Brief Announcement: Sustainable Blockchains through Proof of eXercise

Ali Shoker*
HASLab, INESC TEC & Minho University
ali.shoker@inesctec.pt

ABSTRACT

Cryptocurrency and blockchain technologies are recently gaining wide adoption since the introduction of Bitcoin, being distributed, authority-free, and secure. Proof of Work (PoW) is at the heart of blockchain's security, asset generation, and maintenance. Although simple and secure, a hash-based PoW like Bitcoin's puzzle is often referred to as "useless", and the used intensive computations are considered "waste" of energy. A myriad of *Proof of "something"* alternatives have been proposed to mitigate energy consumption; however, they either introduced new security threats and limitations, or the "work" remained far from being really "useful". In this work, we introduce *Proof of eXercise* (PoX): a sustainable alternative to PoW where an *eXercise* is a real world matrix-based scientific computation problem. We provide a novel study of the properties of Bitcoin's PoW, the challenges of a more "rational" solution as PoX, and we suggest a comprehensive approach for PoX.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; Cryptography; • **Social and professional topics → Centralization / decentralization**; *Sustainability*; • **Applied computing → Digital cash**; **Physical sciences and engineering**; **Computational biology**; • **Software and its engineering → Distributed systems organizing principles**; • **Theory of computation →** *Theory of database privacy and security*; • **Computer systems organization →** *Peer-to-peer architectures*;

KEYWORDS

Blockchain, Cryptocurrency, Bitcoin, Security, Sustainablity

ACM Reference Format:
Ali Shoker. 2018. Brief Announcement: Sustainable Blockchains through Proof of eXercise. In *PODC '18: ACM Symposium on Principles of Distributed Computing, July 23–27, 2018, Egham, United Kingdom.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3212734.3212781

*The research leading to these results has received funding from the European Union's Horizon 2020 - The EU Framework Programme for Research and Innovation 2014-2020, under grant agreement No. 732505. Project "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

PODC '18, July 23–27, 2018, Egham, United Kingdom

ACM ISBN 978-1-4503-5795-1/18/07.
https://doi.org/10.1145/3212734.3212781

1 INTRODUCTION

Blockchains and cryptocurrenies are increasingly drawing the attention of business, industry, and academia [4, 8, 9, 14, 15, 28]. The concept is based on using cryptographic tamper-proof public ledger, called *blockchain*, to protect the generation and transfer of "digital" money in a fully distributed peer-to-peer (P2P) fashion. The goals of cyrptocurrenices are mainly to avoid central authorities (like banks), reduce transaction delays and fees, and preserve the real value of money by backing the currency with some "work", done through *mining*. At the heart of blockchain, mining maintains the security and correctness of the system and generates (a.k.a., mines) money as a reward for the miner's work, namely, adding new blocks (of *transactions*) to the blockchain, and verifying the protocol's invariants [25]. Being a critical part of these systems, the "work" is made credible through providing a tamper-proof Proof of Work (PoW)—whose properties are discussed further in [2].

Problem. In blockchain protocols that are based on PoW (or similar Proof-of-Something variants), the "work" a miner must do is to solve a cryptographic puzzle: find a random $n \in \mathbb{N}$ such that given the last seen block header B_h, the following inequation holds:

$$H_{B_h}(n) = \text{SHA-256}^2(B_h \mid n) \leq \tau$$

where H is a SHA-256 [30] hashing function that once applied twi to the concatenation of B_h with the nonce n, returns a positive integer not greater than a predefined target τ, known as *difficulty* [29]. This puzzle together with n represent the PoW and live forever in the blockchain (together with the block), allowing for future verifications. Since SHA-256 is random, the best strategy for the solver is to start with an initial n and keep incrementing it with a set-and-test loop until the puzzle is solved. Unfortunately, this is a very computation-hungry process that manifests in very high energy consumption. Recent studies have shown that the annual electricity consumption of Bitcoin system is almost equivalent to that of entire countries like Ireland, Portugal, and Denmark [10, 12, 21]. This raised the voices referring to Bitcoin's hash-based puzzle as "useless" work; whereas, Bitcoin proponents consider this a legitimate price for maintaining the system. We argue that the work can be more *rational* if the puzzle itself is useful, rather than being random.

