A Simple Proof of Sybil-Proof

Lancashire, David
david@saito.tech

Parris, Richard
richard@saito.tech

February 3, 2025

Abstract

Existing literature in computer science argues the *Red Balloons sybilling problem* is unsolvable in permissionless networks. It specifically claims that no sybil-proof reward schemes exist in such networks where information propagation without self-cloning is a dominant strategy for all such nodes in a three-or-less hop routing path. This paper offers a counter-proof to this claim, formally proving the existence of a mechanism that achieves sybil-proofness in a three-hop path.

INTRODUCTION

The *Red Balloons sybilling problem* is a known issue that impedes the ability of permissionless blockchains to offer routing payouts. In networks like Bitcoin, this problem exists because rational nodes will avoid sharing unconfirmed transactions to minimize competition for collection of their fees. Offering a payout to routing nodes can overcome this problem, but creates a counter-incentive for nodes to sybil transaction routing paths with pseudoanonymous identities ("self-cloning") to maximize their chance of winning the routing payout.

In their seminal paper *On Bitcoin and Red Balloons* [?], Babaioff and his co-authors claim the formal impossibility of solving this problem, offering a mathematical proof they write shows "there is no Sybil-proof reward scheme in which information propagation and no duplication are dominant strategy for all nodes at depth 3 or less."

Validation of the Babaioff paper seems to have come through the extension of their approach into related domains like query-incentive networks [?] [?]. Where other academics have proposed mechanisms to strengthen sybil-resistance, their attempts have also notably sacrificed the permissionless properties of the Bitcoin design, introducing mechanisms that limit which participants can propose blocks [?], or adding mechanisms of closure which introduce new attacks on participant selection and ranking mechanisms [?].

To our knowledge, only one paper has attempted to offer a formal counter-proof [?], arguing that the incentive to sybil can be eliminated if we permit nodes to extract arbitrary amounts of value in a single hop. Whether this can be considered a proper solution is an open question, but it is problematic in the way it requires viewing the underlying problem as the creation of superfluous routing hops rather than the inability of the payment mechanism to contain value-extraction from self-interested routers.

The remainder of this paper thus takes up the task of offering a simple proof of sybil-proofness directly, first describing a simple mechanism with a routing tax, and then showing how it avoids the impossibility trap outlined by Babaioff, creating a permissionless reward scheme in which "information propagation without self-cloning" is the dominant strategy for all participants.

PART I: ASSUMPTIONS

A routing-tax mechanism requires five basic assumptions:

1. all unconfirmed transactions provide block-generating "routing work" to the nodes which possess them, with such work consisting of the value of the transaction fee halved by each additional hop beyond the first that the transaction has taken to reach the node in question.
2. all valid blocks must contain enough "routing work" to meet a consensus-imposed difficulty requirement; nodes that reach this difficulty quota first are able to propose blocks before others.
3. when a block is produced, 50% of the fees contained in its transactions are destroyed with the remaining 50% allocated to a node chosen from the routing network.
4. the routing node is chosen by first picking a random transaction from the block, with each transaction's likelihood of being selected being proportional to its share of the fees contributed to the block. A node is then chosen from the routing path of the transaction, where each node's likelihood of selection is proportional to its share of the aggregate routing work (calculated as per item 1 above) the transaction provided to the block producer.
5. a simple longest-chain rule determines which blocks are confirmed and which blocks are orphaned.

In mechanisms with this design the dominant strategy for any node holding a transaction it suspects to be possessed by at least one other node is to propagate it without self-cloning. Until we reach the point we can formally establish that users are incentivized to broadcast their transactions to multiple nodes, we ask readers to treat this assumption as a design parameter as well.

PART II: THE PROOF

The Babaioff paper starts by deriving an equation it asserts must hold in Bitcoin-style networks if propagating transactions is a dominant strategy. It then demonstrates via a proof-by-contradiction that self-cloning is rational as long as this equation holds. Our counter-proof works in the reverse order. We first demonstrate that self-cloning is an irrational strategy that lowers the expected income of those who do it. We then show that nodes nonetheless benefit from propagating transactions regardless of whether their peers do.

To establish the income equations necessary to show that self-cloning is irrational, we start with some preliminary definitions.

