PROOF OF REPUTATION

CYRIL GRUNSPAN AND RICARDO PÉREZ-MARCO

Abstract. We present the new mining protocol Proof-of-Reputation (PoR) for decentralized Proof-of-Work (PoW) blockchains, in particular for Bitcoin. PoR combines the classical PoW with the new ingredient of cryptographic reputation. The same level of security compared to pure PoW can be achieved with a significant energy consumption reduction (of the order of 30%) for the same security level. The proper implementation of a decentralized reputation protocol is suitable with an extra layer of mining security: Certified Mining.

1. Introduction and background

1.1. **Proof-of-Work and decentralization.** The aim of the present article is to enhance the classical Proof-of-Work (PoW) at the base of the decentralized consensus algorithm which runs Bitcoin protocol invented by Satoshi Nakamoto [1]. Although we have in mind the Bitcoin protocol, our analysis is not restricted to it and applies to blockchains based on PoW.

One of the major criticism of Bitcoin is its important energy use. The security of the protocol relies on this energy consumption. Some fundamental reasons, based on the Second Law of Thermodynamics, indicate that there is no viable satisfactory shortcut without energy consumption in order to achieve the same level of security and full decentralization. Other propositions as Proof-of-Stake (PoS) lack of the main security features as Bitcoin, for example an attacker can rebuild the full blockchain, therefore monitorization by a central entity is constantly necessary. So far, no blockchain with PoS does function without supervision as Bitcoin does.

A criterion for a blockchain to be fully decentralized is that no entity or individual is able to stop it. Observe that this entails a level of cybersecurity unseen elsewhere, for example, no hacking can stop it. Also this implies that the network is permisionless and no entity is able to censor transactions. For a PoW based blockchain this is true as long as no one is in control of more than 50% of the available computing power running the PoW algorithm, which, assuming homogeneous hardware efficiency is equivalent to more than 50% of the energy in use.

2010 *Mathematics Subject Classification.* 68M01, 91-00.
Key words and phrases. Bitcoin, blockchain, proof-of-work, selfish mining.

1.2. **Decentralization and reputation.** The energy consumption defends against Sybil attacks of the network, that is, it prevents the network being flooded by dishonest validators (miners). In an open protocol, decentralization requires that anyone is free to join the network. Censorship is incompatible with decentralization. This is the reason that the so-called "private blockchains", where participation is approved by a higher entity, cannot be decentralized blockchains. The simplest solution in order to avoid a Sybil attack would be to weight the voting rights of any participant (node) by some system of reputation. Unfortunately, so far, no decentralized reputation system has been invented. Reputation is a subtle notion difficult to implement algorithmically.

In Nakamoto's PoW proposal, the voting weight of each participant is directly proportional to the work provided. This is possible thanks to the pseudo-random properties of the hash function used in Bitcoin's PoW algorithm. It has the advantage that we don't need any "memory" or knowledge of past behavior of the participant. It has the drawback that no reputation of past well behavior is incorporated. This presents also a security weakness since anyone without any prior credentials that can achieve more than 50% of the energy dedicated to mining, and can take control of the whole network. In some sense, Nakamoto's PoW replaces reputation by actual verifiable work being done in the present.

As was observed in bitcointalk forum, it is interesting to point out that in the early Bitcoin code, Satoshi was trying to implement a reputation system ([2]) based on mining as his comment suggests *"Add atoms to user reviews for coins created"*. More recently, in the context of Proof of Stake, there have been propositions of reputation system as RepuCoin ([3]).

Thinking along these lines, we explore how to go one step further. We want to encode reputation mathematically with a secure cryptographic procedure. The essential nature of reputation is based on proper past behavior. The most common way of acquiring reputation is by past verifiable work. Other ways to acquire reputation is also by endorsement by reputed individuals, but this "second level reputation" is easier to manipulate and we will discard it for now.

The classical PoW is modulated by a parameter, the *difficulty*, which is adjusted so that the validated blocks flow at a constant rate. The *difficulty* is the same for all miners. If we are able to assign some type reputation to miners, it is natural to require for those that are reputed a lower difficulty. This is how we can implement a reputation bonus. This type of reasoning is a standard procedure in everyday dealings. For example, we trust more a dealer that we already know and that has a verifiable past track. The security requirements for engaging in trading with him is lower that with an unknown dealer that may well be a scammer.

