


Defeating a Black Marble Flood Against Monero: Best Options for Ring Size and Transaction Fee

Draft v0.1
Rucknium 

May 28, 2024

1 Summary

Increasing Monero’s ring size and minimum transaction fee are two options for defeating black marble flooding. This document attempts to answer the question: Is it better to increase the ring size or the transaction fee, or some combination of the two? Cost-Effectiveness Analysis is used to analyze this question. It considers the additional costs imposed on transacting users and node operators compared to the benefit of stronger resistance to black marble flooding.

Consider an adversary with a daily budget of 12.5 XMR, five times higher than the daily expenditure of the suspected March 2024 black marble flooder. Given the constraints considered, the most cost-effective combination of defense parameters are ring size 60 and minimum 70 nanonero per byte fee. Effective ring size would be 22.8 if the adversary spent his entire budget every day. The 2in/2out reference transaction with ring size 60 would be about 140% larger than the transaction with current ring size 16. The user’s cost to send this transaction would be about 4.4 USD cents. The total time to verify all transactions in a block of normal transaction volume would increase from 0.5 seconds to 1.8 seconds. An unpruned node would grow 59 GB in a year instead of 25 GB. Pruned nodes would grow 14 GB instead of 8 GB.

2 Black marble flooding as a game

We will analyze the problem as a game with two players. One player aims to flood the Monero blockchain with black marble outputs. This player is limited by his budget. The other player aims to deter the first player, or at least limit the damage, by choosing minimum fee and ring size. This player is limited by the costs that fees and ring size impose of transacting users and node operators.

Sam is a privacy adversary. His goal is to reduce Monero’s effective ring size by flooding the Monero blockchain with black marble outputs that he owns. He has some budget b denominated in XMR to spend on transaction fees per block.

Alice wishes to defeat Sam. She can set Monero’s ring size and minimum transaction fee to try to accomplish her goal. Sam would have to spend more XMR per output if the minimum fee per byte were higher. A larger ring size would require Sam to own a larger share of outputs to achieve a specified effective ring size. (Without changing the minimum fee per byte, a larger ring size also requires Sam to spend more XMR to produce each output because transaction size is larger.)

Larger ring sizes and fees help Alice accomplish her goal of defeating Sam, but Alice cannot raise ring size and fee without limit. Users who send Monero transactions need to pay a higher fee when the minimum transaction fee is higher. Larger ring sizes mean that transactions are larger. At a given transaction volume, larger transactions

36 make the blockchain grow faster. People who operate Monero nodes need to store the blockchain on their storage
37 media such as Solid State Drives (SSDs). Alice needs to balance the benefit of greater defense against Sam against
38 the cost imposed on transacting users and node operators.

39 These are the factors on Alice’s mind:

- 40 • I do not know Sam’s budget b . I do not know what effective ring size he hopes to achieve. If I set ring size
41 and fee so that he cannot achieve his desired effective ring size with his budget b , he will choose not to flood
42 the blockchain with black marbles. This is the deterrence outcome.
- 43 • If I fail to deter Sam, at least I can hold him to a specific effective ring size when he spends his budget b . This
44 is the fallback outcome.
- 45 • I do not want to set ring size and transaction fee unnecessarily high because transacting users and node
46 operators pay higher costs when these parameters increase.

47 We will simplify the problem:

- 48 • Sam may actually change the budget he is willing to spend based on the effective ring size he is able to achieve.
49 In other words, Sam may have a tradeoff function between budget and effective ring size. We will ignore this
50 complication and assume that Sam’s budget is fixed, but unknown to Alice.
- 51 • We will use the fallback outcome to measure the effectiveness of Alice’s options. When the fallback outcome
52 is better for Alice, the deterrence outcome is more likely. Therefore, it is a little redundant to compute the
53 probability of the deterrence outcome as an effectiveness metric.
- 54 • Transaction volume by normal users is assumed to be constant and unaffected by changes in the transaction
55 fee. In other words, we will assume that the demand for Monero transactions is completely inelastic with
56 respect to transaction fee.
- 57 • We will assume that Sam’s black marble transactions are 1in/2out because the suspected black marble flood
58 of March 2024 used this type of transaction. Sam could produce black marble outputs more cheaply with
59 1in/16out transactions, but the flood transactions would be easier for an observer to identify.

60 Alice will use Cost-Effectiveness Analysis (CEA) to evaluate her ring size and fee options. Cost effectiveness is the
61 ratio of cost to effectiveness:

$$CE = \frac{\text{Cost}}{\text{Effectiveness}} \quad (1)$$

62 A lower value of CE is better. Alice must define cost and effectiveness as functions of ring size, transaction
63 fee, and the adversary’s budget. Let n be nominal ring size, f be the fee per byte in nanonero units, and b be the
64 adversary’s budget. Costs will be measured in terms of XMR per block.

