

Cryptocurrencies without Proof of Work

Iddo Bentov¹, Ariel Gabizon², and Alex Mizrahi³

¹ Department of Computer Science, Technion, iddo@cs.technion.ac.il

² Department of Computer Science, Technion, ariel.gabizon@gmail.com

³ chromaway.com, alex.mizrahi@gmail.com

Abstract. We study decentralized cryptocurrency protocols in which the participants do not deplete physical scarce resources. Such protocols commonly rely on *Proof of Stake*, i.e., on mechanisms that extend voting power to the stakeholders of the system. We offer analysis of existing protocols that have a substantial amount of popularity. We then present our novel pure *Proof of Stake* protocols, and argue that they help in mitigating problems that the existing protocols exhibit.

1 Introduction[†]

The decentralized nature of Bitcoin [7, 12] means that anyone can become a “miner” at any point in time, and thus participate in the security maintenance of the Bitcoin system and be compensated for this work. The miners continuously perform *Proof of Work* (PoW) computations, meaning that they attempt to solve difficult computational tasks. The purpose of the PoW element in the Bitcoin system is to reach consensus regarding the ledger history, thereby synchronizing the transactions and making the users secure against double-spending attacks.

The miners who carry out PoW computations can be viewed as entities who vote on blocks of transactions that the users recently broadcasted to the network, so that the decision-making power of each miner is in proportion to the amount of computational power that she has. Thus, an individual miner who has a fraction p of the total mining power can create each new block with probability $\approx p$, though other factors such as “selfish mining” [1, 5, 6] can influence p .

Under the assumption that the majority of the PoW mining power follows the Bitcoin protocol, the users can become increasingly confident that the payment transactions that they receive will not be reversed [7, 12, 15].

By means of the PoW mechanism, each miner depletes physical scarce resources in the form of electricity and mining equipment erosion, and thereby earns cryptographic scarce resources in the form of coins that can be spent within the Bitcoin system.

Hence the following question is of interest: can a *decentralized* cryptocurrency system be as secure as Bitcoin even if the entities who maintain its security do not deplete physical scarce resources?

[†] The full version of this work includes extra material such as a section on the initial issuance of the money supply, and is available at <http://arxiv.org/abs/1406.5694>.

Cryptocurrency protocols that attempt to avoid wasting physical scarce resources commonly rely on *Proof of Stake*, i.e., on mechanisms that give the decision-making power regarding the continuation of the ledger history to entities who possess coins within the system. The rationale behind *Proof of Stake* is that entities who hold stake in the system are well-suited to maintain its security, since their stake will diminish in value when the security of the system erodes. Therefore, in an analogous manner to Bitcoin, an individual stakeholder who possesses p fraction of the total amount of coins in circulation becomes eligible to create the next extension of the ledger with probability $\approx p$.

We use the terminology “pure” *Proof of Stake* to refer to a cryptocurrency system that relies on *Proof of Stake* and does not make any use of PoW. To the best of our knowledge, the idea of *Proof of Stake* in the context of cryptocurrencies was first introduced in [17], though that discussion focused on non-pure *Proof of Stake* variants (cf. [3]).

PoW based cryptocurrencies become insecure when a significant enough portion of the total mining power colludes in an attack. Likewise, the security of pure *Proof of Stake* cryptocurrencies deteriorates when enough stakeholders wish to collude in an attack. If the majority of the stake wishes to participate in attacks on a pure *Proof of Stake* system, it can be argued that there is no longer enough interest that this system should continue to exist, hence assuming that the majority of the stake will not participate in an (overt) attack is sensible. The same does not necessarily hold in a PoW based system, i.e., the majority of the mining power might be under the control of an external adversary during some time period, while the majority of the participants in this system still wish for it to remain sound. See [3] and Section 3 for additional considerations.

2 Pure *Proof of Stake*

There are two apparent hurdles with decentralized pure *Proof of Stake* systems: fair initial distribution of the money supply to the interested parties, and network fragility if the nodes are rational rather than altruistic. PoW offers an elegant solution to the first hurdle, by converting physical scarce resources into coins in the system. We provide here an analysis of the second hurdle in an existing pure *Proof of Stake* system, and also describe our novel CoA and Dense-CoA pure *Proof of Stake* systems that seek to mitigate this problem. Let us note this second hurdle is less severe in PoW systems, though bribe attacks on Bitcoin have indeed been considered, for example in [16].

2.1 The PPCoin system

PPCoin is a pure *Proof of Stake* system, in the sense that PoW is used only⁵ for distributing the initial money supply. Stakeholders in the PPCoin network can

⁵ See <http://peercoin.net/assets/paper/peercoin-paper.pdf>.

create the next block according to the following type of condition:

$$\text{hash}(\text{prev_blocks_data}, \text{time_in_seconds}, \text{txout}_A) \leq d_0 \cdot \text{coins}(\text{txout}_A) \cdot \text{timeweight}(\text{txout}_A) \quad (*)$$

In the inequality (*), `time_in_seconds` should correspond to the current time (with some leniency bounds), thus restricting hash attempts to 1 per second and preventing PoW use at creating the next block, because nodes will regard a new block as invalid unless the difference between its time and their local time is within the bounds. The notation `coins(txoutA)` refers to the amount of coins of some unspent transaction output `txoutA`, hence if stakeholder *A* has the private key `skA` that controls `txoutA` then she can create a valid block by signing the block with `skA` and attaching the signature as evidence that condition (*) holds. This means that a stakeholder who controls an output of e.g. 50 coins is 10 times more likely to create a block than a stakeholder who controls an output of 5 coins. See † regarding `timeweight(txoutA)`, and Section 2.1.3 regarding `prev_blocks_data`. The constant `d0` is readjusted according to a protocol rule that dictates that blocks should be created in intervals of 10 minutes on average, i.e., if fewer stakeholders are online during a certain time period then `d0` gets increased. The winning blockchain is the one with the largest cumulative stake, i.e., the blockchain with the most blocks such that stake blocks are weighted according to their `d0` difficulties, and PoW blocks have a negligible weight.

