0 specification and reference implementation, newsatoshi.org, 2026.
7. U. Haböck, Al Kindi, “A note on adding zero-knowledge to STARKs,” IACR ePrint 2024/1037.
8. B. Barak, M. Mahmoody-Ghidary, “Merkle Puzzles Are Optimal,” CRYPTO 2009.
9. M. Lundeberg, “Static difficulty adjustments, with absolutely scheduled exponentially rising targets,” 2019; J. Toomim, aserti3-2d, Bitcoin Cash, 2020.
10. Namecoin, merged mining (AuxPoW), 2011; Dogecoin, AuxPoW with Litecoin, 2014; Rootstock, “Merged Mining Insights Report,” Q2 2026.
11. R. Freivalds, “Probabilistic machines can use less running time,” IFIP Congress, 1977.
12. M. Carlsten, H. Kalodner, S. M. Weinberg, A. Narayanan, “On the Instability of Bitcoin Without the Block Reward,” ACM CCS 2016.
13. G. Brassard, P. Høyer, A. Tapp, “Quantum cryptanalysis of hash and claw-free functions,” LATIN 1998.
14. D. J. Bernstein, “Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?” SHARCS 2009.
15. A. Chiesa, P. Manohar, N. Spooner, “Succinct Arguments in the Quantum Random Oracle Model,” TCC 2019.
16. J. Bonneau, I. Meckler, V. Rao, E. Shapiro, “Mina: Decentralized Cryptocurrency at Scale,” 2020.
17. S. Bowe, I. Miers, “A Note on Notes: Towards Scalable Anonymous Payments via Evolving Nullifiers and Oblivious Synchronization,” IACR ePrint 2025/2031.
18. Defenwycke, “Running Core's real consensus code inside a zkVM,” Delving Bitcoin, 15 August 2026.
19. Shielded Labs, Orchard flaw disclosure, 4 June 2026; Project Tachyon, Ironwood's proof in Lean, 28 July 2026.
20. A. Szepieniec, T. Værge, Neptune Cash, neptune.cash, 2025.