The novel idea at the root of the reputation system presented is to have a variable difficulty parameter for miners. In this way we can improve the protocol with a reputation ingredient: We assign a reputation bonus to miners that mined blocks in the past, that is, to anyone that has in the past contributed to the security of the network by providing energy and work. Because of his verifiable past track, we allow this miner, to validate blocks with a lower difficulty, hence with a lower use of energy.

The nature of reputation makes that this reputation bonus should decay with time. Common sense indicates a higher trust in a dealer with recent successful deals that someone else with the same successful deals but from many years ago. In the next section we propose a mathematical implementation of these natural ideas.

2. Proof-of-Reputation.

2.1. **Reputation bonus sequence.** We consider a positive sequence $(\lambda_n)_{n\geq 0}$ with $0 < \lambda_n < 1$, (λ_n) decreasing, and summable with total sum Λ with

$$0 < \Lambda = \sum_{n\geq 0} \lambda_n < 1 \tag{1}$$

The quantity Λ is the total reputation bonus of the network and can be adjusted from 0 to 1 interpolating between a pure PoW and a pure PoR protocols. The quantity λ_n is the reputation bonus earned by validating the block at depth $H - n$ where H is the current blockchain height.

More precisely, for a miner x, we define

$$\lambda_n(x) = \begin{cases} \lambda_n & \text{if } x \text{ mined block } H - n \\ 0 & \text{otherwise} \end{cases}$$

The total bonus reputation of miner x is then

$$\Lambda(x) = \sum_{n\geq 0} \lambda_n(x) \ . \tag{2}$$

Observe that the total reputation bonus of the network is

$$\Lambda = \sum_{x} \Lambda(x) \tag{3}$$

We use the coinbase public payment address as the miner digital identity. In this way the bonus reputation of a miner can be computed from the data in the blockchain. If a miner wants to accumulate bonus reputation he should use always the same payment address. We can easily add the possibility of multiple payment addresses by requesting a proof of ownership of these other addresses by signing simultaneously with the other addresses for which the miner wants to earn reputation bonus. This opens the possibility of selling reputation bonus that we may not want. Also a simpler way of having a reputation market would be to issue a reputation token with each payment in the coinbase transaction, but this is not in the spirit of the proposal presented in this article because we want to prevent anyone with large ressources to launch an immediate 51% attack.

2.2. **Taylored difficulty.** The output of the Proof-of-Work is in the range of SHA256 which is a 256 binary number. The *target value* $0 \leq D \leq 2^{256} - 1$ is the threshold under which the output is successful. The empiral pseudo-randomness property of SHA256, shows that the average number of iterations before finding a solution is $d = 2^{256}/D$ that we define as the difficulty. Thus, for a given hardware, the average energy needed for a validation is proportional to the difficulty $d = 2^{256}/D$.

If miner x has a reputation bonus $\Lambda(x)$ his new taylored difficulty will be

$$d(x) = d(1 - \Lambda(x))$$

and target value

$$D(x) = \frac{D}{1 - \Lambda(x)}$$

Thus, with a reputation bonus $\Lambda(x)$ the miner saves a percentage $\Lambda(x)$ of energy to ensure the same security. Assuming that all miners use their reputation bonus, we can achieve the same level of security saving a percentage Λ of the total energy. We can also argue that for the same energy consumption, we increase the security (measured in energy) by a factor $(1 - \Lambda)^{-1}$.

2.3. **Structure of the reputation bonus sequence.** As explained before, it is natural to request that the reputation bonus decays over time, this is why we want the sequence (λ_n) to be decreasing. It is also natural to request decay uniform over time. This leads to the natural choice of

$$\lambda_n = \lambda_0 e^{-\chi n}$$

where $\chi > 0$ is the decay constant. We can request that the reputation bonus is halved every year. For example, in the case of Bitcoin protocol this leads to a value

$$\chi = \frac{\log 2}{52560} \approx 1.31877.10^{-5}$$

which leads to a decay of 0.19% per day of the bonus reputation.