Although the PPCoin cryptocurrency had a market cap of over \$100 million in 2014, the PPCoin protocol has the following problems:

2.1.1 Rational forks

On every second we have that $\Pr[\{\text{some block is solved}\}] \approx \frac{1}{600}$, therefore multiple blocks will be solved simultaneously every ≈ 360000 seconds ≈ 4 days. Rational stakeholders can increase their expected reward by maintaining and trying to solve blocks on the multiple forked chains that were transmitted to them, which would lead to a divergent network. An individual stakeholder can either tie her hands behind her back by ignoring all the forked chains except for one, or opt to gain more rewards by keeping all the forked chains, which may render her entire stake worthless in case the network becomes divergent. The strategy of tying your hands behind your back is not a Nash equilibrium: if all the stakeholders follow this strategy then it is better for an individual stakeholder to deviate and maintain all the forked chains, as her influence on the overall convergence of the network is minor. Network propagation lag implies an even greater frequency of forks, as a stakeholder will get competing blocks sent to her even if those blocks were honestly solved a few seconds apart from one another. Worse still, when a rational stakeholder who currently tries to extend the block B_i receives B_{i+1} from her peers, she may opt to increase her expected reward by attempting to extend both the chain \dots, B_i, B_{i+1} and the chain \dots, B_i simultaneously. Rational stakeholders may thus prefer to reject blocks whose timestamp is later than another block that they currently try to extend, though an attempt to extend both \dots, B_i, B_{i+1} and \dots, B_i can still be

possible if the rule that the stakeholders deploy does not retrace to an earlier chain that is received late due to propagation lag.

2.1.2 Bribe attacks on PPCoin

An attacker can double-spend quite easily. After the merchant waits for e.g. 6 block confirmations and sends the goods, the attacker can publicly announce her intent to create a fork that reverses the last 6 blocks, and offer bribes to stakeholders who would sign blocks of her competing branch that starts 6 blocks earlier. The attacker may offer a larger bribe to stakeholders who sign only her branch, and may commit to giving bribes even after her competing branch wins, to encourage more stakeholders to participate in the attack. Notice that the stakeholders who collude with the attacker will not lose anything in case the attack fails. As long as the value of the goods is greater than the total value of the bribes, this attack will be profitable. Let us note that a bribe attack in a pure PoW network has to surmount far greater obstacles: miners who join the attack would deplete their resources while working on a fork with a 6 blocks deficit, and it is a nontrivial task to assess the success probability by measuring how many other miners participate in the attack. See also [3, Section 5.3].

2.1.3 Opportunistic attacks in relation to the need to disallow PoW

A stakeholder who holds a significant fraction of all the coins is able to generate a significant fraction of the blocks, as the probability to generate a block is proportional to the amount of coins that a stakeholder holds. Therefore, from time to time a stakeholder will be able to generate chains of consecutive blocks.

We can analyze this event by using a simplified model where stakeholders who own $\frac{1}{M}$ of all coins can generate a block with probability $\frac{1}{M}$, and the probability to generate k sequential blocks is $(\frac{1}{M})^k$. This approximation is accurate under the assumption that the stakeholder holds a number of unspent transaction outputs significantly larger than k , so that `timeweight` will have no impact. We can estimate the average number of blocks between groups of k sequential blocks generated by one stakeholder as a mean of exponential distribution, which would be equal to $1/(1/M)^k = M^k$.

If merchants wait for k confirmations before sending their goods, the stakeholder has a chance to attack the merchant when she is able to generate k sequential blocks, thus the mean number of blocks between such attacks is M^k . For example, a stakeholder who holds $\frac{1}{4}$ of all coins participating in stake mining will be able to carry out a 6-block reorganization each $4^6 = 4096$ blocks, i.e., approximately once per month if one block is generated every 10 minutes.

An attacker who is able to create k sequential blocks would prefer to know about it as early as possible, so that she has enough time to send the payment transaction (that she intends to reverse) to the merchant. If the possible stakeholders' identities who may create the next blocks are derived from a low entropy process that only takes into account the identities who created the previous blocks, then the attacker can "look into the future" by carrying out brute-force

computations to assess the probabilities that she will be able to create the k consecutive blocks at certain points in time. In order to gain a measure of unpredictability, PPCoin re-calculates once every 6 hours a “stake modifier” value that depends on the transactions that the previous blocks included, i.e., this stake modifier is part of `prev_blocks_data` in condition (*). Therefore, a stakeholder who obtains an opportunity to generate k blocks in a row can know about this approximately 6 hours in advance, so she has plenty of time to mount an attack. If the protocol required the stake modifier to be re-calculated at a shorter time interval, this would open the door for a stakeholder to do PoW attempts at deriving herself as being able to create future blocks more frequently.

2.2 The CoA pure *Proof of Stake* system

The Chains of Activity (CoA) system that we hereby present is a pure *Proof of Stake* protocol that aims to overcome the problem of rational forks (cf. Section 2.1.1) by dictating that only a single stakeholder identity may create the next block, and solidifying the random choices for these identities in the earlier ledger history via an interleaving mechanism.

The CoA protocol is based in part on the core element of PoA [3], i.e., on a lottery among the online stakeholders via the *follow-the-satoshi* procedure. This procedure takes as input an index of a satoshi (smallest unit of the cryptocurrency) between zero and the total number of satoshis in circulation, fetches the block of ledger data in which this satoshi was minted, and tracks the transactions that moved this satoshi to subsequent addresses until finding the stakeholder who can currently spend this satoshi (cf. [3, Section 3 and Appendix A]). Note that if for example Alice has 6 coins and Bob has 2 coins then Alice is 3 times more likely to be picked by *follow-the-satoshi*, regardless of how their coins are fragmented. This implies that a stakeholder who holds her coins in many Sybil addresses do not obtain any advantage with regard to *follow-the-satoshi*.