65 The cost has two components: cost to transacting users and cost to node operators. The i th transaction has
66 some number of inputs and outputs. Changing the ring size n changes the total size of the i th transaction, which
67 affects the total minimum fee to send the transactions. And changing the minimum fee per byte changes the total
68 fee, of course. Let $w_i(n)$ be the weight of transaction i when ring size is n . When a transaction has two outputs,
69 transaction weight is equal to transaction size in bytes. Weight is larger than size when the number of outputs
70 is greater than two.¹ The block is assumed to contain an average set of transactions T . The average is based on
71 observed transactions confirmed on the blockchain in the four weeks before the March 2024 suspected black marble
72 flooding: February 5 – March 3. $C_u(f, n)$ is the aggregate users’ cost to send transactions for an average block:

¹See Section 7.3.2 of koe, Alonso, K. M., & Noether, S. (2020). *Zero to Monero: Second Edition*.

$$C_u(f, n) = \sum_{i \in T} f \cdot w_i(n) \quad (2)$$

73 The cost to node operators is a function of ring size. Node operators do not pay higher costs when the minimum
 74 transaction fee is higher. All units of computer storage in this document will be SI units, i.e. a kilobyte, megabyte,
 75 gigabyte and terabyte are 10^3 , 10^6 , 10^9 , and 10^{12} bytes, respectively. The retail price of one consumer 1 TB SATA
 76 SSD is about 1 XMR.² A node operator’s cost C_{SSD} to store one byte of Monero blockchain data is 10^{-12} XMR (a
 77 piconero). According to `monero.fail/map`, there were about 20,000 Monero nodes on the network in April 2024.
 78 Currently the minimum relay fee is 20,000 piconeros (20 nanoneros) per byte. Therefore, by coincidence Monero
 79 transactions pay for their own storage space on the node network when users pay the minimum fee per byte.

80 Let d_n , the number of nodes (daemons), be 20,000. $z_i(n)$ is the size of the i th transaction in the T set when
 81 ring size is n . The m is an adjustment parameter that raises or lowers total node operators’ costs by a linear factor
 82 to adjust for uncertainty about the true number of nodes and to add costs that are more difficult to compute like
 83 CPU and RAM use. In the analysis below m will be set to 2. We will assume that each node is an unpruned node
 84 that stores all transaction data in full. The total cost to node operators is the sum of the size of transactions in the
 85 T set multiplied by the storage cost on a single SSD, the number of nodes on the network, and the m adjustment
 86 parameter:

$$C_d(n, m) = m \cdot d_n \cdot C_{SSD} \cdot \sum_{i \in T} z_i(n) \quad (3)$$

87 Notice that $C_d(n, m)$ is the cost to node operators under normal transaction volume, i.e. when there is no black
 88 marble flooding. Total cost is the sum of $C_u(f, n)$ and $C_d(n, m)$:

$$C(f, n, m) = C_u(f, n) + C_d(n, m) \quad (4)$$

89 With budget b , Sam can afford to place $\frac{b}{f}$ bytes of transaction data in a block. Sam would create transactions
 90 with one input and two outputs. The formula for the number of bytes of a transaction like this in terms of the
 91 ring size n is $975 + 35n$. The 975 bytes is the size of the transaction except for the linear cost of the ring size, i.e.
 92 a (invalid) `lin/2out` transaction with ring size 0 would have 975 bytes composed of the input’s key image, other
 93 input data that does not scale up with ring size, the outputs’ bulletproofs+, the outputs’ public key, and `tx_extra`
 94 data. The 35 coefficient on n is the sum of the bytes of the “ s ” component of the CLSAG ring signature of each
 95 ring member (32 bytes) and 3 bytes of the key offset integer that is used to create the output indices of the ring
 96 members. The 3 bytes is an empirical average of the bytes used by each key offset integer. The number of outputs
 97 per byte that Sam produces is $2/(975 + 35n)$ because each of his transaction has two outputs. To calculate the
 98 number of outputs per block that Sam can afford with budget b when fee is f and nominal ring size is n , we compute
 99 the product of $\frac{b}{f}$ and $2/(975 + 35n)$, producing the formula for $s(b, f, n)$:

$$s(b, f, n) = \frac{2b}{f \cdot (975 + 35n)} \quad (5)$$

100 Let r be the number of real user outputs. When the number of outputs owned by Sam is $s(b, f, n)$, the long-term
 101 mean effective ring size³ is

$$n_e(b, f, n) = 1 + (n - 1) \cdot \frac{r}{r + s(b, f, n)} \quad (6)$$

²In April 2024, the median retail price of a 1TB SATA SSD on <https://ssd.userbenchmark.com/> was 114.50 USD. The exchange rate at the time was 120 USD per XMR.

³For a derivation of mean effective ring size, see Section 3 of Draft v0.2 of Rucknium (2024) “March 2024 Suspected Black Marble Flooding Against Monero: Privacy, User Experience, and Countermeasures” <https://github.com/Rucknium/misc-research/blob/main/Monero-Black-Marble-Flood/pdf/monero-black-marble-flood.pdf>

102 Alice wants to have a larger n_e when Sam is producing black marbles. n_e is the desired outcome in the cost-
 103 effectiveness analysis:

$$CE = \frac{C(f, n, m)}{n_e(b, f, n)} \quad (7)$$

104 Alice’s goal is to choose minimum fee per byte f and nominal ring size n to minimize CE when Sam spends
 105 his budget b producing black marbles and the node cost multiplier is some specified m . In game theory, a player’s
 106 *best response* in a two-player game is a strategy that gives the player the best payoff when the other player plays
 107 some specified strategy. Alice’s best response to Sam playing some b as a strategy is to set f and n to minimize
 108 CE . Alice does not know what value of b Sam intends to play, but reasonable values of b can be analyzed to guide
 109 reasonable choices of f and n . In game theory terms, Alice’s uncertainty about Sam’s b means that this is a game
 110 of imperfect information. Sam’s player “type” is the unknown b . Sam has some probability of being each type. In
 111 this document I will not explicitly declare some probability distribution of Sam’s type, but one could determine
 112 Alice’s single best response for the expected value of her cost effectiveness when Sam’s type has some probability
 113 distribution.