Current approaches. Three directions are being followed to reduce the "wasted" energy. The first, e.g., *Proof of Stake* (PoS) [18], is based on Game Theory where creating new blocks is based on *coin age*: a function of coin balance and earning time. The proposal is often criticized that coin age accumulates even when the node is not connected to the network, and being non-democratic solution — biased to wealthy peers. Other variants like Delegated Proof-of-Stake [19] and Proof of Stake Velocity [26] tried to address each issue aside, leaving the other open and inducing new limitations or

security threats [27]; whereas, Proof of Activity (PoA) [6] is a hybrid solution of PoS and PoW, where computation is still considered wasted on a useless nonce. The second direction—usually adopted by academics—is to use Byzantine Fault Tolerance approaches in permissioned blockchains; these are rather not scalable in public settings [28]. The third approach, like ours, is to simply replace the puzzle with a more useful real world problem. However, the usefullness of work in current proposals is questionable and do not address a wide range of real interesting problems. For instance, Primecoin [17] suggested finding prime numbers instead of a random useless nonce; Permacoin [20] tried to use have the miners to invest on the system's storage and memory through *Proof of Retrievability*; while PieceWork [24] tried to outsource work like spam deterrence and Denial of Service defense.

Proposed approach. We introduce *Proof of eXercise* (PoX), an approach to rationalize mining in cryptocurrencies — focusing on Bitcoin — through solving a real eXercise: a scientific computation matrix-based problem. The choice behind matrix-based problems is two-fold: (1) matrices have interesting composability properties that help in tuning difficulty, collaborative verification, and *pool-mining* (see later); and (2) matrix-based problems span a wide range of useful real world problems, being a principle abstraction for most scientific computation problems, among them: DNA and RNA sequencing and data comparison [1, 7], protein structure analysis, image comparison, object superposition, surface matching [3, 11], collaborative-filtering recommendation, data mining [16], computational geometry [13], face detection, and many others [16, 23].

In the following, we overview the Proof of eXercise approach, and we present it as a potential promising approach to make PoW-based blockchains sustainable—that requries further research in the future. A more detailed study on the proprties of PoW, the challenges and details of PoX can be found in the extended version [2].

2 PROOF OF EXERCISE (POX)

We propose Proof of eXercise (PoX), an approach to replace the hash-based puzzle with solving a matrix-based scientific computation problems [22, 23]. To identify the challenges of PoX and address them, we need baseline properties to compare against. Since we are unaware of such a comprehensive study, we found it intuitive to first analyze the properties of Bitcoin's PoW first, then address the PoX challenges and potential solutions, and finally present our solution. Given the submission limits, we only convey the latter contribution and we urge the reader to read the first two in the extended paper [2]. For better presentation, we explain the approach through referring the steps in the PoX workflow in Figure 1.

Task proposals. Consider an employer E having a scientific problem, a.k.a., an eXercise X, that requires computing a matrix product. E stores X in a highly available database XDB (step 1), and gets the corresponding credentials and hash digest H(X)—Figure 1, step 2. For simplicity, assume that XDB is an external paid DB service. Then, E creates an eXercise Transaction XT (step 3) that comprises the PoX version, H(X), meta-data about X, e.g., "type:matrix product; Proof of Hardness: OK; dimension: 1 Billion, etc". Then, it deposits a credit (in Bitcoins) for a tolerated period of time after which E can give up (i.e., E is only interested in the solution before that

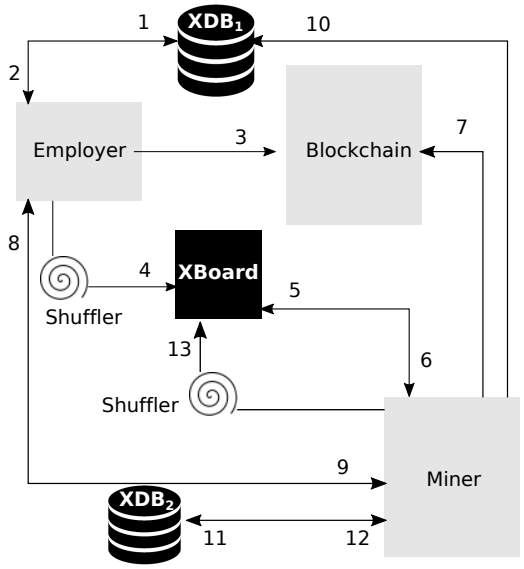

Figure 1: The workflow of PoX without verification. Verification occurs in a similar manner to the steps from 5 through 13 on a verified instance. Refer to Section 2 for more details.