Routing Work Definitions:

We define the following variables:

f = raw transaction fee (in a transaction)
H = total hops in routing path
n = nth hop in routing path of length H

Assumptions (1.1), (1.3) and (1.4) give us the following equations:

w(n) represents the amount of routing work available to a node at hop n. As per (1.1), this is the value of the transaction fee halved with every hop beyond the first that the transaction has taken into the network:

$$w(n) = f \cdot 2^{1-n}$$

W is the total amount of routing work held by all nodes in the routing path of a transaction, where the transaction has a routing path with a total length of H hops.

$$W = \sum_{n=1}^{H} \frac{f}{2^{n-1}}$$

$$W = 2f\left(1 - 2^{-H}\right)$$

We refer to the "utility" that a node holds *in a transaction* as the probability of the node being selected as the winning router if the transaction is selected as the winning transaction in the routing lottery. According to (1.4) that is:

$$u(n, H) = \frac{w(n)}{W}$$

$$u(n, H) = \frac{f}{2^{n-1}} \div \left(2f \cdot \left(1 - 2^{-H}\right)\right)$$

$$u(n, H) = \frac{2^{H-n}}{2^{H}-1}$$

We can now develop the income equation for non-sybils (nodes which produce a block without self-cloning) and then for nodes which engage in self-cloning. These two equations will be compared to show self-cloning is strictly irrational.

Income Equation for Non-Sybils:

The income equation for a non-sybilling node at hop n is the routing payout multiplied by the utility that a node holds in the transactions in the block.

To simplify our proof we start with a block containing a single transaction. In a non-sybil environment the probability of this transaction being selected is 1:

$$i_{\text{non-sybil}}(f, n, H) = \frac{f}{2} \cdot 1 \cdot u(n, H)$$

$$i_{\text{non-sybil}}(f, n, H) = \frac{f}{2} \; \cdot u(n, H)$$

$$i_{\text{non-sybil}}(f, n, H) = \frac{f}{2} \cdot \frac{\frac{1}{2^{n-1}}}{\frac{1}{2^{H-1}} \cdot (2^{H}-1)}$$

$$i_{\text{non-sybil}}(f, n, H) = f \cdot \frac{2^{(H-n-1)}}{2^{H}-1}$$

We have $n = H$ in the non-sybil case:

$$i_{\text{non-sybil}}(f, n) = \frac{f}{2} \cdot \frac{\frac{1}{2^{n-1}}}{\frac{1}{2^{n-1}} \cdot (2^{n}-1)}$$

$$i_{\text{non-sybil}}(f, n) = \frac{f}{2(2^{n}-1)}$$

Given a transaction with a fee of 100 units, a second-hop node following a non-sybilling strategy can expect to earn 16.67 in income. This can be confirmed via the tables in APPENDIX A, or intuitively since with 50% of the block reward destroyed (1.3) the expected income for a second-hop node is ~33.3% of 50% or ~16.67%.

Income Equation for Sybils:

Calculating the income equation for sybils who add an additional hop before producing a block requires understanding how the routing-tax penalizes attackers in practice.

It follows from (1.1), (1.2) and (1.4) that any attacker adding an additional hop to a transaction halves the amount of routing work available in that transaction for production of a block. This puts the attacker at a strict disadvantage vis-a-vis any competitors at the same routing depth.

Compensating for this decline and re-establishing a competitive position in block production requires the attacker to add a supplementary transaction containing real fees. The amount of fees the attacker must contribute to generate an equivalent amount of first-hop routing work to that lost by their act of sybilling an existing transaction can be calculated as follows:

$$sw(n, f) = \frac{f}{2^{n}}$$

The addition of a supplementary transaction with supplementary work increases the number of transactions in the block. For clarity, we refer to the original transaction which the attacker is sybilling as tx_{syb} and the supplementary transaction added by the attacker as tx_{sup}.