114 Define f_{\min} and f_{\max} as the minimum and maximum f that Alice is willing to set. Let n_{\min} and n_{\max} be the
 115 minimum and maximum n that Alice is willing to set. Assume that Alice wants to make sure that the effective ring
 116 size does not fall below some specified minimum acceptable limit \tilde{n}_e . Alice will try to minimize (7) except when
 117 the effective ring size would be below \tilde{n}_e at the minimum of (7). In that case, Alice will exclude the values of n and
 118 f that cause effective ring size to be below \tilde{n}_e , then choose n and f to minimize (7) from the set of n and f values
 119 that remain.

120 Alice’s best response correspondence given Sam’s choice of b and the node cost multiplier m is the solution to

$$\begin{aligned} & \arg \min_{f, n} \frac{C(f, n, m)}{n_e(b, f, n)} \\ & \text{subject to} \\ & f_{\min} \leq f, f \leq f_{\max} \\ & n_{\min} \leq n, n \leq n_{\max} \\ & \tilde{n}_e \leq n_e(b, f, n) \end{aligned} \quad (8)$$

121 The problem in (8) is a nonlinear minimization problem with nonlinear inequality constraints. Note that the
 122 constraint set is convex, but the objective function is neither globally convex nor globally concave.⁴ The necessary
 123 conditions for the solution could be found analytically by checking the Karush-Kuhn-Tucker conditions. I will solve
 124 it numerically with a grid search. The grid is formed by evaluating (7) many times at different values of f and n .
 125 The values of f are 40 equally-spaced values between f_{\min} and f_{\max} . The values of n are each integer between n_{\min}
 126 and n_{\max} .

127 We will start with a simple example. Assume that the adversary’s budget is 2.5 XMR per day. This is
 128 approximately the expenditure rate of the suspected black marble flooder in March 2024. We will evaluate cost-
 129 effectiveness at each combination of $f = \{10, 20, 40, 100, 200\}$ nanoneros per byte and $n = \{16, 30, 45, 60\}$ ring
 130 size.

131 Table 1 contains the cost effectiveness (CE) computations with other metrics like transaction size, total projected
 132 growth of the blockchain, and estimated transaction verification time. Note that the cost to send a 2in/2out
 133 transaction increases when ring size increases even if the fee per byte does not increase because users have to pay
 134 for larger total transaction size. The numerator of CE has been scaled to millineros. The lowest value in the CE

⁴The full proof of this statement is TODO. The first four constraints form a convex set because they are affine. The $\tilde{n}_e \leq n_e(b, f, n)$ constraint is more complicated. The Hessian matrix of the second-order partial derivatives of n_e with respect to f and n is negative definite as long as $n > 1$. That means that its superlevel set for some \tilde{n}_e is convex. (The $\tilde{n}_e \leq n_e(b, f, n)$ inequality defines the superlevel set.) The intersection of two convex sets is convex, so the constraint set of (8) is convex.

135 column is 0.48 when nominal ring size is 60 and minimum fee is 40 nanoneros per byte. Sam can achieve a 37.5
136 effective ring size with a 2.5 XMR/day budget when nominal ring size is 60 and minimum fee is 40 nanoneros per
137 byte. Estimation of transaction and block verification time is explained in Appendix A..

138 Figure 1 is a shaded contour plot of cost effectiveness when Sam has a budget of 50 XMR per day. Lighter
139 colors on the plot indicate lower CE values at the specified minimum fee and ring size values. The blue triangle
140 indicates the fee and ring size values that minimize the CE when the minimum acceptable effective ring size of 5
141 is disregarded. When we allow only fee and ring size values that produce effective ring size above the minimum
142 acceptable effective ring, the green circle indicates the fee and ring size values that minimize the CE. In this plot
143 the triangle and circle are at the same location because the minimum CE produces an effective ring size of 12.8,
144 above the minimum effective ring size of 5.

145 Table 2 shows the values of minimum fee and ring size that produce optimal cost effectiveness when Sam has
146 different budgets. The maximum budget, 500 XMR per day, exceeds Monero’s daily security budget provided by
147 tail emission. An adversary’s budget higher than 500 might imply that the adversary could directly 51 percent
148 attack the blockchain by renting CPU hashpower. It seems unnecessary to consider a black marble flooder’s budget
149 greater than 500 XMR per day because an adversary with a higher budget might be able to do more damage to
150 Monero than flooding the blockchain with black marble outputs.

