Iceman: Freezing Blockchain Assets such as Bitcoin

Dr. Virgil Griffith
i@virgil.gr

Abstract

The traditional wisdom is that there's no effective way to freeze cryptocurrency assets such as Bitcoin. To the consternation of some, this is actually not true. We introduce ICEMAN, a novel and surprisingly inexpensive system for operating a blacklist of addresses that are then unable to make transactions. Like Bitcoin, ICEMAN works at the mining level yoking technical innovation and economic incentives. Our system applies to any existing blockchain-based asset and has three parameters s, ϵ, and m which can be tuned for economic efficiency. The appendix glossary contains the various mathematical notation and unsolved mathematics.

1 Background: Bitcoin Mining

Before publishing a new Bitcoin [1] block, a miner listens to the Internet for Bitcoin transactions. After listening, the miner ensembles a list of valid transactions (e.g., no double-spends) and then adjusts two arbitrary digits in the header (the block and coinbase headers) until the hash of the entire ensemble (block) is below a known threshold value.[1]

A miner currently earns an average of Ƀ25.34 (∼$10,136) per published block. The vast majority of this comes from the Ƀ25.00 block reward, with the rest from the summed transaction fees. As-is, miners refrain from including any transaction that doesn't pay some minimal transaction fee; so each miner already exercises some discretion in choosing which transactions to publish. In ICEMAN, we aim for miners to additionally refrain from publishing transactions originating from blacklisted addresses.

2 From Punitive Forking to Feather Forking

Lets start with the simplest deterrent—say all miners in the United States are legally persuaded to no longer publish transactions originating from some address X—say X belongs to the Islamic State.[2] Unfortunately, address X would still be able to transact whenever an indifferent miner publishes a block.

In response, you could try to punish the indifferent miners with punitive forking. You announce a "Federation of Legally-Persuaded Miners" who refuse to build atop any future block containing a banned transaction. This is an extreme strategy, and one that fails if the federation has $< 51\%$ of the total hashrate. If the federation has $< 51\%$, every time an indifferent miner publishes a banned transaction, the ledger bifurcates and a race ensues between the Federation and the Indifferents for who will make the longer chain. Even if the Federation gets lucky and wins the first few races, eventually the Indifferents will win a race and from that point forward the federation chain will fall behind to never catch up, and the federation chain is subsequently ignored. Given that US miners constitute less than half of the Bitcoin hashrate, punitive forking is a non-starter.

[1] As of January 2016, this threshold is `00000000000000000928f928f928f980000000000000000000000000000000000`.
[2] Such as `16itdFoTN5HSKqJG6Jxh9znbT7n2oEViGy`. [2]

But there's a more clever way to do it. Instead of forking forever, we will *give up after a while*. [3,4] Specifically, if we have not gotten lucky and won the race after m blocks, we will accept defeat and rejoin the Indifferents in mining atop the longest chain.

3 Enlisting Miners to Join Our Fork

Say a block containing a transaction originating from address X has just been published. The race is on between the Federation mining atop Fork A without the banned transaction and the Indifferents mining atop fork B whose head contains the banned transaction. The Indifferents constitute the vast majority of hashrate, but our hypothetical federation still has a minuscule probability ω of winning the race before giving up.

3.1 Enlisting All Rational Miners... for Free?

By definition, Rational miners are neither for nor against the blacklist—they simply want to maximize their income. The Nash equilibrium [5] is the bread-and-butter for predicting Rational agents, so lets examine it closely. Each Rational miner is now faced with a decision—which fork should it mine on? To decide this, each Rational miner considers the payout matrix in Figure 1a. The expected return for joining each fork is:

$$\begin{aligned} \mathbb{E}[\text{build on fork } A] &= (\omega)r + (1-\omega)0 = \omega r \\ \mathbb{E}[\text{build on fork } B] &= (\omega)0 + (1-\omega)r = (1-\omega)r \end{aligned} \quad . \tag{1}$$

Subtracting these two and factoring r, a Rational miner will build upon fork A if and only if our probability of winning the race $\omega > 1/2$. As that's never going to happen, we must try something else—enter Nash-shifting.

