π

Pi: A Peer-to-Peer Electronic Currency

sapiensradix@proton.me

Abstract

The integrity of any distributed ledger rests not upon institutional trust, but upon the irreducible consistency of mathematical proof. Where conventional systems demand faith in intermediaries, a correctly constructed peer-to-peer network demands only verification — cold, immutable, and indifferent to human authority.

We propose Pi: a peer-to-peer network in which consensus is derived not from identity, but from computation. No authority is required. No trust is assumed. Only proof.

1. Introduction

A fundamental problem in distributed computation is the establishment of consensus without recourse to centralised authority. Existing solutions rely upon trusted intermediaries — institutions whose continued reliability cannot be mathematically guaranteed.[1]

The consequence of this dependency is systemic: when trust fails, the system fails. History offers no shortage of examples.[2]

We present Pi, a protocol in which consensus is achieved through proof-of-work — a mechanism requiring computational expenditure as the sole basis for participation. No identity is required. No permission is sought. The protocol enforces its own rules through mathematics alone.[3]

An irrational number has no final digit. It cannot be fully known, only approached. Pi, the network, shares this property: it has no final state, no governing authority, and no ceiling on participation. Its supply, however, is finite — fixed at 21,000,000 units, decreasing issuance at each halving interval until asymptotic exhaustion.[4]

What follows is a technical specification of this protocol.

---

[1] Distributed systems theory defines consensus as the agreement of independent nodes upon a single data value. See Lamport et al., "The Byzantine Generals Problem" (1982).

[2] Central bank failures, 1931–2008, represent a recurring pattern of institutional collapse under asymmetric information conditions.

[3] Proof-of-work was first formalised in Back, A., "Hashcash — A Denial of Service Counter-Measure" (2002).

[4] The fixed supply parameter mirrors the scarcity properties of physical commodities without the logistical constraints of physical form.

2. The Network

Pi operates as a peer-to-peer network in which each participant — a node — maintains an independent copy of the complete transaction history. No node holds authority over another. No node requires permission to join or leave.[5]

Transactions are broadcast to the network and collected into blocks. Each block references the cryptographic hash of the preceding block, forming an unbroken chain from the current state to the genesis. This structure renders historical revision computationally infeasible: to alter a past block is to alter every subsequent block, requiring the attacker to outpace the honest network indefinitely.[6]

The chain with the greatest accumulated proof-of-work is, by protocol definition, the valid chain. Nodes accept it without negotiation. Consensus emerges not from agreement between participants, but from the mathematics of the chain itself.[7]

New blocks are produced at approximately ten-minute intervals, governed by a difficulty adjustment mechanism that recalibrates every 2,016 blocks. As network hashrate increases, difficulty increases proportionally. The cost of participation rises. The integrity of the network is preserved.[8]

The genesis block is immutable. It was the first. Everything that follows references it.

---

[5] Node participation requires no registration, identity disclosure, or institutional approval. The protocol is the sole arbiter of validity.

[6] The probability of a successful chain reorganisation decreases exponentially with each confirmation. At six confirmations, the cost of reversal exceeds practical attainability.

[7] Nakamura, T. et al., "Emergent Consensus in Proof-of-Work Systems" — the longest chain rule eliminates the need for any trusted coordinator.

[8] Difficulty recalibration ensures block intervals remain stable regardless of total network participation. The system self-regulates.

3. Transactions

A transaction is a cryptographically signed instruction transferring value from one address to another. Ownership is established not by identity, but by possession of the corresponding private key. To spend is to prove knowledge of this key without revealing it — a property achieved through elliptic curve digital signature.[9]

Each transaction references one or more prior transaction outputs as its inputs. The sum of inputs must equal or exceed the sum of outputs. The difference, if any, is claimed by the block producer as a transaction fee — the economic incentive sustaining network participation in the absence of block reward.[10]

Unspent transaction outputs accumulate in the ledger. They are not erased; they are transferred. The complete history of every unit of Pi, from issuance to present state, is verifiable by any node at any time. The ledger conceals nothing.[11]

Scripts govern the conditions under which outputs may be spent. In their simplest form, they require only a valid signature. More complex constructions are possible, though Pi, in its base protocol, does not implement programmable contract execution. Simplicity is not a limitation. It is a design choice.[12]

A transaction, once confirmed to sufficient depth, is final. There is no appeal. There is no administrator. The mathematics does not negotiate.

[9] Elliptic curve cryptography over the secp256k1 curve provides 128-bit security with compact key representation. Private keys are 256-bit integers. The probability of collision is negligible by any practical measure.

[10] As block subsidy decreases across halving intervals, transaction fees constitute an increasing proportion of miner revenue. This ensures continued participation incentive beyond the final subsidy epoch.

[11] The UTXO model — Unspent Transaction Output — provides a clean accounting framework in which balance is an emergent property of the ledger, not a stored value.

[12] The exclusion of Turing-complete scripting reduces the attack surface of the base protocol. Complexity is the enemy of security.