Table 1: Cost effectiveness of minimum fee and ring size options when adversary budget is 2.5 XMR per day

Adversary XMR budget per day	Nominal ring size	Min fee (nanoneros per byte)	CE	Size of 2in/2out (bytes)	User's cost to send 2in/2out (USD cents)	Seconds to verify 2in/2out	Normal tx volume		Normal tx volume + black marble flooding						
							Block size (KB)	Seconds to verify txs in block	One year blockchain growth (GB)				Effective ring size	Block size (KB)	Seconds to verify txs in block
									Unpruned	Pruned	Unpruned	Pruned			
2.50	16	10	1.36	2,219	0.27	0.012	95	0.54	25	8	116	34	3.59	443	2.69
2.50	16	20	1.08	2,219	0.53	0.012	95	0.54	25	8	71	21	5.41	269	1.62
2.50	16	40	0.99	2,219	1.07	0.012	95	0.54	25	8	48	14	7.82	182	1.08
2.50	16	100	1.21	2,219	2.66	0.012	95	0.54	25	8	34	10	11.14	130	0.75
2.50	16	200	1.75	2,219	5.33	0.012	95	0.54	25	8	30	9	13.10	113	0.64
2.50	30	10	0.97	3,196	0.38	0.022	136	0.94	36	10	127	34	7.25	483	3.82
2.50	30	20	0.74	3,196	0.77	0.022	136	0.94	36	10	81	22	11.29	310	2.38
2.50	30	40	0.69	3,196	1.53	0.022	136	0.94	36	10	59	16	16.19	223	1.66
2.50	30	100	0.87	3,196	3.83	0.022	136	0.94	36	10	45	12	22.26	171	1.23
2.50	30	200	1.29	3,196	7.67	0.022	136	0.94	36	10	40	11	25.54	153	1.09
2.50	45	10	0.76	4,242	0.51	0.032	180	1.38	47	12	138	35	12.31	527	4.68
2.50	45	20	0.59	4,242	1.02	0.032	180	1.38	47	12	93	24	19.00	353	3.03
2.50	45	40	0.56	4,242	2.04	0.032	180	1.38	47	12	70	18	26.55	267	2.21
2.50	45	100	0.73	4,242	5.09	0.032	180	1.38	47	12	56	15	35.14	215	1.71
2.50	45	200	1.10	4,242	10.18	0.032	180	1.38	47	12	52	13	39.45	197	1.55
2.50	60	10	0.63	5,289	0.63	0.043	223	1.82	59	14	150	37	18.37	571	5.39
2.50	60	20	0.50	5,289	1.27	0.043	223	1.82	59	14	104	26	27.84	397	3.60
2.50	60	40	0.48	5,289	2.54	0.043	223	1.82	59	14	82	20	37.90	310	2.71
2.50	60	100	0.65	5,289	6.35	0.043	223	1.82	59	14	68	17	48.60	258	2.18
2.50	60	200	1.01	5,289	12.69	0.043	223	1.82	59	14	63	16	53.69	241	2.00

Row in green is the status quo. Row in orange is the best cost effectiveness.

Figure 1: Most cost-effective minimum fee and ring size when adversary budget is 50 XMR per day