	A wins	B wins
	ω	$1-\omega$
build on fork A	r	0
build on fork B	0	r

(a) Before Nash-shifting

	A wins	B wins
	ω	$1-\omega$
build on fork A	r	s
build on fork B	0	r

(b) After Nash-shifting

Figure 1: Payout matrices for a miner building atop fork A versus fork B.

In Nash-shifting, a government or law enforcement organization credibly offers some payment s to any Bitcoin miner who publishes atop fork A, yet fork B still wins. This results in the payout matrix in Figure 1b. After Nash-shifting, the expected return for joining each fork is:

$$\begin{aligned} \mathbb{E}[\text{build on fork } A] &= (\omega)r + (1-\omega)s = \omega r + (1-\omega)s \\ \mathbb{E}[\text{build on fork } B] &= (\omega)0 + (1-\omega)r = (1-\omega)r \end{aligned} \quad . \tag{2}$$

Subtracting the two, we see that, after Nash-shifting, a Rational miner will join fork A if and only if,

$$s > r\frac{1-2\omega}{1-\omega} \; . \tag{3}$$

Note that if $s > r$, eq. (3) is satisfied for all values ω—**even when $\boldsymbol{\omega = 0}$**. Therefore to enlist all Rational miners into joining fork A, we actually don't need any legally-persuaded miners at all! The government can have nothing, and still win simply by offering a reward $s > r$. This is important to fully understand—if *every miner* was Rational, **a credible offer to reimburse $\mathbf{s > r}$ results in a 100% effective blacklist and the government would never have to pay for it** because fork A would always win. [6]

Presumably a sizeable fraction of major miners are Rational, but there will always exist at least one miner that is neither Legally-Persuaded nor Rational. Ergo, there's always a nonzero chance that our fork will lose and the government have to pay s. On the bright-side, actually paying s greatly bolsters the credibility of the government's offer, and will perhaps convince more miners to simply be Rational.

3.2 Enlisting All Superrational Miners... for Cheap.

Whereas Rational agents move to a Nash equilibrium, *Superrational* [7] agents move to a Hoftstadter equilibrium.[3] A Superrational agent thinks to itself, "The other agents are just as smart as me, so in a symmetric game we're all going to do the same thing." So whereas a Rational agent maximizes its payout given the independent actions of all other agents, a Superrational agent maximizes its payout under the constraint that *all agents will follow the same strategy.* This has some unusual consequences—for example, in the Prisoner's Dilemma a Rational agent always defects, but a Superrational agent always cooperates.

Faced with our fork, each Superrational miner is forced to decide between mining on fork A or fork B. Being Superrational, it only considers the relative payouts of (build on fork A, A wins) versus (build on fork B, B wins). Per Figure 1b, these choices have equal payout r. Faced with equal pay, the Superrational miner randomly chooses to mine atop A or B with probability $1/2$.

At first glance, the Superrationals choosing the "half A, half B" strategy might be acceptable because, on average, the Superrational miners "cancel out". For example—say there are 101 miners where one miner is Rational and the rest are Superrational. In expectation, fork A will have 51 miners behind it but fork B will only have 50 miners, making fork A the winner. This result holds even when miners have varying hashrates.

That said, we prefer to minimize variability. So what would it cost to enlist all Superrational miners to mine atop fork A? Simply offer an iota of payment, say $\epsilon = ฿0.0125$ ($5), to the miner who publishes atop fork A and fork A wins. This results in the payout matrix in Figure 2. Although this enlists all Superrational miners, it has the downside that the government now pays between $[\epsilon, m\epsilon]$ per successful fork.

	A wins	B wins
	ω	$1-\omega$
build on fork A	$r+\epsilon$	s
build on fork B	0	r

Figure 2: Payout matrix after Nash-shifting and Hofstadter-shifting.

4 The Fee to Begin the Race

In Section 3 we showed that, starting from any (even zero) Legally-Persuaded hashrate α_{LP}, we can enlist the Rational as well as Superrational hashrates α_R and α_{SR}, to our fork. This yields a total blacklist-enforcing hashrate of $\alpha = \alpha_{LP} + \alpha_R + \alpha_{SR}$. In practice, how big can we expect α to be? Apriori, we don't know, but it's certainly a force to be reckoned with—even $\alpha > 1/2$ is quite plausible! Additionally, the enforcing hashrate α will slowly become common knowledge as summing the hashrates of the miners empirically observed building atop fork A yields a lowerbound that converges to α.