Then we have

$$\Lambda = \frac{\lambda_0}{1 - e^{-\chi}}$$

If we set a target energy savings of 30%, that is $\Lambda = 0.3$, then for the previous example with Bitcoin we get

$$\lambda_0 \approx 3.956.10^{-6}$$

It also makes sense to implement a reputation bonus sequence (λ_n) that does not decay exponentially initially, but only after some period of time (like a year for example). Several other reasonable choices are possible.

3. Security analysis.

3.1. **51% attack.** We consider a miner x owning a fraction $0 < q < 1$ of the hashrate. In the long run he can expect to earn a reputation bonus of $\Lambda' = q\Lambda$ since he mines on average a proportion q of the blocks. If everybody else uses his reputation bonus, the apparent hashrate will still be q. Then the 51% can still only be performed when $q > 1/2$. But, in the most unfavorable situation where no one else uses its reputation bonus, his apparent hashrate is higher. We have

Proposition 3.1. *Assuming the previous conditions, the apparent hashrate of the only miner using his reputation bonus is* $\frac{h'}{1-\Lambda'} = \frac{h'}{1-q\Lambda}$ *where* h' *is his hashrate.*

Proof. Let $\mathbf{T}'$, resp. $\mathbf{T}$, be the random variable of the time that it takes to miner using his reputation bonus, resp. the rest of the miners, to mine a block. Then we have

$$\mathbb{E}[\mathbf{T}'] = \frac{d(1-\Lambda')}{h'} = \frac{d}{h'/(1-q\Lambda)}$$

$$\mathbb{E}[\mathbf{T}] = \frac{d}{h}$$

where h is the hashrate of the rest of the miners in the network. □

Now we prove:

Proposition 3.2. *Assuming the previous conditions, the miner only miner using his reputation bonus succeeds in a 51% attack when*

$$q > q_0 = \frac{1}{2} - \frac{\sqrt{4+\Lambda^2}-2}{2\Lambda}$$

Proof. The condition to succeed a 51% attack exactly means that it is more likely that the attacker miner mines a block earlier than the rest of the network, i.e.

$$\mathbb{P}[\mathbf{T}' < \mathbf{T}] > 1/2$$

The random variables $\mathbf{T}$ and $\mathbf{T}'$ follow exponential distributions (see for example [4]) with parameters α', resp α,

$$\alpha' = \frac{1}{\mathbb{E}[\mathbf{T}']} = \frac{h'}{d(1-q\Lambda)}$$
$$\alpha = \frac{1}{\mathbb{E}[\mathbf{T}]} = \frac{h}{d}$$

We have

$$\mathbb{P}[\mathbf{T}' < \mathbf{T}] = \frac{\alpha'}{\alpha'+\alpha} = \frac{\frac{h'}{d(1-q\Lambda)}}{\frac{h}{d}+\frac{h'}{d(1-q\Lambda)}} = \frac{h'}{(1-q\Lambda)h+h'} = \frac{\frac{h'}{h+h'}}{1-q\Lambda\frac{h}{h+h'}} = \frac{q}{1-q\Lambda(1-q)}$$

Therefore, the condition $\mathbb{P}[\mathbf{T}' < \mathbf{T}] > 1/2$ becomes

$$q > \frac{1}{2}(1-\Lambda q(1-q))$$

which is equivalent to

$$\Lambda q^2 - (\Lambda+2)q + 1 < 0$$

The discriminant is $\Delta = 1 + \Lambda^2$ and the quadratic polynomial has one root > 1 and another root $0 < q_0 < 1$ given by the formula in the statement. □

For a value of $\Lambda = 0.3$ this means that $q > q_0 = 0.4627\ldots$. This is not significantly lower than 50 %. The attacker must prepare the attack with time ahead in order to gain the 4% reputation bonus. We clearly achieve the same or betterlevel of security. From the view pont of the security of the network, in some sense, we trade pure hashrate for time and work previously performed.