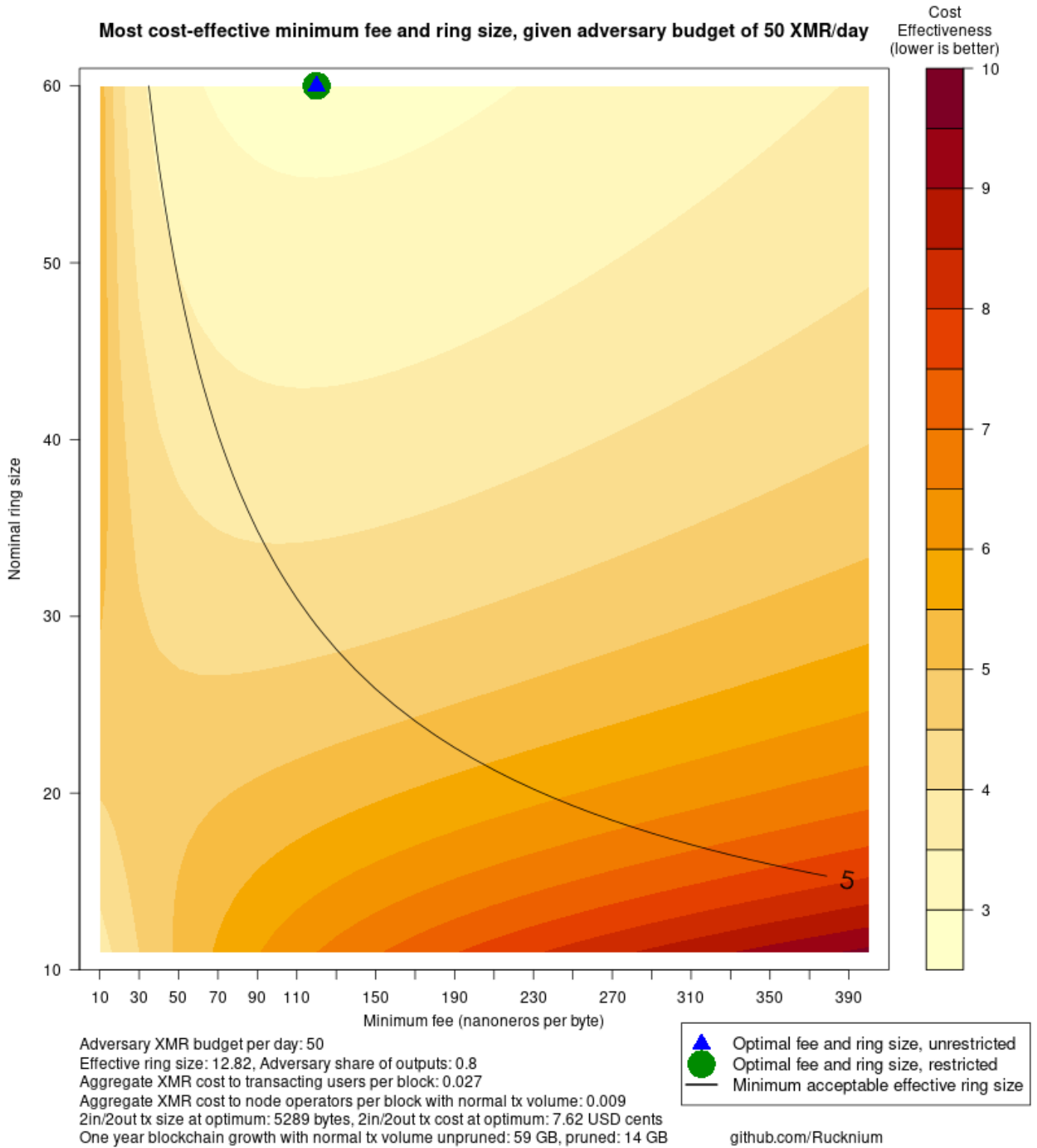


Table 2: Minimum fee and ring size at optimal cost effectiveness, adversary budget scenarios

Adversary XMR budget per day	Nominal ring size	Min fee (nanoneros per byte)	CE	Size of 2in/2out (bytes)	User's cost to send 2in/2out (USD cents)	Seconds to verify 2in/2out	Normal tx volume		Normal tx volume + black marble flooding				Effective ring size	Block size (KB)	Seconds to verify txs in block
							Block size (KB)	Seconds to verify txs in block	One year blockchain growth (GB)						
									Unpruned	Pruned	Unpruned	Pruned			
2.50	16	20	1.08	2,219	0.53	0.012	95	0.54	25	8	71	21	5.41	269	1.62
0.25	60	10	0.24	5,289	0.63	0.043	223	1.82	59	14	68	17	48.60	258	2.18
0.50	60	10	0.29	5,289	0.63	0.043	223	1.82	59	14	77	19	40.89	293	2.53
1.25	60	20	0.37	5,289	1.27	0.043	223	1.82	59	14	82	20	37.90	310	2.71
2.50	60	30	0.48	5,289	1.90	0.043	223	1.82	59	14	89	22	33.80	339	3.01
5.00	60	40	0.66	5,289	2.54	0.043	223	1.82	59	14	104	26	27.84	397	3.60
12.50	60	70	1.10	5,289	4.44	0.043	223	1.82	59	14	124	30	22.76	471	4.37
25.00	60	90	1.73	5,289	5.71	0.043	223	1.82	59	14	160	39	17.11	609	5.78
50.00	60	120	2.83	5,289	7.62	0.043	223	1.82	59	14	211	52	12.82	802	7.77
125.00	60	140	5.68	5,289	8.89	0.043	223	1.82	59	14	385	94	7.17	1,464	14.56
250.00	60	180	9.69	5,289	11.42	0.043	223	1.82	59	14	566	138	5.12	2,152	21.64
500.00	60	350	17.47	5,289	22.21	0.043	223	1.82	59	14	580	142	5.02	2,208	22.20

Row in green is the status quo

3 Discussion

What have we learned? According to this analysis, raising the ring size is a more cost-effective strategy against a black marble attack than raising fees. A combination of a large increase in ring size and a modest increase in fee seems to provide a good, cost-effective defense.

Consider an adversary with a daily budget of 12.5 XMR, five times higher than the daily expenditure of the suspected March 2024 black marble flooder. Table 2 says the most cost-effective combination of defense parameters are ring size 60 and minimum 70 nanonero per byte fee. Effective ring size would be 22.8 if Sam spent his entire budget every day. The 2in/2out reference transaction with ring size 60 would be about 140% larger than the transaction with current ring size 16. The user's cost to send this transaction would be about 4.4 USD cents. The total time to verify all transactions in a block of normal transaction volume would increase from 0.5 seconds to 1.8 seconds. An unpruned node would grow 59 GB in a year instead of 25 GB. Pruned nodes would grow 14 GB instead of 8 GB.

Put these storage requirements into perspective. Recall that we use base-10 (SI) units to measure bytes in this document. As of May 2024, a unpruned Monero blockchain is 206 GB. A pruned Monero node takes 79 GB of storage space. The 2023 Ultra 4K edition of Call of Duty requires 229 GB of storage.⁵ An unpruned BTC node requires 650 GB of storage and grows about 89 GB per year.⁶ Therefore, with ring size 60 the Monero blockchain would grow slower than the BTC blockchain, crossing the Call of Duty storage threshold within a year.

Encouraging node operators to prune their nodes and implementing a coinbase consolidation transaction type could reduce the impact of increasing the minimum fee and ring size. Pruning could be encouraged by setting pruning as the default in more Monero software interfaces, such as the Monero GUI wallet Pull Request #4320, and public information campaigns.⁷ A coinbase consolidation type would reduce the transaction size for small coinbase outputs.⁸

4 Summary: Downsides and benefits of options

1. Increase the minimum relay fee per byte

(a) Downsides:

- i. Users may make fewer transactions. That would reduce Monero's total anonymity set because the rate of creation of new outputs would fall.
- ii. Users could move to another means of payment.
- iii. Monero might lose its reputation as a low-cost means of payment.
- iv. Large changes in Monero's fiat exchange rate could make the purchasing power of the minimum fee much higher or lower than anticipated.