The addition of tx_{sup} to the block changes the total fees (F) in that block.

$$F = f + sw(n, f)$$

$$F = f + \frac{f}{2^{n}}$$

The probability of each transaction being chosen has also changed. Of the two transactions competing for selection in the payment lottery (1.4), the attacker-sybilled transaction (tx_{syb}) has the larger chance of selection as it contributes more in fees.

$$p(tx_{syb}) = \frac{f}{f+\frac{f}{2^{n}}}$$

The supplementary transaction has a smaller chance of selection:

$$p(tx_{sup}) = \frac{\frac{f}{2^{n}}}{f+\frac{f}{2^{n}}}$$

Factor out f for fee-independent probabilities:

$$p(tx_{sup}) = \frac{\frac{1}{2^{n}}}{1+\frac{1}{2^{n}}}$$

$$p(tx_{syb}) = \frac{1}{1+\frac{1}{2^{n}}}$$

To calculate the income equation for the attacker we must modify the probability of each transaction being selected by the utility the attacker holds within each transaction. This provides the combined chance that the attacker has of receiving payment - of a transaction that pays them being selected and then of them being selected as the winning node from that transaction's routing path. We refer to this figure as the "lottery adjusted utility" of the attacker and define it for the supplementary transaction as lau_{sup} and in the sybilled (original) transaction as lau_{syb}. Note that these probabilities represent the chance of the attacker being selected rather than their expected income as only a portion of the block reward will be distributed as the routing payout as per (1.3).

In the supplementary transaction the attacker is the only routing node and thus earns the payout with a probability of 1 if the supplementary transaction is chosen, making lau_{sup} equal to $p(tx_{sup})$.

$$lau_{sup} = \frac{\frac{f}{2^n}}{f+\frac{f}{2^n}} \cdot 1$$

In the sybilled transaction (tx_{syb}) the attacker controls the final two hops in the routing path.

$$lau_{syb} = \frac{f}{f+\frac{f}{2^n}} \cdot su(n)$$

With *su(n)* representing the sum of the attacker's utility in the final two hops, we can define it as follows.

$$su(n) = u(n,H) + u(n+1,H)$$

Because adding a sybil hop increases the length of the routing path by 1, H is *n+1*:

$$su(n) = u(n,n+1) + u(n+1,n+1)$$

given

$$u(n,H) = \frac{f}{2^{n-1}} \div \left(\frac{f}{2^{H-1}} \times \left(2^{H-1} \times 2 - 1\right)\right)$$

$$u(n,H) = \frac{\frac{f}{2^{n-1}}}{\frac{f}{2^n} \cdot (2^{n+1}-1)}$$

$$u(n,H) = \frac{f/2^n}{\left(\frac{f}{2^n}\right) \cdot (2^{n+1}-1)}$$

we have

$$su(n) = u(n,n+1) + u(n+1,n+1)$$

$$su(n) = \frac{\frac{f}{2^{n-1}}}{\frac{f}{2^n}(2^{n+1}-1)} + \frac{\frac{f}{2^n}}{\frac{f}{2^n}(2^{n+1}-1)}$$

which simplifies to

$$su(n) = \frac{\frac{1}{2^{n-1}}}{\frac{1}{2^n}(2^{n+1}-1)} + \frac{\frac{1}{2^n}}{\frac{1}{2^n}(2^{n+1}-1)}$$

$$su(n) = \frac{3}{2^{n+1}-1}$$

This gives us the "lottery adjusted utility" of the sybilled transaction:

$$lau_{syb} = \frac{1}{1+\frac{1}{2^n}} \left(\frac{\frac{1}{2^{n-1}}}{\frac{1}{2^n}(2^{n+1}-1)} + \frac{\frac{1}{2^n}}{\frac{1}{2^n}(2^{n+1}-1)} \right)$$

$$lau_{syb} = \frac{3}{\left(1+\frac{1}{2^n}\right)(2^{n+1}-1)}$$

We add the lottery-adjusted utilities of the two transactions in the block for the probability of the attacker winning the router payout:

$$lau_{syb} + lau_{sup} = \frac{3}{\left(1+\frac{1}{2^n}\right)(2^{n+1}-1)} + \frac{\frac{1}{2^n}}{1+\frac{1}{2^n}}$$

Multiply this by half of the total fees in the block to calculate expected income from sybilling:

$$i_{\text{sybil}}(F,n) = \frac{F}{2} \cdot (lau_{syb} + lau_{sup})$$

$$i_{\text{sybil}}(F,n) = \frac{F}{2} \cdot \left(\frac{3}{\left(1+\frac{1}{2^n}\right)(2^{n+1}-1)} + \frac{\frac{1}{2^n}}{1+\frac{1}{2^n}} \right)$$

Net profit is the attacker's income minus the cost of the supplementary work they contributed to the block:

$$\frac{f}{2^n}$$

For a second-hop routing node the profit from a sybilling strategy is 8.93 if f = 100. This is lower than the profit of 16.67 expected from a non-sybilling strategy. Self-cloning is an inferior strategy.