3.2. **Reputation attack.** In the Proof-of-Reputation protocol, we will have some miners that have acquired along their PoW history some reputation. The new miners are free to gain this reputation bonus through their work. As we know, most miners pool together in "pools" in order to mitigate the volatility of their income because a block validation is a rare event for a low hashrate. Thinking about the pool system in place, we can see a dangerous scenario in the PoR protocol: The pool with higher reputation will be more profitable and will attract all the miners. Therefore will quickly reach more than 50% hashrate. This can compromise the security of the network.

As we can see, what is really happening is that miners joining the pool take advantage of the reputation bonus previously earned by the pool in order to lower their mining difficulty. We need another key idea for a proper functioning of a reputation system: We should be able to certify that whoever benefits from any reputation bonus is the same virtual identity that has done the past work that provided the reputation bonus. In the present situation, there is a cryptographic solution to this problem: *Certified Mining*.

3.3. **Certified mining.** We describe *Certified Mining* for a PoW algorithm based on a hash function as the one for Bitcoin. We remind that Bitcoin's Pow consists in formatting a block, and hash (more precisely, double SHA256) its header which contains critical information about the block and the transactions, in particular it contains the Merkle root of the transactions in the body of the block. A variation of the nounce, the extra nounce, or the format of the body of the block, provides new output hashes and the goal is to get one hash below the difficulty threshold. In their regular operations, a mining pool formats the blocks and provide the headers to hash to the miners. They do format the block with a payment address of the coinbase transaction that they do control (this is where the block reward is paid). The solution to avoid the reputation usurpation is simple: Once the block is formatted as it is done classically, we require the header of the block to be signed with the secret key of the payment coinbase address. This signature is appended at the end to the block header. Each time the nonce is changed, a new signature is required. The bulk of the PoW is then the signatures more than the hashes, and this work can only be provided by the entity in control of the payment address. In some sense, the payment public address of the coinbase transaction is the id of the miner (he can also have several addresses that can be used to cumulate reputation bonuses and provide a proof of ownership).

Obviously, for security reasons, the pool cannot share the secret key of the payment address, and in general, only the owner of the payment address can perform the signature. Observe that the signature includes the nonce, extra nonce, etc and the main computational task is to sign the header (the double hash is much faster to do). Hence this modification of the PoW does not allow to subcontract the PoW taking advantage of a reputation bonus earned by someone else.

This procedure seems to make impossible a pooling system, but this is not so. A pooling system is viable where each one contributes and gets a share weighted by its verifiable reputation.

3.4. **Transition to a PoR system.** It is natural to imagine how will be a transition to PoR. Obviously this requires a hardfork of the protocol that changes the mining algorithm to allow a taylored difficulty for different miners. The authors do not advocate for any hardfork of the Bitcoin protocol whose ideal state is converging to ossification as a payment protocol. If such a hardfork is implemented, it needs to be performed with extraordinary caution. The transition to PoR can indeed be realized gradually. For example, the value of Λ could be increased gradually during a large period of time (for example, 10 years) from 0 to its limit value (0.3 in our numerical example). For these conservative values, there is little risk for Bitcoin security.

References

[1] S. Nakamoto. *Bitcoin: a peer-to-peer electronic cash system*, https:\\bitcoin.org, 2008.
[2] Bitcointalk forum. *Weird note by Satoshi on unreleased Bitcoin 2008 version*, https:\\bitcointalk.org\index.php?topic=5381576.0 , 2022.
[3] J. Yu, D. Kozhaya, J. Decouchant, P. Esteves-Verissimo. *Repucoin: Your reputation is your power*, IEEE Transactions on Computers, vol. 68, no. 8, p. 1225-1237, 2019.
[4] C. Grunspan, R. Pérez-Marco. *Double spend races*, International Journal of Theoretical and Applied Finance, Vol. 21, 2018.

Léonard de Vinci, Pôle Univ., Research Center, Paris-La Défense, France

E-mail address: cyril.grunspan@devinci.fr

CNRS, IMJ-PRG, Paris, France

E-mail address: ricardo.perez.marco@gmail.com

Author's Bitcoin Beer Address (ABBA)[1]: 1KrqVxqQFyUY9WuWcR5EHGVvhCS841LPLn

[1] If you find this article useful, you can send some anonymous satoshis to support our research at the pub.