With our blacklist-enforcing hashrate α in hand, we want to know the probability of winning the fork-race within m blocks. Analytically, a feather-fork is an instance of a Binomial Random Walk with bias α over m steps. After some long-winded algebra, this results in ω, the probability of winning the race, as,

$$\omega(m,\alpha) = \sum_{k=1}^{\lfloor m \rfloor} \mathcal{C}_k \, \alpha^{k+1}(1-\alpha)^{k-1} \,, \tag{4}$$

where $\mathcal{C}_k$ is the kth Catalan [9] number.[4] We evaluate eq. (4) plugging in specific values for m and α resulting in Table 1. For notational brevity we will often write $\omega(m,\alpha)$ as simply ω.

For conservatism and simplicity, we choose $m = 1$. Plugging $m = 1$ into eq. (4) yields the probability of winning $\omega = \alpha^2$; this is because we must find *two consecutive blocks*, and α^2 is the chance we get lucky twice

[3] Some literature refers to Hofstadter equilibriums as "Schelling Points". [8]

[4] The Catalan numbers are defined by $\mathcal{C}_k \equiv \frac{1}{k+1}\binom{2k}{k}$.

		α					
		$1/4$	$1/3$	$1/2$	$2/3$	$3/4$	$5/6$
m	1	6.3%	11.1%	25.0%	44.4%	56.3%	69.4%
	2	8.6%	16.0%	37.5%	64.2%	77.3%	88.7%
	3	9.7%	18.8%	45.3%	75.2%	87.2%	95.4%
	4	10.3%	20.5%	50.8%	82.0%	92.4%	98.0%

Table 1: Probability of winning the fork, $\omega(m, \alpha)$, plugging in various values for m and α.

in a row. Unfortunately, the probabilities along the top row of Figure 1 do not immediately inspire confidence. For example, if we have $\alpha = 1/2$ of the hashrate, once X's transaction is published, there's only a $.5^2 = 25\%$ chance of stopping it. But as even Indifferent miners can lookup[5] our parameter m as well as estimate our hashrate α, even low probabilities pack economic punch.

4.1 The Financial Cost of Publishing a Banned Transaction

Even if a miner is unbowed by our incentives s and ϵ, the unbowed miner knows that if he publishes a block containing a banned transaction, there's probability ω that his block will be orphaned and he will earn *nothing*. Ergo, if he publishes banned transactions willy-nilly, his income will drop precipitously and he will be slowly ground into bankruptcy—where incentive s is the carrot, loss of income due to orphaned blocks is the stick. So how much would address X have to offer for an unbowed miner to break even and avoid eventual bankruptcy? We turn to the payout matrix in Figure 3.

	A wins	B wins
	ω	$1-\omega$
ignore X's transaction	r	r
publish X's transaction	0	$r + r_X$

Figure 3: Payout matrix for publishing X's transaction. r_X is the transaction fee offered by address X.

$$\begin{aligned} \mathbb{E}[\text{ignore } X\text{'s transaction}] &= (\omega)r + (1-\omega)r &&= r \\ \mathbb{E}[\text{publish } X\text{'s transaction}] &= (\omega)0 + (1-\omega)(r + r_X) &&= r + r_X - \omega(r + r_X) \end{aligned} \tag{5}$$

Subtracting these two, a miner will break even if,

$$r_X > r\frac{\omega}{1-\omega}\ . \tag{6}$$

The value of r varies from block to block, but the average is Ƀ25.34 = \$10,136. In Table 2 we list the minimum transaction fee for a miner to break even.

4.2 Would a Rational Miner ever Publish a Banned Transaction?

There does exist a transaction fee r_X that would be sufficient even for a Rational miner to publish a banned transaction and then build atop fork B hoping the transaction would stick (for example, $r_X = \$1,000,000$). However, as Rational miners are swayed by our incentives s and ϵ, the minimum r_X for a Rational miner will be higher than for an Unbowed miner. We believe it's reasonable to assume there will always be at least one ideologically motivated, Unbowed miner and therefore suggest using the estimates in Table 2.

[5]For their convenience, we could even make a dynamic status page listing these various values.