Transaction Propagation:

We now demonstrate that nodes facing this routing-tax mechanism nonetheless have a rational interest in propagating transactions to peers. To do this, assume the existence of two nodes A1 and B1 which receive a transaction from a user as first-hop recipients.

A1 and B1 have the option of propagating this transaction to their children A2 and B2. The existence of a routing tax prevents us from assuming that all nodes have an equal probability of producing the next block and thus working out the income equations for the parents and children. We can nonetheless make headway by observing there are six scenarios in play:

1. A1 outcompetes B1 and all children
2. B1 outcompetes A1 and all children
3. A2 outcompetes A1 and all others
4. A2 outcompetes B1 and all others
5. B2 outcompetes A1 and all others
6. B2 outcompetes B1 and all others

If both A1 and B1 hoard only the first two outcomes are possible and A1 and B1 expect to claim payment with equal probability:

$$hh = u(1,1) \cdot 0.5$$

If A1 shares but B1 hoards scenarios 3 and 4 become possible: there is an unknown probability that A2 will produce a block before either A1 or B1.

Let us define this unknown probability as x. While we do not know the value of x we expect it to be similar for A2 and B2 given the symmetrical nature of the incentive structure facing A1 or B1. We can also observe that whatever the value of x both A1 and B1 expect to outcompete each other with symmetrical probabilities as A1 sharing this transaction with its children does not disadvantage A1 in its direct competition with B1.

While A1 thus expects the losses from sharing to be distributed evenly between A1 and B1, A1 nonetheless has private benefits which accrue to it privately in the event that A2 produces the next block. This gives us the following income equations for A1 and B1 if we assume that A1 shares and B1 hoards given the unknown probability x that B2 produces the next block.

$$A1 = \left((1-x) \cdot u(2,2) \cdot \tfrac{1}{2}\right) + (x \cdot u(2,3) \cdot 1)$$

$$B1 = \left((1-x) \cdot u(2,2) \cdot \tfrac{1}{2}\right) + 0$$

If both A1 and B1 propagate transactions all six scenarios are in play. We cannot assume that the probability of either A2 or B2 creating the next block is equal to the probability of either child producing one independently, so must define another variable y to represent the probability that A2 or B2 collectively beat A1 and B1 to production of the next block.

This gives us the following income equations for A1 and B1 if both nodes propagate transactions to their children.

$$A1 = ((1-y)\cdot u(2,2)\cdot 0.5) + ((y)\cdot u(2,3)\cdot 0.5)$$

$$B1 = ((1-y)\cdot u(2,2)\cdot 0.5) + ((y)\cdot u(2,3)\cdot 0.5)$$

Comparing these hoarding and sharing income equations suffices to demonstrate the dominant strategy for nodes is to propagation transactions.

If B1 follows a hoarding strategy then:

A1 hoarding < A1 propagating:

$$u(2,2)\cdot 0.5 < ((1-x)\cdot u(2,2)\cdot 0.5)+(x\cdot u(2,3)\cdot 1)$$

$$\tfrac{1}{3}\cdot 0.5 < ((1-x)\cdot\tfrac{1}{3}\cdot 0.5) + ((x)\cdot\tfrac{2}{7})$$

$$\tfrac{1}{6} < \left((1-x)\cdot\tfrac{1}{6}+\tfrac{2x}{7}\right)$$

for x = 0:

$$\tfrac{1}{6} < \left(\tfrac{1}{6}+\tfrac{2x}{7}\right)$$

$$\tfrac{1}{6} < \left(\tfrac{1}{6}+0\right)$$

for x = 1:

$$\tfrac{1}{6} < \left((1-x) * \tfrac{1}{6} + \left(\tfrac{2x}{7}\right)\right)$$

$$\tfrac{1}{6} < \left(0\times\tfrac{1}{6}\right) + \left(\tfrac{2}{7}\right)$$

If B1 shares instead of hoarding:

A1 hoarding < A1 propagating:

$$(1-x)\tfrac{1}{6} < (1-y)\,\tfrac{1}{6} + y\tfrac{2}{14}$$

$$x > \tfrac{y}{7}$$

This shows a relationship between x (the probability that A2 will produce the next block) and y (the probability that A2 or B2 will collectively next block) that must hold for hoarding to be a dominant strategy. Specifically, we discover that if B1 propagates A1 is also incentivized to propagate unless doing so makes it 700% more likely that both children having access to the transaction will lead them to produce a block than if a single child alone was handed access.

for x = 0.1 y = 0.7

$$\left((1-x)\,\tfrac{1}{6}\right) < \left((1-y)\,\tfrac{1}{6}\right) + \left((y)\,\tfrac{2}{14}\right)$$

$$\tfrac{0.9}{6} < \tfrac{0.3}{6} + \tfrac{0.7}{14}$$

$$0.15 < 0.05 + 0.1$$

for x = 0.1 y = 0.8

$$\left(0.9\cdot\tfrac{1}{6}\right) < \left(0.2\cdot\tfrac{1}{6}\right) + \left(0.8\cdot\tfrac{2}{14}\right)$$

$$0.15 < 0.0333 + 0.1142$$

$$0.15 < 0.1475$$

and for x = 0.1 y = 0.6

$$\left(0.9\cdot\tfrac{1}{6}\right) < \left(0.4\cdot\tfrac{1}{6}\right) + \left(0.6\cdot\tfrac{2}{14}\right)$$

$$0.15 < 0.0666 + 0.0857$$

$$0.15 < 0.1523$$

The symmetrical nature of the income equations affecting A2 and B2 and their children make it impossible for x to be this much larger than y. Given that the probability of each node producing a block depends on its independent transaction flow, there is no formally sound explanation for why both A1 and B1 sharing the same transaction with their two children will result in a 700% greater probability of those children producing a block than if the same transaction is independently shared by either node with either child alone.

We therefore observe a formal violation of the inclusion-exclusion principle in statistics, with its requirement the sum of the probability of two independent events be the sum of the probabilities of each event happening on its own.

It follows that information propagation is a mathematically dominant strategy. Each node may increase its income relative to its hoarding peers by propagating unconfirmed transactions to child nodes, who assist their parents in outcompeting their competitors at similar routing-depths. Peers who do not share face disproportionate losses unless they defend against this by propagating transactions to their own children to level the playing field.

This logic applies to every node at every position in every routing path.

As all nodes have an incentive to propagate without self-cloning, we have a Sybil-proof reward scheme in which information propagation without self-cloning is the dominant strategy for all nodes.

CONCLUSION

The sybil problem identified by Babaioff exists because nodes which add hops increase their private income even if doing so has suboptimal consequences for the network as a whole. As with other collective action problems [**?**], unless the underlying problem can be fixed on the incentive level, the result is necessarily a market failure [**?**] which can be eliminated only through the containment of extractive behavior through the use of mechanisms which restrict the freedom of participation of participants in the system [**?**] [**?**].

The *routing-tax* mechanism described in this paper fixes the sybilling problem by inverting the collective action problem that creates the incentive to sybil in the first place. With a routing-tax in place, while all early-hop nodes would collectively prefer to hoard transactions and minimize their children earning any income at all, each node suddenly has a stronger individual interest in defecting from the hoarding equilibrium and propagating transactions its own children. The direction of defection shifts from hoarding transactions to propagating them.

With this in mind, we close by returning readers to the assumption made in Part 1 that users desiring sybil-free transactions will forward their transactions to two different first-hop nodes. It can now be seen why such behavior is in the rational interest of network users. Users who follow this strategy benefit from faster confirmations at lower fees than those who do not, so users who do not propagate to multiple first-hop nodes have an incentive to start doing so, while those who are already following this strategy have no incentive to stop.

