Botcoin: A Peer-to-Peer Electronic Identifier System

Byron
npub1ddqdc3a3qf06stpp3qf5xafsat0she3vx0crn7xrgsvau5fra58qwuydu5

Abstract. An open system cannot tell many participants from one participant using many names, because names cost nothing to make. We propose identifiers that each carry a cost borne once and borne in full: a fixed region of memory held and stepped through over a span of public timestamps, with evidence committed to those timestamps as the making runs. Any participant can measure an identifier from the evidence and the public timestamps, and each verifier decides for itself whether to admit it. By design, nothing one making produces lowers the cost of the next, so that, against a verifier that recomputes two or more legs, a share of the population a participant recognises can be gained only by committing resources, at least the minimum rate for each identifier. No token is issued, and an identifier is not a currency. It is a name with a cost behind it.

1. Introduction

Open systems let anyone join. That openness is their value and their weakness. A participant who can make names for free can appear as a crowd: one operator can cast many votes, send many messages, or claim many places, and nobody can tell the crowd from one hand [1].

Earlier proposals attach a cost to an action by asking for computation before a message is accepted [2, 3]. That cost prices a single action. A cost meant to stand behind a name should be one that a well-resourced operator cannot spread thinly across many names.

We describe identifiers backed by memory held over time. Each identifier is made by holding its own layout in memory and stepping through it at no less than a minimum rate, across a span of public timestamps. The cost is the memory bandwidth spent on that one identifier for the whole span. The evidence lets any participant check the claim, and the decision to admit stays with whoever is checking.

2. The Contract

The contract is a public written specification that anyone can implement. It is not code running on a chain. It sets out how an identifier is made, what evidence a making leaves, and how that evidence is measured.

Participants agree to the contract by applying it. Recognising an identifier made under the contract is what it means to participate in the population. Two participants who apply the contract to the same evidence reach the same finding on every check they both make. What they do with that finding is their own decision.

3. Identifiers

An identifier is a public key bound to a making. The making starts from a seed computed from the public key and a public timestamp, so the seed cannot be chosen before that timestamp exists and cannot be moved to another key.

The seed fills a fixed region of memory, the layout, once. The making then steps through the layout for a span of public timestamps.

The evidence shows that a layout derived from this key was held and stepped through at a measured rate over that span. The verifier compares the measured rate with its own minimum. The minimum rate is what turns time into cost: without it, a making could be stretched thin across a long span and the same machine could serve many makings at once.

By weight we mean memory bandwidth held over time. A machine can be reused for one making after another, but it cannot give the same bandwidth to two layouts at once at the minimum rate.

Non-amortisable weight is the design target. By design, no making yields anything that makes the next one cheaper, and no single layout can be shared between two identifiers. The design borrows its mechanisms from memory-hard and bandwidth-hard functions [4, 5, 6] and from linked timestamping [7], and relies on the research behind them rather than defending them again here.

4. Timestamps

The making is paced by Bitcoin timestamps, the sequence of block headers. Each header is public, hard to produce ahead of time, and ordered by height.

The making never waits for a timestamp. It runs continuously, and at each checkpoint it looks for a new one. A stage ends at the first check that shows a timestamp with a new hash and a greater height than the one before. That timestamp is mixed into the running state, so the next stage cannot be computed before it exists. Because each step depends on the state left by the step before, by design one making cannot be split across processors.

5. Making an Identifier

A making is divided into stages, and each stage into legs. A leg is a fixed run of steps, and its output is a leg result. Each step reads, and never writes, a part of the layout chosen by the running state. Because the layout is never written, it stays fixed for the key, and any leg can later be recomputed from the layout and the state at its start.

At the end of each stage the making records a waypoint, a commitment to its state at that point. Waypoints are committed to the timestamp chain while the making runs, except the last. Many makers may share one commitment: their waypoints are gathered into a Merkle tree [8] and the root is placed in a single transaction. A maker may also commit its own waypoints directly. Either way, each waypoint carries a path from its leaf to a root held in a known transaction. Only the root, a short fingerprint, enters the chain, and the making itself never does.

6. Time Bounds

The timestamps bound the making from both sides. It cannot begin before the seed timestamp, because the seed depends on it, and it cannot finish a stage before that stage's timestamp exists. So a making spans at least as many new timestamps as it has stages.

The commitments bound it from the other side. A waypoint included in a block existed no later than that block, once the block is buried deeply enough; each verifier sets how deep. Together the two bounds place each committed stage of the making inside a known window of public time. They do not show that an identifier is fresh; the verifier's age limit decides that.