The CoA protocol is parameterized by an amount of minted satoshis 2^κ , a subgroup length $w \geq 1$, a group length $\ell = \kappa \cdot w$, a function $\mathbf{comb} : \{0, 1\}^\ell \rightarrow \{0, 1\}^\kappa$, a minimal block interval time G_0 , a minimal stake amount C_0 , an award amount C_1 where $0 \leq C_1 < C_0$, and a double-spending safety bound T_0 .

The blocks creation process of CoA assembles a blockchain that is comprised of groups of ℓ consecutive blocks:

$$\overbrace{\square \square \dots \square}^\ell, \overbrace{\square \square \dots \square}^\ell, \overbrace{\square \square \dots \square}^\ell, \dots$$

The rules of the CoA protocol are specified as follows:

The CoA Protocol

1. Each block is generated by a single stakeholder, whose identity is fixed and publicly known (as will be explained in the next steps). This stakeholder collects transactions that are broadcasted over the CoA network as she sees

fit, and then creates a block B_i that consists of these transactions, the hash of the previous block, the current timestamp, the index i , and a signature of these pieces of data as computed with her private key.

2. Every newly created block B_i is associated with a supposedly uniformly distributed bit b_i that is derived in a deterministic fashion, for example by taking the first bit of $hash(B_i)$.
3. The time gap between B_i and B_j must be at least $|j - i - 1| \cdot G_0$. This means that if for example the next four blocks $B_i, B_{i+1}, B_{i+2}, B_{i+3}$ were supposed to be generated by the four stakeholders $A_i, A_{i+1}, A_{i+2}, A_{i+3}$ but A_{i+1} and A_{i+2} were inactive, then the difference between the timestamp of B_{i+3} and B_i must be at least $2G_0$. Nodes in the network will consider a newly created block to be invalid if its timestamp is too far into the future relative to their local time.
4. After a group of ℓ valid blocks $B_{i_1}, B_{i_2}, \dots, B_{i_\ell}$ is created, the network nodes will form a κ -bit seed $S^{B_{i_\ell}} = \mathbf{comb}(b_{i_1}, \dots, b_{i_\ell})$. The function \mathbf{comb} can simply concatenate its inputs (if $w = 1$), and several other alternatives are explored in Section 2.2.1.
5. The seed $S^{B_{i_\ell}}$ is then used in an interleaved fashion to derive the identities of the *after* next ℓ stakeholders, via *follow-the-satoshi*. That is, if the next ℓ valid blocks are $B_{i_\ell+j_1}, B_{i_\ell+j_2}, \dots, B_{i_\ell+j_\ell}$, then the nodes who follow the protocol will derive the identity of the stakeholder who should create the block $B_{i_\ell+j_\ell+z}$ by invoking *follow-the-satoshi* with $hash(i_\ell, z, S^{B_{i_\ell}})$ as input, for $z \in \{1, 2, \dots\}$.
6. If the derived satoshi is part of an unspent output of $c < C_0$ coins, the stakeholder must also attach an auxiliary signature that proves that she controls another output of at least $C_0 - c$ coins, or else she will not be able to create a valid block. Neither the derived output nor this auxiliary output may be spent in the first T_0 blocks that extend the newly created block. In case the stakeholder A_i who should create the i^{th} block signs two different blocks B_i, B'_i , any stakeholder A_j among the next T_0 derived stakeholders can include it as evidence in the block that she creates, in order to confiscate at least C_0 coins that A_i possessed. The stakeholder A_j is awarded with C_1 of the confiscated coins, and the rest of the confiscated coins are destroyed.
7. If the network nodes see multiple competing blockchains, they consider the blockchain that consists of the largest number of blocks to be the winning blockchain.

The interleaving in step 5 is crucial as a cementing mechanism. Otherwise, competing last stakeholders may extend the chain with seeds that derive different ℓ next identities, introducing divergence risk because it is rational for the next identities to extend the different forks. This cementing process ensures that unless $\approx \ell$ stakeholders collude by bypassing their turn on the honest chain and creating a hidden fork instead, only a single stakeholder will be eligible to create each next block. Thus the rational forks hazard is avoided.

The punishment scheme in step 6 expires after T_0 blocks, because honest stakeholder must eventually regain control over their security deposit (see also

Section 3). Note that a stakeholder can divide her coins among multiple outputs, so that only one of the outputs would become unspendable for T_0 blocks. If $C_1 \approx C_0$, an attacker might double-sign and publish the double-signing evidence in a next block to recover her security deposit, so $C_1 \leq \frac{C_0}{2}$ is a better choice.

If ℓ is very large (in the extreme $\ell = \infty$, i.e., practically equivalent to selecting the identities of the stakeholders via a round-robin), then an attacker may try to gain possession of future consecutive satoshis to mount a double-spending attack (cf. Section 2.2.3). On the other hand, small ℓ makes it easier for coalitions to influence the future identities (cf. Section 2.2.1). Moreover, if the range of **comb** were $\kappa' < \kappa$, an attacker could more easily see into the future, e.g. with $\kappa' = 10$ the attacker could buy satoshis of consecutive identities in one possible next group and succeed with probability $1/1024$ to carry out a double-spending attack. A sensible recommendation for the CoA parameters can be $\kappa = 51$ (for ≈ 21 million coins of 10^8 satoshis each), $w = 9$ with **comb** as the iterated majority function (see Section 2.2.1), $\ell = 459$, $G_0 = 5$ minutes, and $T_0 = 5000$.