(b) Benefits:

- i. Miners would earn more from fees. This would increase Monero's resistance to 51 percent attack because its mining security budget would increase a little.
- ii. Higher fees would increase the cost of all spam regardless of motivation. (Increasing the ring size only negatively affects spammers that want to reduce the effective ring size.)

2. Increase the ring size

⁵<https://web.archive.org/web/20231214215231/https://www.callofduty.com/blog/2023/10/call-of-duty-modern-warfare-III-specs-preloading-pc-trailer>

⁶<https://bitcoin.stackexchange.com/a/116350> and <https://transactionfee.info/charts/block-size/>

⁷<https://github.com/monero-project/monero-gui/pull/4320>

⁸<https://github.com/monero-project/research-lab/issues/108>

192 (a) Downsides:

- 193 i. Greater storage requirements for operating a Monero node could cause some node operators to stop
194 running their nodes. This would make the Monero network less decentralized.
- 195 ii. Some Monero wallet users may stop running local nodes and switch to remote nodes. This would
196 increase the load on public remote nodes and potentially expose the wallet users to some privacy
197 risks from malicious remote nodes.⁹
- 198 iii. Verification time per transaction would increase. During normal operation, the Monero node would
199 use more CPU resources. During initial blockchain download, total sync time would be greater.
200 Syncing a Monero node on a HDD, which is already very difficult, might become completely nonviable
201 because of the necessary random reads for ring signature verification.
- 202 iv. At extremes, long verification times can threaten network stability. In 2023 Pirate Chain suffered
203 a transaction spam attack that caused chain splits because of long transaction verification times.¹⁰
204 Monero uses the Fluffy Blocks protocol to verify transactions as they arrive in the txpool instead of
205 bottlenecking verification at the time new blocks are mined. It is unclear if Pirate Chain, a code fork
206 of Zcash, uses a compact block protocol.¹¹ As long as the time to verify each block's transactions
207 does not become a large fraction of mean time between blocks (120 seconds), this is probably not
208 a threatening issue, *in theory*. In practice, the Monero node performs many more actions than just
209 verifying the cryptography of transactions. There may be hidden bottlenecks. Recently, spikes of
210 transactions with large numbers of inputs have seemed to cause excess RAM usage of nodes, shutting
211 down nodes in some cases.¹²

212 (b) Benefits:

- 213 i. Increasing the ring size increases the anonymity set of all transaction inputs. Other statistical attacks
214 unrelated to black marble flooding like EAE attacks and timing analysis would become more difficult.

215 (c) Neutral:

- 216 i. Increasing the ring size has very little effect on wallet sync times. The bandwidth costs for syncing
217 transactions in mined blocks are only about three bytes per ring member for the ring offset data.
218 No additional computation is required. However, ring signature data is sent from nodes to wallets
219 when transactions are still in the txpool.¹³

220 3. Encourage blockchain pruning

221 (a) Downsides:

- 222 i. New unpruned nodes may have to connect to more nodes to create an unpruned copy of the
223 blockchain.
- 224 ii. All pruned nodes keep one-eighth of the transaction data that is designated “prunable”. If all nodes
225 on the network are pruned, there is an extremely small chance that one of the eight pruning slices
226 will not exist on the whole network. That would mean that not all signature data on the blockchain
227 could be verified. When blockchain pruning is enabled, a Monero node randomly chooses one of eight
228 possible pruning seeds independently of the pruning seeds that other nodes have chosen. By chance,
229 the network could be missing one of the eight slices of the pruneable part of the blockchain because

⁹See <https://docs.featherwallet.org/guides/nodes>

¹⁰https://web.archive.org/web/20230803171107/https://old.reddit.com/user/SignificantRoof5656/comments/15h9reh/pirate_chains_045_spam_attack_2_months_later/

¹¹See <https://github.com/zcash/zips/issues/360>

¹²<https://github.com/monero-project/monero/issues/9317>

¹³Thanks to jeffro256 for explaining this.

230 the choice of pruning seed is not coordinated between nodes. This chance is extremely small. If the
231 network only has pruned nodes and the total number of nodes on the network is 681, the probability
232 of missing one of the eight pruning slices is less than 2^{-128} , which is the probability of guessing a
233 specific 12-word BIP39 bitcoin seed phrase with a single guess. See Appendix B for how to compute
234 this probability. This probability does not consider the challenge of new nodes finding their pruning
235 slices by connecting to multiple nodes throughout the network.

236 (b) Benefits:

- 237 i. Pruned nodes consume much less storage space.

238 (c) Neutral:

- 239 i. “There are no privacy or security downsides when using a pruned node.”¹⁴

240 4. Implement “Coinbase Consolidation Tx Type”¹⁵

241 (a) Downsides:

- 242 i. koe, the original proposer of this protocol modification, said, “After thinking more, I am not sure this
243 proposal is the right direction. Enote consolidation being statistically significant, and consolidating
244 enotes with small amounts being expensive, is a general problem not specific to coinbase enotes.
245 Implementing a specific solution for coinbase enotes amounts to elevating the circumstances of miners
246 to first-class status in the protocol, without solving the more general problem. If another major
247 project on the scale of p2pool becomes active in Monero and would benefit from specific protocol
248 changes (not trivial benefits - privacy and scaling benefits even), should we hard fork to accommodate
249 them? To support protocol longevity by reducing hardforks (and not setting precedents that would
250 justify a relatively higher rate of future hardforks), it seems better to aim for general solutions to
251 problems. In this case, one general solution to the privacy impacts of consolidation would be a global
252 membership proof. The cost of consolidations might be addressed by using aggregate membership
253 proofs that scale sub-linearly with the number of memberships being proven (i.e. number of tx
254 inputs).”¹⁶