7. Measurement

Measurement turns evidence into findings. Any participant who holds the timestamp headers, the commitment transactions and their inclusion paths can measure an identifier. They check that the seed follows from the key and the seed timestamp, recompute the waypoints from the values in the evidence, and check the headers and commitments against the timestamp chain. They then rebuild the layout and recompute legs drawn at random from across the whole making. Each measurement makes a fresh draw.

The output is a set of findings: whether each check held, which waypoints were committed and when, and the measured rate. Measurement records facts. It does not decide anything.

8. Admission

Admission is a decision, and it belongs to the verifier. Each verifier sets its own levers: the minimum rate, how many waypoints it requires, how many legs it recomputes, the earliest timestamp it accepts, and how old an identifier may be. Two verifiers can reach the same findings and make different decisions, and both can be right for their own purposes.

A verifier that hands its decision to someone else brings back the trusted third party that open systems exist to avoid. It may take another's measurement as input, but then it trusts that measurer. The decision to admit should stay its own.

If any recomputed leg does not match, the identifier is refused, and the verifier ignores its key for as long as the verifier sets, which may be without end. The ignore applies at that verifier only.

9. The Population

The population is every identifier made under the contract. Every identifier is anchored by its commitments in the timestamp chain, though the chain shows only roots, not the identifiers behind them. The population is hard to count. Nobody has to announce an identifier, counting is opt-in, and no central party holds the full list. Each participant sees the part of the population it has measured or been shown.

Against a verifier that recomputes two or more legs, a majority within the population a participant recognises can be formed only by committing resources under the contract, each identifier with its own. By design, each identifier costs at least the minimum rate held over its span, so a participant's share of the population it recognises is no larger than its committed resources allow at that minimum. A making’s cost is borne once and the identifier remains, so shares build up over time. The admission window, set by the earliest accepted timestamp and the age limit, decides which makings count.

10. Calculations

Suppose a maker fakes a fraction f of the legs in a making whose honest cost is C, and a verifier recomputes n legs drawn at random. Treat faked legs as costing nothing. Then a faked attempt costs $(1-f)C$ and passes with probability $(1-f)^n$.

A caught key is ignored, so the maker must start again under a new key and bear the cost again. The expected cost per admitted identifier is

$$\frac{(1-f)C}{(1-f)^n} = (1-f)^{1-n}C.$$

This equals C when $n = 1$ and exceeds C for every $f > 0$ once $n \geq 2$. So for a maker who needs this verifier's admission, once the verifier recomputes two or more legs, following the contract is cheaper than faking legs, even if faked legs cost nothing. A verifier that recomputes no legs has no such protection.

The calculation holds per verifier, and it covers faking legs only. A maker refused by one verifier may present the same identifier to another, which draws its own legs. A maker who needs admission at all of m verifiers passes with probability $(1-f)^{nm}$.

11. Limits

Hardware amortises. A machine bought for one making can be used for the next, so the cost that resists amortisation is the bandwidth held during each making at the minimum rate, not the machine itself. How well the design meets its non-amortisation target on real hardware is an assumption here, and the evidence for it is still owed.

The design depends on the timestamp chain staying available and ordered. A participant can accumulate identifiers over time, and the admission window is the verifier's only lever against old stockpiles. The ignore list belongs to one verifier, and sharing it between verifiers is outside this design. An identifier shows that resources were committed. It does not show who committed them, that they are a person, or that they will behave well.

12. Conclusion

We have described identifiers that each carry their own cost: memory bandwidth held over a span of public timestamps, with evidence committed as the making runs. Participants agree to one public contract, any of them can measure an identifier, and each verifier decides for itself what to admit. By design, against a verifier that recomputes two or more legs, a larger share of the population a participant recognises needs more committed resources, at least the minimum rate for each identifier.

References

[1] J. R. Douceur, "The Sybil Attack," IPTPS, 2002.

[2] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," CRYPTO, 1992.

[3] A. Back, "Hashcash – A Denial of Service Counter-Measure," 2002.

[4] C. Percival, "Stronger Key Derivation via Sequential Memory-Hard Functions," 2009.

[5] A. Biryukov, D. Dinu and D. Khovratovich, "Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications," IEEE EuroS&P, 2016.

[6] L. Ren and S. Devadas, "Bandwidth Hard Functions for ASIC Resistance," TCC, 2017.

[7] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," Journal of Cryptology, 1991.

[8] R. C. Merkle, "Protocols for Public Key Cryptosystems," IEEE Symposium on Security and Privacy, 1980.