2.2.1 Using low-influence functions to improve chain selection

To give an intuitive illustration of the advantages of different choices, we focus on the prominent case of analyzing the probability that the last stakeholder in the chain, A_ℓ , can choose herself again as one of the first possible stakeholders A'_1, \dots, A'_ℓ of the next round (see Figure 1). Denote this probability by μ . We also make the simplifying assumptions that the previous players have indeed picked their bits b_i randomly, and that the function *hash* is a random oracle. Let us assume that A_ℓ has a q -fraction of the coins in the system, and denote $p = 1 - (1 - q)^\ell$. Thus, $\mu = p$ in case A_ℓ picks a random bit.

Simple concatenation: We let $\mathbf{comb}(b_1, b_2, \dots, b_\ell) \triangleq b_1 \circ b_2 \circ \dots \circ b_\ell$, where b_i is the supposedly random bit that stakeholder A_i provided. The probability μ that A_ℓ can choose herself in the next round is the probability that $\exists b' \in \{0, 1\}, i \in \{1, \dots, \ell\}$ such that $\mathit{hash}(i_0 + i, \mathbf{comb}(b_1, \dots, b_{\ell-1}, b'))$ maps to a coin of A_ℓ under *follow-the-satoshi*. Using the simplifying assumption that these are random independent values we have $\mu = 1 - (1 - p)^2 = 2p - p^2 \approx 2p$.

Combining majority with concatenation: Assume now that $\ell = \kappa \cdot w$ for positive integer w . We now split the ℓ stakeholders into groups of size w : $A_1, \dots, A_w, A_{w+1}, \dots, A_{2w}, \dots, A_{(\kappa-1) \cdot w+1}, \dots, A_{\kappa \cdot w} = A_\ell$. Each group will determine a bit of the seed using the majority function. That is, the i^{th} bit of the seed, denoted s_i , will be the majority of the bits $b_{(i-1) \cdot w+1}, \dots, b_{i \cdot w}$. And $s = \mathbf{comb}(b_1, \dots, b_\ell) \triangleq s_1 \circ s_2 \circ \dots \circ s_\kappa$. Note first that when the bits b_i are all chosen randomly, s is random – as the majority of random inputs is a random bit. Now, we analyze again the probability μ that A_ℓ can choose herself in the next round. It can be shown, using Stirling’s approximation, that with probability roughly⁶ $1 - \sqrt{2/\pi w}$ the last bit of the seed, s_κ , will already be determined

⁶ More precisely, as w goes to infinity this is the limit of the probability of the event.

by the bits of the previous stakeholders. This is because when w players choose a bit randomly, the probability that *exactly half* of the bits came out one tends to $\binom{w}{w/2}/2^w \approx \frac{2^{w+1/2}}{\sqrt{\pi w}}/2^w = \sqrt{2/\pi w}$. In the absence of this event we have $\mu = p$, “as it should be”. When this event happens, as before A_ℓ can get to probability $\approx 2p$. In total we have $\mu \approx p \cdot (1 - \sqrt{2/\pi w}) + 2p \cdot \sqrt{2/\pi w}$. Taking a large enough w , this is much closer to the “correct” p than in the previous choice of **comb**.

Protection against larger coalitions: Let us use the terminology that a function **comb** : $\{0, 1\}^\ell \rightarrow \{0, 1\}^\kappa$ is an ε -*extractor* if for any choice of the coalition C of size c , and any strategy of C to choose their input bits after seeing the bits of the honest players, **comb**(b_1, \dots, b_ℓ) produces an output that is ε -close to uniform.

[9], using the analysis of [2], give the following construction of an ε -extractor – that is in fact the same one we described earlier when replacing the majority function with *iterated majority* (defined in [2] and illustrated in Figure 1).

$KZ(b_1, \dots, b_\ell)$:

- Choose $w = 3 \cdot (c/\varepsilon)^{1/\alpha}$, where $\alpha = \log_3 2$. Set $\ell = w \cdot \kappa$.
- Output κ bits via *iterated majority* of consecutive groups of w inputs.

Upon fixing ε as the desired statistical error, κ as the desired output length, and ℓ as the total number of players in a chain, KZ can handle a coalition of size $c \leq \varepsilon \cdot (1/3 \cdot \ell/\kappa)^\alpha$. On the other hand, [9] show that any such ε -extractor can handle coalitions of size at most $c \leq \varepsilon \cdot 10 \cdot \ell/(\kappa - 1)$.

Since $\alpha > 1/2$, it follows that this choice of **comb** is less than quadratically worse than the optimal choice. Notice that this assumes that stakeholders who are not honest are non-oblivious, i.e., that they see the choices of the honest stakeholders before they play. This conservative assumption makes a certain sense in our context, as it easier for stakeholders who play in the last locations to try to collude in order to influence the seed.

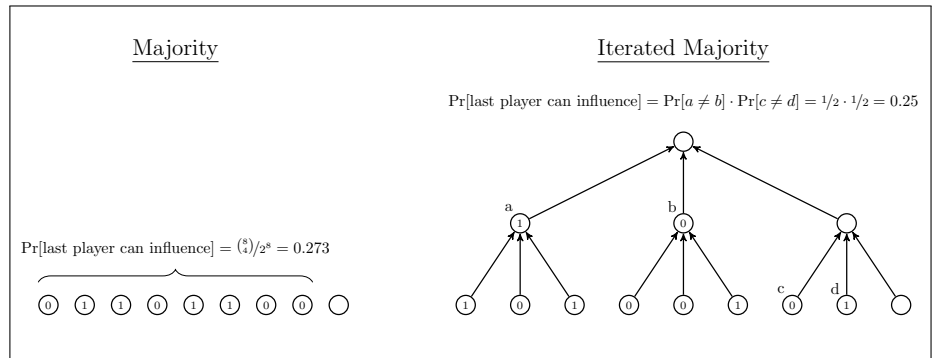


Fig. 1. Majority versus Iterated Majority.