ω	minimum fee	
5%	Ƀ 1.33	$ 532
10%	Ƀ 2.82	$ 1,126
25%	Ƀ 8.45	$ 3,379
50%	Ƀ25.34	$10,136
60%	Ƀ38.01	$15,204
70%	Ƀ59.13	$23,651

Table 2: Plugging-in the average value of r, $r = $ Ƀ25.34 yields the minimum transaction fee, r_X, address X has to pay for a miner to break even when publishing its transaction.

4.3 Would a Superrational Miner ever Publish a Banned Transaction?

A Superrational miner will think to itself, "I should publish X's transaction if only if it's Superrational to build upon the block containing X's transaction. As $\epsilon > 0$, it's not Superrational to build atop the block; therefore I will ignore X's transaction no matter the offered transaction fee r_X." This is convenient as a single incentive ϵ both enlists the Superrationals to our fork as well as precludes any Superrational from publishing a banned transaction.

5 Suggested Implementation

5.1 The Three Parameters s, ϵ, and m

I suggest launching with very conservative parameters: high s, high ϵ, and low m. As miners acclimate to the idea of an incentivized blacklist, we can start experimenting with lowering s, lowering ϵ, and increasing m.

For miners to behave Rationally, they must understand the calculations in eqs. (2) and (3). To make this calculation easy, at launch I suggest simply setting $s = r + 1$. As of January 2016, the average $r = $ Ƀ25.34 ($10,136), making $s = $ Ƀ26.34 ($10,536). We can later experiment with lowering s as long as $s > r\frac{1-2\omega}{1-\omega}$.

For Superrational miners, technically any payment $\epsilon > 0$ is sufficient to enlist their support, however, there is a human psychology angle in play akin to the Ultimatum Game. [10] Put simply, it's offensive to be offered too far below a "fair compensation", and people return the offense by rejecting the offer. At launch I suggest a relatively high $\epsilon = \$5$. We can later experiment with lowering ϵ as long as $\epsilon > 0$.

Per eq. (4), increasing m increases ω. Unfortunately, increasing m also permits longer forks which destabilize the Bitcoin ledger and thus more likely to be opposed by the community. At launch I suggest the maximally conservative $m = 1$. We can later experiment with increasing m.

5.2 Maintaining the Blacklist

The most natural way to do the bookkeeping is publishing the blacklist on the Bitcoin ledger itself. This provides both convenience for the miners as well as common knowledge for which addresses are blacklisted, and when each address was added. I suggest creating a "vanity" Bitcoin address[6] whose sole action is adding entries to the blacklist. When our branded address adds n new entries to the blacklist, it creates a single transaction with n outputs of value 0 Satoshi and uses the array of output scripts in Table 3. The `OP_RETURN` instruction signifies that everything afterward (up to 80 characters) is free-form data. `OP_RETURN` is frequently used for colored-coin transactions [11, 12] and is considered best-practice for storing arbitrary data in the Bitcoin ledger. [13] This simple design can be augmented with additional features such as entries that expire after so many days or a mechanism for removing entries.

[6]For example, I generated the vanity address: `1CEMANsqTrKSyvF3ftdKaPG6nWwZDeChFZ` which has private key: `5KFxZ4M71g5ubWHDiptRZ7Ud9P3ETYcBtkdZxxRbrWSEVojsNUj`.

PubKeyScript_1:	OP_RETURN <address_1> OP_CODESEPARATOR <URL_1 with additional info>
PubKeyScript_2:	OP_RETURN <address_2> OP_CODESEPARATOR <URL_2 with additional info>
⋮	⋮ ⋮
PubKeyScript_n:	OP_RETURN <address_n> OP_CODESEPARATOR <URL_n with additional info>

Table 3: Array of output scripts for adding n new addresses to the blacklist.

5.3 Compensating a Defeated Rational Miner

It's inevitable that, eventually, the government will have to pay the reward s to a miner (for $n = 2$, this will only ever be a single miner). To keep track of the miner who published atop fork A but whose efforts were ultimately superseded by those behind fork B, we will have to maintain a well-connected full Bitcoin node. This node needn't mine, but it must keep real-time track of all published blocks.[7] This node constitutes our bird's eye view onto the fork racetrack. If our node witnesses fork A's failure, it will reimburse any defeated miner in Bitcoin. Reimbursing via Bitcoin publicly demonstrates we are both capable and willing to follow-through on our reimbursements.