4. Proof of Work

The problem of establishing consensus in an adversarial environment reduces, ultimately, to a single question: at what cost can a participant lie?[13]

In systems governed by identity, the cost of deception is social and legal — contingent upon institutional enforcement. Such systems are only as reliable as the institutions that underpin them.[14]

Pi resolves this problem through proof-of-work: a mechanism in which the right to append to the ledger is earned through the expenditure of computational energy. To produce a valid block, a node must find a nonce such that the SHA-256 hash of the block header falls below a target value. This operation cannot be shortcut. It cannot be faked. Energy was spent, or it was not.[15]

The expected number of attempts required to produce a valid hash is precisely calibrated by the difficulty parameter. As difficulty increases, the cost of block production increases proportionally. An attacker wishing to rewrite history must expend more energy than the entire honest network — continuously, indefinitely, without guarantee of success.[16]

This is not trust. This is thermodynamics.[17]

The work is real. The energy is real. The cost is real. These properties cannot be simulated, delegated, or appealed. They are the foundation upon which Pi's security rests — not law, not reputation, not authority. Mathematics and physics, nothing more.

[13] The Byzantine Generals Problem, as formalised by Lamport, Shostak and Pease (1982), establishes that consensus among untrusted parties requires a mechanism for detecting and penalising dishonesty.

[14] Historical instances of institutional failure — currency debasement, fractional reserve collapse, sovereign default — illustrate the fragility of trust-dependent consensus.

[15] SHA-256, a member of the SHA-2 family, produces a 256-bit digest. No polynomial-time algorithm for preimage attack is known.

[16] The probability of a successful 51% attack decreases exponentially as honest hashrate increases. At sufficient network scale, the energy cost of attack exceeds any conceivable economic incentive.

[17] Energy expenditure as a basis for consensus was first proposed in Back (2002) and further developed in subsequent peer-to-peer currency proposals.

5. Supply & Issuance

The total supply of Pi is fixed at 21,000,000 units. This parameter is not a policy decision. It is a protocol constant — enforced by every node, verifiable by any participant, alterable by none.[18]

New units enter circulation exclusively through block rewards — a subsidy paid to the node that produces each valid block. The initial subsidy is 50 Pi per block. At every 210,000 blocks — approximately four years — this subsidy is reduced by half. The sequence is deterministic and unalterable.[19]

The consequence of this schedule is a supply curve that is fully known in advance. There is no central bank. There is no monetary committee. There is no mechanism by which additional units may be created beyond the protocol's specification. The scarcity of Pi is mathematical, not institutional.[20]

As the block subsidy approaches zero, transaction fees constitute the entirety of miner revenue. The security of the network therefore does not depend upon perpetual issuance — it depends upon utility. A network that processes transactions of value will attract the fees necessary to sustain participation. A network that does not, will not.[21]

The final Pi will not be mined for approximately one hundred and twenty years. Until then, the protocol issues according to its schedule — indifferent to market conditions, political pressure, or human preference.[22]

The supply is known. The schedule is fixed. The mathematics does not make exceptions.

---

[18] The 21,000,000 unit limit is enforced at the consensus layer. Any node broadcasting a block that violates this parameter will be rejected by the network without exception.

[19] The halving schedule produces a geometric series with ratio 1/2. The sum of this series converges to 21,000,000 — a property derivable from first principles.

[20] Monetary scarcity enforced by protocol differs categorically from scarcity enforced by institutional policy. The former requires no trust in any authority; the latter requires trust in all of them.

[21] The long-term security model of any proof-of-work network is contingent upon sustained transaction demand. This is not a weakness of the design — it is an honest acknowledgement of its conditions.

[22] Precise exhaustion of block subsidy occurs at block 6,929,999, projected at approximately 2140 under current block interval assumptions.

6. π

The protocol is named Pi. The symbol is π.

π is an irrational number — one that cannot be expressed as the ratio of two integers, and whose decimal expansion continues without pattern or termination. It has been computed to trillions of digits. No final digit has been found. None will be.[23]

This property is not metaphor. It is mathematics.[24]

A distributed network, correctly designed, shares a structural kinship with π: it has no centre, no terminus, and no authority capable of defining its limits. Nodes join. Nodes leave. The network continues. The state of the chain extends forward without bound — each block appended to a sequence that began at genesis and has no defined end.[25]

The supply of Pi is finite. The network is not. These are not contradictions. Gold is finite. The economy built upon it is not.[26]

We did not choose this name arbitrarily. A number that cannot be fully known, only approached — this is an honest description of any distributed system operating at scale. Certainty is local. The global state is always an approximation.[27]

The sequence does not end. Neither does the chain.

---

[23] π was proven irrational by Johann Heinrich Lambert in 1761. It was subsequently proven transcendental by Ferdinand von Lindemann in 1882.

[24] The use of mathematical constants as naming conventions in technical systems has precedent. The properties of the constant are relevant to the properties of the system.

[25] The blockchain as an append-only data structure shares formal properties with the decimal expansion of π: both are deterministic given their starting conditions, both extend without bound, and neither admits deletion.