2.2.2 Rational collusions

Stakeholders may wish to collude and skip the last several blocks as if they did not exist, i.e., to extend the blockchain from an earlier block, in order to gain the fees that went to previous stakeholders. This can be mitigated by including in each transaction the index of the latest block that the user who made this transaction is aware of. For example, if the last block of the chain is B_i and it contains a transaction tx_0 that specifies that block $i - 1$ exists, and a new transaction tx_1 that specifies that block i exists is broadcasted, then the stakeholder who creates B_{i+1} cannot reverse B_i to collect the fees of both tx_0 and tx_1 , because B_i must exist in the chain that contains tx_1 . The user can even specify in her transaction the index of the block that is currently being created, but this implies that the user will need to send another transaction in the case that the current stakeholder is offline. The colluding stakeholders diminish the overall value of their stake when they participate in such attacks, hence this strategy is not necessarily rational. It is also possible to reward stakeholders via monetary inflation and have the transaction fees destroyed to provide a counterbalance, though bribe attacks may then become more likely (see Section 2.2.3).

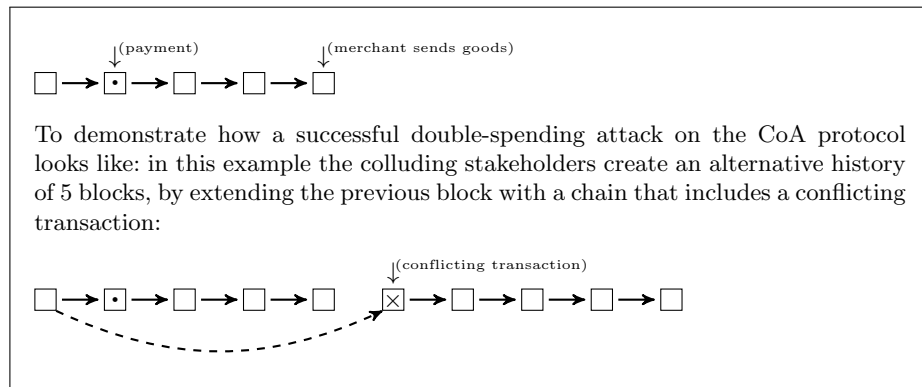


Fig. 2. Illustration of a double-spending attack in the CoA system.

2.2.3 Bribe attacks on CoA

Suppose that the number of blocks that merchants consider to be secure against double-spending attacks is d , i.e., a merchant will send the goods after she sees that the payment transaction that she received in block B_{i_1} has been extended by $B_{i_2}, B_{i_3}, \dots, B_{i_d}$ extra blocks. An attacker can now offer bribes to $d + 1$ or more stakeholders, for example to the next $i_d + 1, i_d + 2, \dots, i_d + d + 1$ stakeholders so that they would extend the blockchain starting from the block that preceded B_{i_1} and exclude that payment transaction. The attacker will need to bribe more than $d + 1$ stakeholders if some of them refuse the bribe. Since rational stakeholders will not participate in the attack without an incentive, the cost of the attack

is at least $\mu(d + 1)$ where μ is the average bribe amount that is given to each stakeholder.

Observe that $\Pr[\{\text{successful attack}\}] < 1$ since some of the stakeholders might be altruistic, some of the rational stakeholders may think that it would be unprofitable to participate in such attacks, and the attacker's funds are not unlimited. Hence, a rational stakeholder will choose to accept the bribe by weighing whether $(\mu + F') \cdot \Pr[\{\text{successful attack}\}] > F \cdot (1 - \Pr[\{\text{successful attack}\}])$, where F and F' are the fee amounts that this stakeholder will collect on the honest chain and the attacker's chain, respectively. Note that $F' = 0$ is likely when the safety mechanisms of Section 2.2.2 are deployed, since it is rational for users to continue to transact on the honest chain as long as the attacker's chain is inferior. Overall, the attacker may need to spend substantially more than $\mu(d + 1)$ coins for the attack to succeed.

In Figure 2 we illustrate the nature of a double-spending bribe attack.

The above stands in stark contrast to Section 2.1.2, as the short-term dominant strategy of the PPCoin stakeholders is to participate in the attack, while the CoA stakeholders will forfeit their reward F if the attack fails. In our setting, the premise of a short-term strategy can be regarded to be that the utility per coin is constant, while the premise of a long-term strategy can be regarded to be that the utility per coin may change due to actions taken by the player.

Notice that the attacker cannot simply bribe the stakeholders who generated the blocks $B_{i_1}, B_{i_2}, B_{i_3}, \dots, B_{i_d}$ to create an alternative history of length d in a risk-free manner, as their coins will be confiscated if they double-sign.

Formally, let us restrict ourselves to a limited strategy space (cf. [10]) in which players have to choose one of only these two actions (**),

1. Follow the protocol honestly by signing a block that extends the longest known chain.
2. Accept bribe and sign the attacker's block which extends the secretive chain that the attacker builds.

This restriction can be justified under plausible assumptions. In particular, the C_0 penalty can be assumed to be high enough to make the action of double-signing unappealing. This requires the presupposition that the double-signing punishment mechanism is effective in the sense that the evidence of double-signing will be recorded on every fork, and hence the utility of a player is the value of attacker's bribe minus the loss of her C_0 security deposit. This also implies that our analysis here only covers forks that are shorter than the T_0 deposit duration, in Section 3 we discuss attacks that involve longer forks.

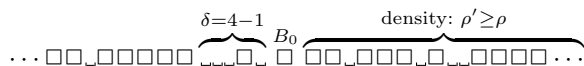
Our objective is to show that the honest strategy is dominant. In fact, we will show that under further assumptions no attack will be initiated, thus only the honest action will be available to the players.

To analyse what merchants can consider to be an appropriate confidence level for security against double-spending in the CoA system, let us make a reasonable assumption regarding the participation rate of stakeholders in the CoA network.

Density assumption. Let $\rho > 1/2$. In the longest blockchain, for every segment of K or more potential blocks, at least ρK of those blocks were created.