5.4 Social Issues

The Bitcoin community is suspicious of government regulation. The US government in particular is often seen as being too dictatorial within manifestly global affairs. To side-step this perception, I suggest tapping an international organization to be the official face and publisher of the Bitcoin blacklist.[8] Even with an international organization as the face, there will be social pushback against the blacklist and I suggest fielding a representative to the Bitcoin conferences to guide the larger miners into thinking Rationally as well as assuage knee-jerk reactions.

One caveat—and this is important. The blacklist should not stridently contravene miners' "community values". For example, blacklisting Edward Snowden's or Wikileaks' donation address would almost certainly cause a revolt. Fortunately, I don't imagine there's a shortage of nigh-universally loathed actors.

6 Conclusion

Our system, Iceman, is a novel, principled system leveraging technical innovation (generalized feather-forking, eq. (4)), and game-theory (Nash-shifting, Hofstadter-shifting, and block discouragement) to freeze blockchain-based assets such as Bitcoin.

Iceman is surprisingly cheap to maintain—the mere *credible offer* of a large reimbursement s is sufficient for all Rational miners to enforce the blacklist. As Rational miners constitute a considerable portion, perhaps a plurality, of the hashrate, this means that actually paying the reimbursement will be exceedingly rare. Although Superrational miners presumably constitute a smaller fraction, the cost ϵ of enlisting them into our fork is minuscule compared to the cost of losing a race. Additionally, even when we lose a race, *every transaction* from a blacklisted address requires an enormous transaction fee (Table 2) that will be prohibitive for all but the largest players.

Technically, Iceman could be developed and implemented by myself leading a team of 2–3 strong engineers. Socially, a representative (not me) should be fielded to reach out to the larger miners as well as handle the initial social pushback.

Altogether, Iceman is an opportunity to bring the illicit activities newly enabled by the "west wild" of cryptocurrencies to heel.

[7] A machine with 250GB of RAM and a 1GB/s internet connection would be sufficient.

[8] I defer to anyone who knows this space, but to me some reasonable candidates are INTERPOL, the International Money Laundering Information Network, and the International Monetary Fund.

References

[1] Nakamoto S (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org/bitcoin.pdf.

[2] "Captaintod" (2015). Active bitcoin addresses for isis. https://www.reddit.com/r/Bitcoin/comments/2j8mdg/active_bitcoin_addresses_for_isis/.

[3] Miller A (2013). Feather-forks: enforcing a blacklist with sub-50% hash power. https://bitcointalk.org/index.php?topic=312668.0.

[4] Bahack L (2013) Theoretical Bitcoin Attacks with less than Half of the Computational Power. Technical Report abs/1312.7013, CoRR.

[5] Nash J (1951) Non-cooperative games. Annals of Mathematics 54.

[6] Buterin V (2015). The $p + \epsilon$ attack. https://blog.ethereum.org/2015/01/28/p-epsilon-attack.

[7] Hofstadter D (1985) Dilemmas for superrational thinkers, leading up to a luring lottery. In: Metamagical Themas. Basic Books, pp. 737—755.

[8] Schelling TC (1960) The Strategy of Conflict. Harvard University Press.

[9] Stanley R, Weisstein EW (2016). Catalan number. from mathworld–a wolfram web resouce. http://mathworld.wolfram.com/CatalanNumber.html.

[10] Bearden JN (2001). Ultimatum bargaining experiments: The state of the art. http://ssrn.com/abstract=626183.

[11] Rosenfeld M (2012). Overview of Colored Coins. https://bitcoil.co.il/BitcoinX.pdf.

[12] Willett JR (2013). MasterCoin Complete Specification, v1.1.

[13] Coin Sciences Ltd (2016). Coin Secrets: Recent Metadata in the Bitcoin Blockchain. http://coinsecrets.org/.

[14] Bitcoin Core (2016). Bitcoin Core homepage. https://bitcoincore.org/.

Appendix

A Glossary of Notation

All conversions between Bitcoins and Dollars use Ƀ1.00 = $400.