- 255 ii. There is a small privacy impact to some miners. Most centralized mining pools already publish
256 the blocks that they mine, so the ownership of mining pools’ coinbases is usually publicly known
257 already.¹⁷ P2Pool payout addresses are public on the P2Pool side chain, allowing good guesses
258 about which transactions are consolidating coinbases to specific mining addresses.¹⁸ The P2Pool
259 README recommends miners to use a separate mining wallet.¹⁹ Therefore, a coinbase consolidation
260 transaction type would not have a large negative impact on the privacy of most miners because the
261 on-chain privacy for miners is low to begin with. The privacy of solo miners could be negatively
262 impacted, however. With the new transaction consolidation type, those miners could send coins to
263 themselves once to create outputs that would enter the non-coinbase anonymity set.

264 (b) Benefits:

- 265 i. If the coinbase consolidation transaction type is implemented at the same time as much larger rings,
266 coinbase consolidations would not take up so much storage. In the 60 ring member scenario, annual

¹⁴<https://web.getmonero.org/resources/moneropedia/pruning.html>

¹⁵<https://github.com/monero-project/research-lab/issues/108>

¹⁶<https://github.com/monero-project/research-lab/issues/108#issuecomment-1379288635>

¹⁷Wijaya, D. A., Liu, J. K., Steinfeld, R., & Liu, D. (2021) “Transparency or anonymity leak: Monero mining pools data publication”. Paper presented at Information Security and Privacy - 26th Australasian Conference, ACISP 2021, Virtual Event, December 1-3, 2021, Proceedings.

¹⁸<https://p2pool.observer/sweeps>

¹⁹<https://github.com/SChernykh/p2pool?tab=readme-ov-file#general-considerations>

267 blockchain growth would be 2.7 GB less if all coinbase outputs are spent by inputs with ring size
268 one.

- 269 ii. If the ring size and/or fee per byte increases a lot, P2Pool mining may become uncompetitive
270 compared to centralized pool mining, especially for the P2Pool mini chain. Consider the 10th
271 percentile of multi-output coinbase outputs during February 2024: 0.000272 XMR. (10% of the
272 likely P2Pool outputs are below this amount.) With the status quo ring size and minimum fee per
273 byte, consolidating this P2Pool payout by adding an input to a transaction costs the miner about
274 5 percent of the value of that output. With the ring size 60 and 70 nanoneros per byte scenario
275 considered above, about 57 percent of the value of that output would be consumed by the cost to
276 spent the output in a transaction's output. But if coinbase outputs only have to have ring size
277 1, then even paying 60 nanoneros per byte would cost the miner only 4.2 percent of the output's
278 value when you spent it in a 1-ring-member input. (The cost quoted here do not include the bytes
279 contributed by outputs or other transaction data.)
- 280 iii. Coinbase outputs can behave like black marbles in the rings of transactions that do not spend
281 coinbase outputs. See the "Avoiding selecting coinbase outputs as decoys" Monero Research Lab
282 issue.²⁰ Implementing a coinbase consolidation transaction type would prevent coinbase outputs
283 from being included in the rings of transactions that do not spend coinbase outputs. This would
284 improve the privacy of those transactions.

²⁰<https://github.com/monero-project/research-lab/issues/109>

285 A Appendix: Transaction verification time estimates

286 The verification time estimates are based on performance tests developed by koe. I modified the test parameters to
287 produce estimates of a large set of ring sizes, input counts, and output counts in [https://github.com/Rucknium/
288 monero-tx-performance](https://github.com/Rucknium/monero-tx-performance). koe provides interpretation of the performance tests in Monero Research Lab issue
289 #91.²¹ I used the same machine as koe for the tests. The verification performance tests do not include the time
290 to read data from storage media. The 2in/2out reference transaction and the assumed 1in/2out black marble
291 transaction type could be tested directly, but there were too many permutations of the real transaction in/out
292 types in the February-March 2024 sample to test those directly. Estimates of the real transaction verification type
293 were necessary to estimate the verification time for an average real block. All tests were in batches of one because
294 setting the batching parameter did not seem to affect the verification time of inputs (it did affect verification time
295 of outputs, but the research question is about varying different ring sizes of inputs, not outputs).

296 A linear regression model was fit by Ordinary Least Squares (OLS) to interpolate the estimated verification
297 times for the set of real transactions at different ring sizes. The performance test developed by koe were originally
298 designed to only compute ring sizes that are powers of two. Therefore, ring size performance was tested for ring size
299 1, 2, 4, 8, 16, 32, and 64. The number of outputs tested was every integer between 2 and 16 because these are the
300 allowed number of transaction outputs by blockchain consensus rules. The maximum number of inputs in a single
301 transaction that a standard Monero wallet can produce is 150. The number of inputs I used for the performance
302 estimates was:

303 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150\}$

304 Taking all permutations of these sets gives $7 \cdot 15 \cdot 24 = 2520$ permutations. Tests with these permutations produce
305 the dataset used in the OLS regression. We can guess a good functional form for the regression equation based
306 on knowledge of the time complexity of the algorithms used to cryptographically verify transaction components. I
307 included ring size and inputs, the base 2 log of each, and their interaction terms. Dummy variables of the ceiling of
308 base-2 log of the number of outputs was included in the regression equation since bulletproofs verification times is
309 a function of the integer ceiling of the power of two of the number of outputs in the transaction. The full regression
310 equation is below.