While this is a simplifying assumption, it is indeed reasonable, as our presupposition for the CoA network is that its security is derived from stakeholders' participation. Notice that we do not assume that the majority of stakeholders are altruistic (i.e., follow the CoA protocol even if it is against their self-interest). Although an altruistic majority would facilitate a system with better security, a rational majority is far more likely to capture reality.

Let B_0 be a block in which some particular payment transactions resides. Let δ denote the amount missing blocks in largest segment with participation rate $\leq 1/2$ prior to B_0 , and let ρ' denote the density of the longest segment that follows B_0 . In the illustration below, $\delta = 3$ and $\rho' = 10/14$.



Claim 1. Let ε be the average fee amount that a stakeholder earns for creating a block. Assume that stakeholders are restricted to the strategy space (**). Assume that reversing B_0 has a value of V coins to the attacker. If the attacker is rational in the sense that she does not wish to lose coins, then the merchant is safe by waiting until S blocks extend B_0 before sending the merchandise, for S that satisfies $V < \varepsilon(\rho'S - \delta + 1)$.

Proof. By using the safety extension that is described in Section 2.2.2, we may consider the blocks in a hostile competing fork to be void of transactions, and therefore it is rational of each colluding stakeholder who could otherwise earn ε coins to demand a bribe of more than this amount. There exist $(1 - \rho')S + \delta$ stakeholders who can contribute to the attack and have already forfeited their turn to create a block, thus the merchant may assume that in the worst case they will collude with the attacker for free. As the other $S + 1$ stakeholders need to be bribed with ε coins each, $V < \varepsilon(S - (1 - \rho')S - \delta + 1) = \varepsilon(\rho'S - \delta + 1)$ implies that the attack is unprofitable. \square

The above argument gives only a crude bound, since it does not capture all the relevant aspects w.r.t. the attack. In particular, the coins that the attacker recovers (in the case of a successful attack) may have less purchasing power, because the cryptocurrency system becomes less valuable whenever double-spending attacks take place.

Claim 2. If the density assumption holds in addition to the assumptions of Claim 1, then the merchant can be confident that it is irrational to carry out a double-spending attack after B_0 has been extended by S blocks, for S that satisfies $V < \varepsilon(\rho S - K + 1)$.

Proof. According to the density assumption, it holds that $K > \delta$, and since the merchant waited until more than K blocks extend B_0 it also holds that $\rho' \geq \rho$. Therefore, $(1 - \rho)S + K \geq (1 - \rho')S + \delta$, and the result follows from Claim 1. \square

To get a better sense of things, let us substitute concrete numbers for the above parameters. Suppose for example that $\rho = 7/10$, $K = 20$, $\varepsilon = 10$ coins, and

$V = 100$ coins. Hence $10 \cdot (7/10 \cdot S - 19) > 100$ implies that $S = 42$ blocks are sufficient. This means that the merchant will need to wait $\leq 42 \cdot 5$ minutes or 3.5 hours before sending the merchandise, in case CoA is parameterized according to $G_0 = 5$ minutes.

2.2.4 Majority takeover

Consider some stakeholders A_1, A_2, \dots, A_m who control all of the first ℓ locations in the current round. Suppose that these m stakeholders possess p -fraction of the *total* stake, and they wish to collude and control all the locations in all of the next rounds, thereby creating a winning chain that consists of *only* their blocks. While this strategy may be irrational as it diminishes the value of their stake, perhaps the m stakeholders prefer a competing system and wish to destroy CoA.

Due to interleaving (cf. Section 2.2), the starting condition for this attack is more difficult to achieve, as these m stakeholders need to control 2ℓ locations.

Suppose that q -fraction of the honest stake is offline, hence the m stakeholders can give on average a head start of $(\frac{1}{(1-p)(1-q)} - 1)\ell$ blocks to a competing group in each round. Denote $\hat{q} \triangleq (\frac{1}{(1-p)(1-q)} - 1)$. Let Y be the random variable that counts how many of the first $(2 + \hat{q})\ell$ locations of the next round will be controlled by the m stakeholders, so $E[Y] = (2 + \hat{q})\ell p$. Using tail inequality, it holds that

$$\Pr(Y > \ell) = \Pr(Y > \frac{1}{(2 + \hat{q})p} E[Y]) \leq \exp\left\{-\left(\frac{1}{(2 + \hat{q})p} - 1\right)^2 \cdot (2 + \hat{q})\ell p \frac{1}{3}\right\}.$$

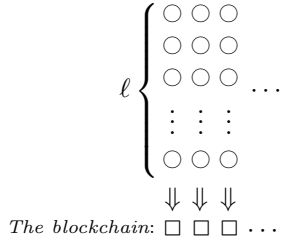
Thus, the amount of hash invocations that these m stakeholders need to compute tends toward infeasibility when p is smaller or when ℓ is larger. For example, with $\ell = 459, p = 1/10, q = 1/5$, the m stakeholders will need more than $e^{371} \approx 2^{535}$ hash attempts on average.

Compared with Bitcoin, in a *Proof of Stake* based system such as CoA it is less reasonable to assume that a large combined stake is an hostile external attacker (see [3, Section 2.1]), hence p is likely to be small.

2.3 The Dense-CoA pure *Proof of Stake* variant

The Dense-CoA pure *proof of stake* protocol is an alternative variant of CoA in which the identities of stakeholders who should create the next blocks are not known far in advance, with the objective of making collusions and bribe attacks more difficult. Another plus point of Dense-CoA is that it makes it more difficult for rational stakeholders to obtain disproportional rewards. The disadvantages of the Dense-CoA protocol are susceptibility to DoS attacks by large stakeholders, and greater communication and space complexities.