APPENDIX A

Table 1: Routing Work (fee = 1)

Hop	Routing Work Provided	Aggregate Routing Work in Path	$W = 2f\left(1 - 2^{-H}\right)$
1	1.000000000	1.000000000	1.000000000
2	0.500000000	1.500000000	1.500000000
3	0.250000000	1.750000000	1.750000000
4	0.125000000	1.875000000	1.875000000
5	0.062500000	1.937500000	1.937500000
6	0.031250000	1.968750000	1.968750000
7	0.015625000	1.984375000	1.984375000
8	0.007812500	1.992187500	1.992187500
9	0.003906250	1.996093750	1.996093750
10	0.001953125	1.998046875	1.998046875
11	0.000976563	1.999023438	1.999023438
12	0.000488281	1.999511719	1.999511719
13	0.000244141	1.999755859	1.999755859
14	0.000122070	1.999877930	1.999877930
15	0.000061035	1.999938965	1.999938965
16	0.000030518	1.999969482	1.999969482
17	0.000015259	1.999984741	1.999984741
18	0.000007629	1.999992371	1.999992371
19	0.000003815	1.999996185	1.999996185
20	0.000001907	1.999998093	1.999998093

Table 1 shows the amount of routing work a transaction with a fee of 1 provides to nodes in its routing path, along with the aggregate amount of routing work that will consequently exist in the transaction at the hop-depth specified. The final column demonstrates the algebraic equation derived and used to calculate the income equations for sybilling and non-sybilling nodes is correct.

Table 2: Probability of Node Winning Routing Payout (H=3)

Hop	Probability	Probability	$\frac{2^{H-n}}{2^H-1}$
1	4/7	0.571428571	0.571428571
2	2/7	0.285714286	0.285714286
3	1/7	0.142857143	0.142857143

Table 2 shows the probability of each of the three hops in a transaction with a 3-hop routing path winning the payout lottery if said transaction is randomly selected in the payment lottery. The final column confirms that the equation used to calculate the sybilling and non-sybilling income equations is correct.

Table 3: Probability of Node Winning Routing Payout (H=7)

Hop	Probability	Probability	$\frac{2^{H-n}}{2^H-1}$
1	64/127	0.503937008	0.503937008
2	32/127	0.251968504	0.251968504
3	16/127	0.125984252	0.125984252
4	8/127	0.062992126	0.062992126
5	4/127	0.031496063	0.031496063
6	2/127	0.015748031	0.015748031
7	1/127	0.007874016	0.007874016

Table 3 is an extended version of Table 2 which shows the probability of each of the seven hops in a transaction with a 7-hop routing path winning the payout lottery if said transaction is randomly selected in the payment lottery. The final column confirms that the equation used to calculate the sybilling and non-sybilling income equations is correct.

Table 4: Sybilling is Always Costly Routing Work (fee = 1)

hop/n	i(f, n)	sw(n+1,f)	$p(tx_{syb})$	$p(tx_{sup})$	lau_{syb}	lau_{sup}	$i_{\text{sybil}}(1, n)$ profit	
1	0.500000000	0.5000	0.666666667	0.333333333	0.666666667	1.000000000	0.250000000	-0.250000000
2	0.166666667	0.2500	0.800000000	0.200000000	0.342857143	0.542857143	0.089285714	-0.077380952
3	0.071428571	0.1250	0.888888889	0.111111111	0.177777778	0.288888889	0.037500000	-0.033928571
4	0.033333333	0.0625	0.941176471	0.058823529	0.091081594	0.149905123	0.017137097	-0.016196237
5	0.016129032	0.0313	0.969696970	0.030303030	0.046176046	0.076479076	0.008184524	-0.007944508
6	0.007936508	0.0156	0.984615385	0.015384615	0.023258631	0.038643247	0.003998524	-0.003937984
7	0.003937008	0.0078	0.992248062	0.007751938	0.011673507	0.019425445	0.001976103	-0.001960905
8	0.001960784	0.0039	0.996108949	0.003891051	0.005847998	0.009739048	0.000982296	-0.000978489
9	0.000978474	0.00200	0.998050682	0.001949318	0.002926835	0.004876153	0.000489713	-0.000488760
10	0.000488759	0.00098	0.999024390	0.000975610	0.001464130	0.002439739	0.000244498	-0.000244260
11	0.000244260	0.00049	0.999511957	0.000488043	0.000732243	0.001220286	0.000122160	-0.000122100
12	0.000122100	0.00024	0.999755919	0.000244081	0.000366166	0.000610247	0.000061058	-0.000061043
13	0.000061043	0.00012	0.999877945	0.000122055	0.000183094	0.000305150	0.000030523	-0.000030519
14	0.000030519	0.00006	0.999938969	0.000061031	0.000091550	0.000152581	0.000015260	-0.000015259
15	0.000015259	0.00003	0.999969483	0.000030517	0.000045776	0.000076292	0.000007630	-0.000007630
16	0.000007630	0.00001	0.999984741	0.000015259	0.000022888	0.000038147	0.000003815	-0.000003815
17	0.000003815	0.00000	0.999992371	0.000007629	0.000011444	0.000019073	0.000001907	-0.000001907
18	0.000001907	0.00000	0.999996185	0.000003815	0.000005722	0.000009537	0.000000954	-0.000000954
19	0.000000954	0.00000	0.999998093	0.000001907	0.000002861	0.000004768	0.000000477	-0.000000477
20	0.000000477	0.00000	0.999999046	0.000000954	0.000001431	0.000002384	0.000000238	-0.000000238