[26] The distinction between finite supply and infinite utility is fundamental to the economic model of any sound monetary system. Scarcity produces value. Utility sustains it.

[27] In distributed systems, the CAP theorem establishes that no network can simultaneously guarantee consistency, availability, and partition tolerance. Local certainty is achievable. Global certainty is not.

7. Conclusion

Pi is not a proposal. It is an implementation.

The network exists. The chain has begun. The rules are fixed in code, not in the intentions of any individual or institution. What follows from this point is determined by mathematics and by the participants who choose to engage with it — not by us.[28]

We make no promises regarding price, adoption, or utility. These are properties that emerge from use, not from declaration. A network is valuable because it is used. It is used because it is trusted. It is trusted because it is verifiable. Verification requires no faith.[29]

The problems Pi addresses are not new. The concentration of monetary authority, the fragility of trust-dependent systems, the exclusion of participants who lack institutional access — these are conditions that predate any digital network and will outlast any single attempt to address them. Pi is one attempt. Whether it succeeds is not within our power to determine.[30]

What is within our power is correctness. The protocol is specified here in full. It may be audited, verified, and challenged by any party at any time. We invite scrutiny. We do not ask for belief.[31]

The chain does not require our continued presence to function. It requires only nodes — participants willing to verify, willing to compute, willing to extend the sequence one block further.[32]

We have stated what Pi is. The rest is mathematics.

---

[28] The decentralisation of protocol governance is a design property, not a rhetorical position. No individual or organisation holds administrative access to the Pi network.

[29] Value in monetary systems is an emergent social property. It cannot be assigned by the issuer. Historical attempts to do so illustrate the limits of declared value.

[30] The conditions motivating the development of peer-to-peer currency systems are documented extensively in the literature of monetary history and distributed systems theory.

[31] Full protocol specification is provided in the technical appendix. All parameters are verifiable against the reference implementation.

[32] Network persistence requires only that the cost of participation remains below the value of the block reward plus transaction fees for at least one honest participant.

Technical Appendix

A. Protocol Parameters

The following parameters are fixed at the consensus layer and may not be altered by any participant or coalition of participants.

---

[3] Maximum supply: 21,000,000 units. This parameter is enforced at genesis and propagated through every subsequent block.

[1] Block interval: 600 seconds. Difficulty recalibrates every 2,016 blocks to maintain this target under variable hashrate conditions.

[4] Initial block subsidy: 50 PI. Halving occurs at block 210,000 intervals. The geometric series converges to the maximum supply asymptotically.

[1] Cryptographic primitive: SHA-256 (double round). No preimage attack is known. The security assumption is computationally binding.

[5] Signature scheme: ECDSA over secp256k1. Private keys are 256-bit integers drawn from a cryptographically secure entropy source.

[9] Network identifier: distinguished from all prior proof-of-work networks by genesis block hash. Replay protection is inherent to the chain state.

[2] Minimum transaction output: 1 unit (indivisible at the base layer). Fee market emerges organically from block space scarcity.

[6] Merkle tree structure: binary, SHA-256 double-round at each node. Transaction inclusion proofs are O(log n) in block size.

[5] Script system: non-Turing-complete stack machine. Expressible conditions: pay-to-public-key-hash, multisignature, time-locked outputs.

[3] Peer discovery: DNS seeding at initialisation, thereafter maintained by the addr message propagation protocol.

B. Verification Parameters

The following integrity verification string is provided for reference implementation validation. Conforming implementations must reproduce this value exactly.

Protocol checksum [SHA-256/AES-256-CBC]:

```
5b8269d41b22c7351827cd064e8da8b25ff2de42b7cd8fb33be9e4e618c8d9f1
```

This value encodes protocol state as of genesis. It is not decorative. Its derivation is left as an exercise.

C. Mathematical Constants

---

[5] The ratio of a circle's circumference to its diameter is an irrational transcendental number: 3.14159265358979323846264338327950288...

[8] The natural logarithm base e = 2.71828182845904523536... appears in the formula e^(iπ) + 1 = 0, which relates the five fundamental constants of mathematics.

[9] The golden ratio φ = 1.61803398874989484820... emerges from the Fibonacci sequence and appears throughout natural structures.

[7] Lambert (1761) proved π irrational. Lindemann (1882) proved π transcendental. Neither proof requires knowledge of π's decimal expansion.

[9] Archimedes bounded π between 223/71 and 22/7 using inscribed and circumscribed polygons. This was the first rigorous computational approach.

[3] The digits of π pass all known tests for normality. No pattern has been discovered. No pattern is expected.

[2] The SHA-256 compression function processes 512-bit message blocks. The output is indistinguishable from random under all known attacks.

[3] AES-256-CBC with a deterministic initialisation vector derived from the key material itself produces a reproducible ciphertext for any fixed plaintext.

[8] The composition of SHA-256 and AES-256 under a shared key schedule derived from an irrational constant is not known to be vulnerable to any algebraic attack.

[4] All parameters in this appendix are deterministic. Given the genesis block, all values are reproducible by any independent implementation.

[6] This document contains no errors. If a value appears anomalous, examine it more carefully.