In Dense-CoA, each block is created by a group of ℓ stakeholders, rather than by a single stakeholder:



Let $h : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a one-way permutation. Let us assume for a moment that the block B_{i-1} is associated with a seed $S^{B_{i-1}}$ that was formed by the ℓ stakeholders who created B_{i-1} . Now, the identity of the stakeholder A_ℓ who determines which transactions to include in a block B_i is derived by invoking *follow-the-satoshi* with $hash(i, \ell, S^{B_{i-1}})$ as input, and the identities of the rest of the stakeholders $A_1, A_2, \dots, A_{\ell-1}$ who must participate in the creation of B_i are derived by invoking *follow-the-satoshi* with $hash(i, j, S^{B_{i-1}})$ for $j \in \{1, 2, \dots, \ell - 1\}$. These ℓ stakeholders engage in a two-round protocol to create the current block B_i :

- In round 1, for every $j \in \{1, 2, \dots, \ell\}$, the stakeholder A_j picks a random secret $R_j \in \{0, 1\}^n$, and broadcasts $h(R_j)$ to the network.
- In round 2, for every $j \in \{1, 2, \dots, \ell - 1\}$, the stakeholder A_j signs the message $M \triangleq h(R_1) \circ h(R_2) \circ \dots \circ h(R_\ell)$, and broadcasts her signature $\text{sign}_{sk_j}(M)$ and her preimage R_j to the network.

We require Dense-CoA to use a signature scheme with multisignature [4, 8, 11, 13] support, therefore A_ℓ can aggregate the signatures $\{\text{sign}_{sk_j}(M)\}_{j=1}^\ell$ into a single signature $\hat{s}(M)$. Note that the size of $\hat{s}(M)$ depends only on the security parameter of the signature scheme (and not on ℓ), and the verification time is faster than verifying ℓ ordinary (ECDSA) signatures.

Hence, the stakeholder A_ℓ signs and broadcasts a block B_i that consists of the (Merkle root of the) transactions that she wishes to include, the hash of the previous block B_{i-1} , the current timestamp, the index i , the ℓ preimages R_1, R_2, \dots, R_ℓ , and $\hat{s}(M)$. To verify that the block B_i is valid, the network nodes invoke h to compute the images $h(R_1), h(R_2), \dots, h(R_\ell)$, then concatenate these images to form M , and then check that $\hat{s}(M)$ is a valid signature of M with respect to the public keys $pk_1, pk_2, \dots, pk_\ell$ that control the winning satoshis of the stakeholders A_1, A_2, \dots, A_ℓ .

The seed S^{B_i} is defined as $hash(R_1 \circ R_2 \circ \dots \circ R_\ell)$. Notice that S^{B_i} is computationally indistinguishable from random even if only a single stakeholder A_j picked a random R_j , under the assumption that n is sufficiently large so that the OWP h is resistant to preimage attacks.

If some of the ℓ stakeholders are offline or otherwise withhold their signatures, then after G_0 time the nodes who follow the protocol will set $t = 1$ and derive alternative ℓ identities from the previous block B_{i-1} , by invoking *follow-the-satoshi* with inputs $hash(i, t\ell + j, S^{B_{i-1}})$ for $j \in \{1, 2, \dots, \ell\}$. The starting index

$t\ell + j$ should be specified in the new block B_i so that the verification of blocks will be simpler, and the gap between the timestamps of B_{i-1} and B_i must be at least tG_0 . As with CoA, the honest nodes consider the blockchain with the largest amount of valid blocks to be the winning blockchain, and disregard blocks with a timestamp that is too far into the future relative to their local clock.

The parameters C_0, C_1, T_0 of the CoA protocol (cf. Section 2.2) are utilized by the Dense-CoA protocol in exactly the same way.

The parameter ℓ should be big enough in order to resist large stakeholders from controlling consecutive seeds $\{S^{B_i}, S^{B_{i+1}}, \dots\}$ and re-deriving themselves. For example, to force a stakeholder who holds 5% or 10% of the total stake into making $\approx 2^{100}$ *hash* invocations on average until re-deriving herself as all of the ℓ identities of the next block, we need $\ell = 23$ or $\ell = 30$, respectively. However, if we set $G_0 = 5$ minutes and $\ell = 23$, a malicious stakeholder with e.g. 10% of the total stake will have $1 - (90/100)^{23} \approx 91\%$ probability to be one of the derived stakeholders A_1, A_2, \dots, A_ℓ and then refuse to participate in creating the next block, hence it will take $5 \cdot (1 - 91\%)^{-1} \approx 56$ minutes on average to create each next valid block while this attack is taking place (actually less than 56 minutes because chains that extend blocks prior to the last block can also become the longest valid chain).

Overall, the main difference between the Dense-CoA and CoA protocols is that Dense-CoA offers improved security over CoA in terms of double-spending attacks, but weaker security against DoS attacks by large stakeholders who wish to harm the cryptocurrency. Also, Dense-CoA prevents a rational stakeholder from influencing the seed in an attempt to earn more rewards than her fair share, unless she colludes with all the other $\ell - 1$ stakeholders who create the next block. The Dense-CoA protocol is less efficient than CoA due to the preimages R_1, R_2, \dots, R_ℓ that need to be stored in each valid block, and the two-round protocol that requires a greater amount of network communication to create each successive block.

3 Solidification of the ledger history

Any decentralized cryptocurrency system in which extending the ledger history requires no effort entails the danger of costless simulation [14], meaning that an alternative history that starts from an earlier point of the ledger can be prepared without depleting physical resources and hence without a cost. This is a problem because a rational adversary who has little or no stake in the system may try to attack by replacing an arbitrarily long suffix of the current ledger history with an alternative continuation that benefits her. Further, a malicious adversary who does not operate out of self-interest is also more likely to attempt this kind of an attack, as she would not incur a monetary loss for executing the attack.

In the case of pure *Proof of Stake* systems, this danger can manifest itself in the following form. Consider participants who held coins in the system a long time ago and have since traded those coins in exchange for other goods, so they are no longer stakeholders of this system. These participants can now collude to extend the ledger from the point at which they had control over the system, and

it may indeed be rational for them to mount this attack because it is costless and would have no detrimental outcome from their standpoint, as they have no stake in the current system.