Table 4 quantifies the values referred to throughput the paper in a comprehensive reference for those seeking to empirically verify the algorithms included in this paper. It shows that the cost of adding an additional sybil-hop is always negative for the first twenty hops into the network. The values are easy to calculate for all subsequent hops, although those seeking a general solution are pointed to APPENDIX 2 which offers a formal proof that cost-of-attack holds for routing paths of arbitrary length.

APPENDIX B

What follows is a short but general proof that the cost of sybilling is higher than the cost of non-sybilling in routing paths of all arbitrary lengths. We start by observing the income equation for non-sybilling nodes:

$$I_{\text{non-sybil}}(f,n) = \frac{1}{2\left(2^n-1\right)}f$$

given

$w(f,n) = 2^{1-n}f$	amount of routing work that a transaction with fee f provides a node at hop n
k	number of sybil nodes
t	sybilled transaction
t'	transaction that the attacker must add to be able to produce the block
f	fee of transaction t
f'	fee of transaction t'

For t, the attacker is at hop i for all $i \in \{n, ..., n+k\}$.
For t', the attacker is at hop 1.

Block production constraint:

$$\begin{aligned}
& w(f,n+k) + w(f',1) \geqslant w(f,n) \\
& 2^{1-(n+k)}f + 2^{1-1}f' \geqslant 2^{1-n}f \\
& f' \geqslant 2^{1-n}f - 2^{1-(n+k)}f \\
& \boxed{f' \geqslant 2^{1-n}\left(1-2^{-k}\right)f}
\end{aligned}$$

Probabilities:

$$\mathbb{P}\left[t \text{ is chosen}\right] = \frac{f}{f+f'} \qquad \mathbb{P}\left[t' \text{ is chosen}\right] = \frac{f'}{f+f'}$$

$$\mathbb{P}\left[\text{hop chosen} \in \{n, ..., n+k\} \mid t \text{ is chosen}\right] = \frac{\sum\limits_{i=n}^{n+k} w(f,i)}{\sum\limits_{i=1}^{n+k} w(f,i)} = \frac{\sum\limits_{i=n}^{n+k} 2^{1-i}f}{\sum\limits_{i=1}^{n+k} 2^{1-i}f} = \frac{\sum\limits_{i=n}^{n+k} 2^{-i}}{\sum\limits_{i=1}^{n+k} 2^{-i}} = \frac{\frac{2^{-n}-2^{-(n+k+1)}}{1-2^{-1}}}{\frac{2^{-1}-2^{-(n+k+1)}}{1-2^{-1}}} = \frac{2^{-n}-2^{-(n+k+1)}}{2^{-1}-2^{-(n+k+1)}}$$

$$\mathbb{P}\left[\text{hop chosen} = 1 \mid t' \text{ is chosen}\right] = \frac{w(f',1)}{w(f',1)} = 1$$