$$\begin{aligned} \text{time} = & \beta_0 + \beta_1 \text{ring_size} + \beta_2 \text{inputs} + \beta_3 \log_2(\text{ring_size}) + \beta_4 \log_2(\text{inputs}) + \beta_5 \mathbb{1}\{[\log_2(\text{outputs})] = 2\} \\ & + \beta_6 \mathbb{1}\{[\log_2(\text{outputs})] = 3\} + \beta_7 \mathbb{1}\{[\log_2(\text{outputs})] = 4\} \\ & + \beta_8 \text{ring_size} \times \text{inputs} + \beta_9 \log_2(\text{ring_size}) \times \log_2(\text{inputs}) + \epsilon \end{aligned} \quad (9)$$

311 $\lceil x \rceil$ means the integer ceiling of x . $\mathbb{1}\{x\}$ is an indicator function. Its value is 1 when the statement in braces is
312 true and is 0 otherwise. The results of the regression are in Table 3.. The adjusted R^2 is extremely high (0.9998),
313 indicating that the model fits the data well. However, a model may have a high R^2 when the scale of the different
314 observations is vastly different, which is the case here.

315 Given the estimated parameters from (9), predicted values of the verification time for all types of transactions
316 and all ring sizes can be computed for the February-March 2024 sample by plugging the number of inputs, outputs,
317 and ring size into the regression equation with the $\hat{\beta}$ estimated parameters.

²¹<https://github.com/monero-project/research-lab/issues/91>

Table 3: CLSAG transaction verification time OLS regression. Units are milliseconds.

	(1)
(Intercept)	-0.555 (0.766)
ring size	0.218 *** (0.015)
inputs	-0.141 *** (0.006)
$\log_2(\text{ring size})$	0.670 ** (0.234)
$\log_2(\text{inputs})$	3.956 *** (0.181)
$[\log_2(\text{outputs})] = 2$	3.462 *** (0.558)
$[\log_2(\text{outputs})] = 3$	9.906 *** (0.510)
$[\log_2(\text{outputs})] = 4$	21.783 *** (0.484)
ring size \times inputs	0.254 *** (0.000)
$\log_2(\text{ring size}) \times \log_2(\text{inputs})$	-1.362 *** (0.044)
N	2520
Adjusted R-squared	0.9998

Standard errors in parentheses. *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

318 B Probability of recovering complete blockchain data from a network 319 with only pruned nodes

320 The problem of collecting all eight of the Monero’s pruning slices is a type of coupon collector’s problem. Holst
321 (1986) provides the formula to find the probability that you need n pruned nodes on the network to be able to recover
322 the intact blockchain from the eight unique slices.²² Holst says, “In this paper we will consider problems connected
323 with drawing with replacement from an urn with r balls of different colours.....The inverse of the occupancy problem
324 is sometimes called the coupon collector’s problem. It reads: how many draws are necessary for obtaining k different
325 balls?” Holst gives the general problem when r is not necessarily equal to k . In the pruned node problem, we only
326 need one copy of each unique slice, so $r = k$ in our case. Holst says that the probability of needing exactly n draws
327 for obtaining k different balls when the urn has r balls of different colors is

$$P(W_{k:r} = n) = r_{(k)}S(n-1, k-1)/r^n \quad (10)$$

328 Holst defines $r_{(k)} \equiv r(r-1)\dots(r-k+1)$. When $r = k$, this is the factorial $r_{(k)} = r!$. $S(n, k)$ is a Stirling
329 number of the second kind:

$$S(n, k) = \sum_{j=0}^k (-1)^j \binom{k}{j} (k-j)^n$$

330 The (10) equation is the probability that you need exactly n nodes on the network to have all eight distinct
331 slices. We want to know the probability that you need more than n nodes to have all the slices. This probability is
332 $1 - \sum_{i=8}^n P(W_{k:r} = i)$.

333 To avoid limitations of floating point computer arithmetic, when computing these values it is recommended to
334 use a software library that uses arbitrary-precision numbers such as the GNU Multiple Precision Arithmetic Library.

335 Figure 2 plots the probability that a Monero network would not contain all 8 pruned node slices. When there are
336 100 nodes on the network, the probability is about 0.001 percent. When the number of nodes is 681, the probability
337 of not having all 8 pruned node slices is less than 2^{-128} , which is the probability of guessing a specific 12-word
338 BIP39 bitcoin seed phrase with a single guess.²³ These probabilities correspond to a network in a single point in
339 time. If we consider that a network will have many “draws” in its lifetime, then the probability of missing one of
340 the eight slices during any point in its lifetime is higher. If the whole set of n nodes re-draws its random pruning
341 seed q times, the probability of never missing one of the eight slices is $(1 - P(\text{missing at least one slice}))^q$ because
342 the draws are independent.

²²Holst, L. (1986). “On Birthday, Collectors’, Occupancy and Other Classical Urn Problems.” *International Statistical Review* /
Revue Internationale de Statistique, 54(1), 15–27. <https://doi.org/10.2307/1403255>

²³<https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki>

Figure 2: Probability of not having all 8 distinct pruned slices on the Monero network