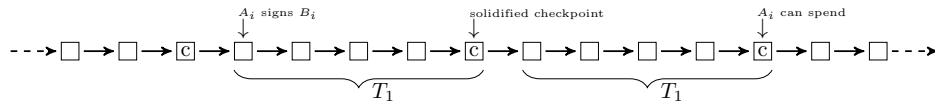
More specifically, let us examine how this attack looks like in the CoA or Dense-CoA systems. Even a single stakeholder with few coins can fork the blockchain and create an alternative branch with large enough time gaps as she re-derives herself to create subsequent blocks, but according to the timestamp rules for valid blocks, the other participants will reject this alternative branch (even though it contains more blocks) because the timestamps will be too far ahead in the future relative to their local time. Therefore, if the average participation level among current stakeholders is $p\%$, and the stakeholders who collude to carry out this attack have had control at the earlier history over $q\%$ of the coins, then $q > p$ implies that the attack will succeed. Because $p\% = 1$ is highly unlikely, and collusion among participants who held $q\% > p\%$ stake at an earlier point is costless and rational, this attack vector appears to be quite dangerous.

To mitigate this attack, we propose periodic checkpointing as a rigid protocol rule that extends the CoA and Dense-CoA protocols, as follows:

- Denote by $T_0 = 2T_1$ the double-spending safety bound of Section 2.2.
- The blocks at gaps of T_1 are designated as *checkpoint* blocks: the genesis block is a checkpoint block, and any block that extends a checkpoint block by exactly T_1 additional blocks is a candidate checkpoint block.
- When a node that follows the protocol receives for the first time a candidate checkpoint block B_j that extends the candidate checkpoint block B_i such that $j = i + T_1$ (or $j > i + T_1$ if stakeholders were inactive), she solidifies B_i meaning that she disallows any changes to the history from the genesis block until B_i , though B_j can still be discarded as a result of a competing fork.

Since the double-spending safety bound is T_0 , a stakeholder who creates a block can spend the coins only after an intermediate checkpoint block is already solidified, so the costless simulation threat is mitigated (if C_0 is substantial).

This can be seen in the following illustration:



However, this checkpointing mechanism presents two significant problems:

1. New nodes who enter the decentralized network for the first time cannot tell whether the checkpoint blocks that they receive are trustworthy.
2. Due to propagation lag, adversarial stakeholders can collude by preparing an alternative branch of length $T_1 + 1$, and broadcast the competing forks at the same time, thus creating an irreversible split among the network nodes.

The first problem needs to be handled by utilizing a “Web of Trust” type of mechanism that is external to the cryptocurrency system. This means that participants who are unaware of the current state of the system should rely on reputable sources to fetch the blockchain data up to the latest checkpoint.

The second problem should also be resolved manually, meaning that participants who become aware of a network split can decide to instruct their node to switch to the other faction, e.g. if they see that they are in the minority. Note, however, that the second problem becomes increasingly unlikely for larger T_1 values. The exemplary parameters that we proposed in section 2.2 imply that a fork of $T_1 + 1$ blocks represents more than one week of ledger history.

4 Conclusion

It is challenging to design sustainable decentralized cryptocurrency protocols that do not rely on depletion of physical scarce resources for their security maintenance. Our analysis argues that the security of existing such protocols is lacking. We offer novel constructions of pure *Proof of Stake* protocols that avoid depletion of physical scarce resources, and argue that our protocols offer better security than existing protocols. Future work could extend the scope of our analysis to broader strategy spaces.

References

1. BAHACK, L. AND COURTOIS, N. 2014. <http://arxiv.org/abs/1402.1718>.
2. BEN-OR, M. AND LINIAL N. 1990. Collective coin flipping. In Silvio Micali, editor, *randomness and computation*, pages 91–115. Academic Press, New York.
3. BENTOV, I., LEE, C., MIZRAHI, A., AND ROSENFELD, M. 2014. Proof of Activity. In *ACM SIGMETRICS Workshop - NetEcon*. <http://eprint.iacr.org/2014/452>.
4. BOLDYREVA, A. 2003. Efficient threshold signature, multisignature and blind signature schemes based on the gap-Diffie-Hellman-group signature scheme. In *PKC2003*.
5. EYAL, I. 2015. The miner’s dilemma. In *36th IEEE S&P*.
6. EYAL, I. AND SIRER, E. 2014. Majority is not enough. In *Financial Cryptography*.
7. GARAY, J., KIAYIAS, A., AND LEONARDOS, N. 2014. The Bitcoin Backbone Protocol: Analysis and Applications. In *Eurocrypt 2015*.
8. ITAKURA K. AND NAKAMURA K. 1983. A public key cryptosystem suitable for digital multisignatures. *NEC Research & Development*, 71:1-8.
9. KAMP J. AND ZUCKERMAN D. 2007. Deterministic Extractors for Bit-Fixing Sources and Exposure-Resilient Cryptography. In *SICOMP, volume 36*.
10. KROLL, J., DAVEY, I., AND FELTEN, E. 2013. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. In *12th WEIS*.
11. LU, S., OSTROVSKY, R., SAHAI, A., SHACHAM, H., AND WATERS, B. 2006. Sequential aggregate signatures. In *J. Cryptology 26(2): 340-373 (2013)*.
12. NAKAMOTO, S. 2008. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org*.
13. MICALI, S., OHTA K., AND Reyzin, L. 2001. Accountable-subgroup multisignatures (extended abstract). In *Proceedings of CCS 2001, pages 245-254. ACM Press*.
14. POELSTRA, A. 2014. <https://download.wpsoftware.net/bitcoin/pos.pdf>.
15. ROSENFELD, M. 2012. <http://arxiv.org/abs/1402.2009>.
16. USER “CUNICULA”. 2012. <https://bitcointalk.org/index.php?topic=122291>.
17. USER “QUANTUMM...”. 2011. <https://bitcointalk.org/index.php?topic=27787>.