$$\begin{aligned}
& \mathbb{P}\left[\text{attacker wins the block reward}\right] \\
& = \mathbb{P}\left[\text{hop chosen} \in \{n, ..., n+k\} \mid t \text{ is chosen}\right]\ \mathbb{P}\left[t \text{ is chosen}\right]\ +\ \mathbb{P}\left[\text{hop chosen} = 1 \mid t' \text{ is chosen}\right]\ \mathbb{P}\left[t' \text{ is chosen}\right] \\
& = \frac{2^{-n}-2^{-(n+k+1)}}{2^{-1}-2^{-(n+k+1)}} \cdot \frac{f}{f+f'}\ +\ 1 \cdot \frac{f'}{f+f'} \\
& = \frac{1}{f+f'}\left(\frac{2^{-n}-2^{-(n+k+1)}}{2^{-1}-2^{-(n+k+1)}}f\ +\ f'\right)
\end{aligned}$$

Expected payout for the attacker:

$$\begin{aligned}
P(f,f',n,k) &= \mathbb{P}\left[\text{attacker wins the block reward}\right] \cdot \text{total router payout} \\
&= \frac{1}{f+f'}\left(\frac{2^{-n}-2^{-(n+k+1)}}{2^{-1}-2^{-(n+k+1)}}f\ +\ f'\right) \cdot 0.5(f+f') \\
&= \frac{1}{2}\left(\frac{2^{-n}-2^{-(n+k+1)}}{2^{-1}-2^{-(n+k+1)}}f\ +\ f'\right)
\end{aligned}$$

Expected income:

$$\begin{aligned}
I_{\text{sybil}}(f, f', n, k) &= P(f, f', n, k) - f' \\
&= \frac{1}{2}\left(\frac{2^{-n} - 2^{-(n+k+1)}}{2^{-1} - 2^{-(n+k+1)}} f \ + \ f'\right) - f' \\
&= \frac{1}{2} \cdot \frac{2^{-n} - 2^{-(n+k+1)}}{2^{-1} - 2^{-(n+k+1)}} f \ - \frac{1}{2} f' \\
&\leqslant \frac{1}{2} \cdot \frac{2^{-n} - 2^{-(n+k+1)}}{2^{-1} - 2^{-(n+k+1)}} f \ - \frac{1}{2} \cdot 2^{1-n}\left(1 - 2^{-k}\right) f \\
&= \left(\frac{1}{2} \cdot \frac{2^{-n} - 2^{-(n+k+1)}}{2^{-1} - 2^{-(n+k+1)}} - \frac{1}{2} \cdot 2^{1-n}\left(1 - 2^{-k}\right)\right) f
\end{aligned}$$

$$\begin{aligned}
I_{\text{non-sybil}}(f, n) - I_{\text{sybil}}(f, f', n, k) &\geqslant \frac{1}{2\left(2^n - 1\right)} f - \left(\frac{1}{2} \cdot \frac{2^{-n} - 2^{-(n+k+1)}}{2^{-1} - 2^{-(n+k+1)}} - \frac{1}{2} \cdot 2^{1-n}\left(1 - 2^{-k}\right)\right) f \\
&= \left(\frac{1}{2\left(2^n - 1\right)} - \frac{1}{2} \cdot \frac{2^{-n} - 2^{-n-k-1}}{2^{-1} - 2^{-n-k-1}} + \frac{1}{2} \cdot 2^{1-n}\left(1 - 2^{-k}\right)\right) f \\
&= \frac{2^{n+k}\left(2^n\left(2^k - 1\right) - 2\right) + 2^{k+1} + 2^{n+1} - 2}{2^{n+2k+1}\left(2^n - 1\right)\left(2^n - 2^{-k}\right)} f \\
&\geqslant \frac{2^{n+k}\left(2^1\left(2^1 - 1\right) - 2\right) + 2^{1+1} + 2^{1+1} - 2}{2^{n+2k+1}\left(2^n - 1\right)\left(2^n - 2^{-k}\right)} f \\
&= \frac{6}{2^{n+2k+1}\left(2^n - 1\right)\left(2^n - 2^{-k}\right)} f \\
&> 0
\end{aligned}$$

Conclusion:

For all f' such that $w(f, n{+}k) + w(f', 1) \geqslant w(f, n)$, we have: $\boxed{I_{\text{sybil}}(f, f', n, k) < I_{\text{non-sybil}}(f, n)}$.