time expires). This guarantees the availability and correctness of X, otherwise the miner may lose (part of) his work. This credit may only be claimed once the eXercise X is solved and verified or the tolerated time has expired. After that, E computes a hash digest H(XT) and submits it to a shuffling service that shuffles H(XT) several times to make it impossible to relate H(XT) to E, and thus prevent collude (step 4). The shuffling service then publishes SH(XT), i.e., the shuffled H(XT), to the eXercise Board (XBoard). Only SH(XT)s that were published for a predefined time may be selected by miners to avoid forks in XBoard — that requires expensive handling as in Bitcoin—since delays are not critical at this level.

eXercise bidding and mining. On the other side, a miner M collects a set of (paid) transactions to be committed and added to the blockchain. To do so, M needs to solve an eXercise chosen from the XBoard and provide a corresponding PoX (step 5). To prevent collude, M gets assigned an eXercise X in a random way, e.g., through matching the hash of block header H(B_h) to the eXercises in XBoard. (Matching can succeed via a pre-defined size of a matching string, or using the hash of H(B_h) and hash digests in XBoard in a similar scheme to Bitcoin's difficulty.) At this stage (step 7), M promises to solve X in the eXercise Transaction XT' through creating a Deal Transaction (DT) that contains: PoX version, SH(XT'), and H(B_h); and then deposits a credit (in Bitcoin's) for a defined period of time — sufficiently long enough — to guarantee its commitment to solve the assigned eXercise. In a similar way to the employer E, the miner M can claim the credit in case the eXercise X is incorrect or became unaccessible. Once the DT is issued, the shuffling service uncovers the onion such that M and E know each other. Consequently, E unveils the meta-data of the eXercise in XT' and gives the credentials of X in the XDB to start working on it (step 9).

PoX Audit. Once the miner M finds Y', i.e., the solution of X, it follows the same process of the eXercise proposing above, making it available for verifiers, called Auditors. In particular, M stores Y' in highly available store, e.g., XDB, and gets a corresponding hash digest H(Y') and access credentials (steps 11 and 12); it creates a corresponding Verify eXercise Transaction (VXT') which is similar to XT, but without requiring a credit this time since M has already deposited a credit through Deal Transaction above. The auditor submits the VXT' to a shuffling service which publishes SH(VXT') — a shuffled version of VXT' to be verified (step 13). Again, this is required to remove any bias in verification. Auditors follow the same bidding procedure as well to choose a random solution Y" to verify, retrieve access details from M and E after the SH(VXT') onion is unshielded, and start auditing Y" through a fast probabilistic verification scheme: an auditor chooses a random number of indices in the matrix to compute; auditing the same eXercise by a sufficient number of auditors will prevent the miner from cheating, as described in more details in the extended version [2].

If the verification *Passed*, the auditor submits a Passed Report through creating an Audit Transaction (AT) that includes the (random) *verification instance* this auditor used for its report, otherwise a Failed Report is submitted. The verification instance is also stored in XDB, and is made available for future audits (within a predefined time frame). Auditors have no interest in submitting false reports since they are at the risk of being caught by other honest auditors in case the same verification instance is repeated. To the contrary, malicious auditors may try submit Failed Reports to compromise the system. This can be prevented by having auditors deposit a credit as a guarantee against false reports — only in the case of submitting Failed Reports.

Committing the block. Once M notices a pre-defined number of Audit Transactions with Passed Reports, it collects the references of all XT, DT, VXT, and AT transactions together with H(X) and H(Y'), and attaches them as a PoX to the block header, that is confirmed by now and can thus safely be added to the blockchain. Finally, all credit deposits are claimed using the PoX of the confirmed block, and the stored matrix in XDB can be removed. (Contrary to Bitcoin, there is no need to verify all the blockchain history since a suffcient number of auditors guarantees the correct PoX correctness with a high probability.) Recall that this verification scheme is important to reduce the overhead of repeated verification of the entire blockchain as well as the data storage and availability costs — which are expensive in the case of PoX.

3 CONCLUSIONS

We introduced Proof of eXercise (PoX): a new proof of work for cryptocurrencies, where the work is a real matrix-based scientific computation problem. Our work shows that the complexity of designing and implementing PoX is much higher than PoW, and therefore, as long as no cheaper alternatives that do not sacrifice the genuine properties of PoW are proposed, it is wise to explore the feasibility of PoX by studying individual scientific computation use-cases, and discussing potential extensions, e.g., as those based on computational complexity [5]. Otherwise, one may opt to stick to cheaper Proof of Stake [6, 18, 19, 26] methods as long as the limitations and constrains are tolerated. Finally, an empirical evaluation that compares the difficulty levels of PoW versus PoX matrices (e.g., dimension, sparseness, etc.) is an interesting future work.