Symbol	Description
X	One of the blacklisted Bitcoin addresses.
A	The fork of the blockchain obeying the blacklist.
B	The fork of the blockchain those head (most recent block) contains a transaction originating from a blacklisted address.
ω	The probability of fork A Winning the fork-race, i.e., the probability of successfully orphaning a block containing a banned transaction. $\omega \in [0, 1]$.
r	The total amount of Bitcoin a miner earns from mining a given block, i.e., $r \equiv$ block reward + transaction fees.
s	The amount of Bitcoin, on the order of $10,000, that the government offers to incentivize Rational miners to enforce the blacklist.
ϵ	The amount of Bitcoin, on the order of $1–5, that the government offers to incentivize Superrational miners to enforce the blacklist.
α	The proportion of mining hashrate that is legally persuaded or sufficiently incentivized to obey the blacklist. $\alpha \in [0, 1]$.
β	The proportion of mining hashrate that ignores the blacklist. $\beta \equiv 1 - \alpha$.
m	The number of confirMations built on top of the banned block after which to "give up" (if fork A hasn't won) and resume mining atop the longest chain. $m = \{x \in \mathbb{N} : x \geq 1\}$
r_X	The transaction fee address X offers to a miner in exchange for publishing its banned transaction.

Table 4: Definitions of the mathematical notation.

B Future Optimizations

B.1 Detecting Coordinated Counter-strategies

If only for economic reasons, a group of miners will eventually try to game ICEMAN into reimbursing more often. To understand the threat, imagine a hypothetical shadowy cabal who collectively control a substantial portion of the hashrate. In their ideal world, they will receive the reimbursement as often as possible, meaning our fork will *barely fail*. So, anytime the cabal sees a banned transaction, they all mine atop fork A. If a cabal member publishes the first block atop A, the whole cabal then pivots to mine atop fork B, hoping that B wins. If B wins, the government reimburses the cabal member who published atop fork A, and the profit, $s - r$, is split among all members.

Such a strategy would be immensely risky. First, we (the government) would inevitably catch on and announce there will be no further reimbursements to <list of suspected cabal members>. Moreover, modern proof-of-work mining is a capital intensive business requiring sinking substantial assets into ASIC rigs. When your business model requires playing for the long-haul, antagonizing the world's law enforcement is an obvious mistake. To paraphrase Singapore's founder Lee Kuan Yew, "They know that we can hurt them much more than they can hurt us."

B.2 Trustless Payments to Miners

Although it doesn't currently exist in Bitcoin, some altcoins and "Bitcoin 2.0" systems have an `OP_HEIGHT` (or equivalent) instruction that permits some ingenious transactions such as compensating depending on whether or not a specific block exists. This allows incentivizing Rational and Superrational miners even without their trust that the government will follow through on its promise of reimbursement. If trust becomes a sticking point, we can petition Bitcoin Core [14] to add such an instruction.

B.3 Social Shaming of Miners

Bitcoin records each miner who publishes a block. Ergo from the blacklist we can easily maintain a list of how often each miner has published blocks from blacklisted addresses, as well as who those actors are. E.g., "Miner Z enables Islamic State."

B.4 Mining Pool?

Dynamic mining pools may be allow us to further increase α (and thus ω). At first impression, it seems like this would require too much additional infrastructure for small gains. But ultimately this is an empirical, financial question. If ICEMAN is funded, we should still examine the cost-benefit trade-offs in instantiating mining pools.

C Yet Unresolved Mathematics

These are three remaining questions that are mathematically tractable, but require some hard probability and combinatorics. If there's sufficient interest in implementing ICEMAN I can start to work on solving these.

1. ~~What is the closed form expression for $\omega(m, \alpha)$ for all values of m?~~ SOLVED! See eq. (3).
2. ~~Given address X has paid the high transaction fee, what is the expected number of blocks until it will get lucky and be confirmed into the bitcoin blockchain?~~ SOLVED. Geometric Distribution.
3. What is the minimum transaction fee address X would have to pay for a Rational miner to publish a banned transaction?
4. If there is always a banned transaction in the queue, how much would the government expect to reimburse per annum? Provide as a function of m, α, s, ϵ, and the average number of minutes between blocks.
5. What is the exact algorithm for optimally feather-forking in the face of pathological circumstances? For example, say a malicious published a banned transaction on top of fork A? The naïve algorithm is simply to recurse into another fork-race, but there may be more efficient solutions. At whatever solution we choose, it'd be nice to prove that our algorithm is optimal.