REFERENCES

[1] Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. 2014. Consequences of faster alignment of sequences. In *International Colloquium on Automata, Languages, and Programming*. Springer, 39–51.

[2] Ali Shoker. 2017. Sustainable Blockchain through Proof of eXercise. In *The 16th IEEE International Symposium on Network Computing and Applications (NCA'17)*. IEEE Computer Society.

[3] Helmut Alt and Michael Godau. 1995. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications* 5, 01n02 (1995), 75–91.

[4] Angel.co. [n. d.]. Blockchains Startups. https://angel.co/blockchains. Accessed: 2017-09-15.

[5] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. 2017. Proofs of Useful Work. (2017).

[6] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. 2014. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract] y. *ACM SIGMETRICS Performance Evaluation Review* 42, 3 (2014), 34–37.

[7] Philip Bille. 2005. A survey on tree edit distance and related problems. *Theoretical computer science* 337, 1 (2005), 217–239.

[8] Vitalik Buterin. 2014. A next-generation smart contract and decentralized application platform. *white paper* (2014).

[9] Intel Corporation. [n. d.]. Intel Software Guard Extensions (Intel SGX) SDK. https://software.intel.com/en-us/sgx-sdk. Accessed: 2017-08-29.

[10] Sebastiaan Deetman. [n. d.]. Bitcoin Could Consume as Much Electricity as Denmark by 2020. https://motherboard.vice.com/en_us/article/aek3za/bitcoin-could-consume-as-much-electricity-as-denmark-by-2020. Accessed: 2017-09-05.

[11] Michel Marie Deza and Elena Deza. 2009. Encyclopedia of distances. In *Encyclopedia of Distances*. Springer, 1–583.

[12] Digiconomist. 2018. Bitcoin Energy Consumption. https://digiconomist.net/bitcoin-energy-consumption.

[13] Anka Gajentaan and Mark H Overmars. 1995. On a class of O (n2) problems in computational geometry. *Computational geometry* 5, 3 (1995), 165–185.

[14] IBM. [n. d.]. Hyperledger Fabric. https://www.ibm.com/blockchain/hyperledger.html. Accessed: 2017-09-15.

[15] Bitmain Technologies Inc. [n. d.]. Antminer Hardware. https://shop.bitmain.com/main.htm?lang=en. Accessed: 2017-09-15.

[16] Maja Kabiljo and Aleksandar Ilic. [n. d.]. Recommending items to more than a billion people. https://code.facebook.com/posts/861999383875667/recommending-items-to-more-than-a-billion-people/. Accessed: 2017-08-29.

[17] Sunny King. 2013. Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th* (2013).

[18] Sunny King and Scott Nadal. 2012. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* 19 (2012).

[19] Daniel Larimer. 2014. Delegated Proof-of-Stake (DPOS). *Bitshare whitepaper* (2014). https://bitshares.org/technology/delegated-proof-of-stake-consensus/

[20] Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. 2014. Permacoin: Repurposing bitcoin work for data preservation. In *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 475–490.

[21] Karl J O'Dwyer and David Malone. 2014. Bitcoin mining and its energy footprint. (2014).

[22] University of California. [n. d.]. BOINC projects. http://boinc.berkeley.edu/projects.php. Accessed: 2017-08-29.

[23] University of Zurich. [n. d.]. The Center for Theoretical Astrophysics & Cosmology Program. http://www.ctac.uzh.ch/research/index.html. Accessed: 2017-09-9.

[24] Ittay Eyal Philip Daian, Emin GÃijn Sirer and Ari Juels. 2017. PieceWork: Generalized Outsourcing Control for Proofs of Work. In *BITCOIN Workshop*. Springer.

[25] Bitcoin Project. [n. d.]. Bitcoin documentation. https://bitcoin.org/en/developer-reference. Accessed: 2017-08-29.

[26] Larry Ren. 2014. Proof of stake velocity: Building the social currency of the digital age. (2014).

[27] Florian Tschorsch and Björn Scheuermann. 2016. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials* 18, 3 (2016), 2084–2123.

[28] Marko Vukolić. 2015. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International Workshop on Open Problems in Network Security*. Springer, 112–125.

[29] Bitcoin Wiki. [n. d.]. Difficulty. https://en.bitcoin.it/wiki/Difficulty. Accessed: 2017-09-06.

[30] Bitcoin Wiki. [n. d.]. SHA-256. https://en.bitcoin.it/wiki/SHA-256. Accessed: 2017-09-06.