Author(s): Heinonen, Henri T.; Semenov, Alexander

Title: Recycling Hashes from Reversible Bitcoin Mining to Seed Pseudorandom Number Generators

Year: 2022

Version: Accepted version (Final draft)

Please cite the original version:

Heinonen, H. T., & Semenov, A. (2022). Recycling Hashes from Reversible Bitcoin Mining to Seed Pseudorandom Number Generators. In K. Lee, & L.-J. Zhang (Eds.), Blockchain – ICBC 2021 : 4th International Conference, Held as Part of the Services Conference Federation, SCF 2021, Virtual Event, December 10–14, 2021, Proceedings (pp. 103-117). Springer. Lecture Notes in Computer Science, 12991. https://doi.org/10.1007/978-3-030-96527-3_7

Recycling Hashes from Reversible Bitcoin Mining to Seed Pseudorandom Number Generators*

Henri T. Heinonen[1][0000−0001−5961−3571] and Alexander Semenov[2][0000−0003−2691−4575]

[1] University of Jyväskylä, Jyväskylä, Finland, henri.t.heinonen@student.jyu.fi
[2] University of Florida, Gainesville, FL, USA, asemenov@ufl.edu

Abstract. We analyzed the Bitcoin difficulty data and noticed that the difficulty has been around the level of 10^{13} for three years (H2 2018 - H1 2021). Our calculation showed about 10^{28} hashes have been generated during bitcoin mining around the world for securing the addition of 703,364 blocks to the Bitcoin blockchain. We introduced a concept of Recycling Hashes in the hope to (a) jump-start bespoke silicon (customized silicon) for reversible computing, (b) open up the possibility of Bitcoin's Proof-of-Work to be less energy-consuming in the future, (c) provide scientific value or new services, in the form of entropy pool or random numbers, to Internet users while still achieving the security level of Bitcoin of today, (d) decrease the old mining hardware e-waste by using them to recycle hashes to the entropy pool, and (e) solve the problem of low mining rewards. We found that the bit rates of the current irreversible bitcoin miners are millions of times as high as the existing Internet connections, so it would be difficult to send all the hashes generated in real-time via the Internet. Even if only 0.000000355% of the hashes can be recycled, it would still mean that $355 \cdot 10^{18}$ hashes (355 EH) would have been recycled since the beginning of Bitcoin. Storing all the hashes, so far, would need storage of $2.560 \cdot 10^{30}$ bits, and it is not currently possible to keep all of them. Our simulation of 10,000 bitcoin hashes showed that the occurrences of zeros and ones in bitcoin hashes are almost 50% and 50%, so it is an encouraging finding for seeding the Pseudorandom Number Generators. We also proposed a second coin for the Bitcoin blockchain, an inflationary coin with a different currency unit (BTCi), to motivate the entropy providers to keep the old mining hardware online. The proposed second coin might keep Bitcoin's security model safe in the future when the deflationary bitcoin (BTC or BTCd) block reward is becoming too low.

Keywords: Reversible Computing · Bitcoin Mining · Random Number Generation.

* Supported by Liikesivistysrahasto.

1 Introduction

In this paper, "Bitcoin" (with uppercase B) is the Bitcoin protocol and the Bitcoin network and "bitcoin" (with lowercase b) is the bitcoin money. Bitcoin was introduced in 2008 by Satoshi Nakamoto [3] and the Bitcoin blockchain was started in 2009. Bitcoin mining has been a controversial topic since the mid-2010s. In 2009 and the early 2010s, CPUs (Central Processing Units) were used for bitcoin mining resembling grid computing projects like those utilizing the BOINC (Berkeley Open Infrastructure for Network Computing) platform. In the mid-2010s, bitcoin mining by CPUs was not profitable anymore because there was already bitcoin mining software using the computer graphics card's GPU (Graphics Processing Unit). The next stage in bitcoin mining evolution was the introduction of FPGA (Field-Programmable Gate Array) chips that were even faster at producing SHA256d (double SHA256) (SHA-2 means Secure Hash Algorithm 2) hashes than GPUs. This stage was even shorter than the GPU bitcoin mining stage because some bespoke silicon projects successfully developed and produced ASICs (Application Specific Integrated Circuits) for bitcoin mining.

SHA256d ASICs can only be used to calculate SHA256d hashes; Scrypt ASICs, used for mining litecoin (LTC), can only be used to calculate Scrypt hashes. For comparison, FPGAs can be programmed to do different calculations, and modern GPUs can also be used flexibly. ASICs are not for general computing, but they are swift. The problem with bitcoin ASIC mining is that the chips are still using lots of energy for the calculations. Another problem is that bitcoin ASIC mining devices are "getting old" very fast. It is not profitable to keep old mining hardware online because newer mining hardware will produce hashes at a faster rate and produce more bitcoin income for the hardware owner. Suppose the cost of bitcoin mining is higher than the bitcoin mining revenue. In that case, the only solution is to sell the mining hardware to someone living in an area where electricity is cheaper. Eventually, it is not profitable to use the old hardware for mining anywhere on the planet. The old mining hardware has become "e-waste".

One alternative solution is to use the old hardware to mine some altcoins with the same hash function (SHA256d) Bitcoin is using. One example is namecoin (NMC) that can be mined either alone or merge mined together with bitcoin, but mining altcoins is still not consistently profitable even in the case of merge mining. Merge mining means mining two or more similar kinds of cryptocurrencies simultaneously without sacrificing overall mining performance.

1.1 Bitcoin Mining

Bitcoin mining is a type of lottery game where one competes against other bitcoin miners. The more mining power (the higher the hash rate) one has, the better is the chance to win in this competition. The winner will get permission to add a new block with bitcoin transactions onto the Bitcoin blockchain. The winner will also get a reward that consists of a block reward of several bitcoin (BTC).

The winner will also get the transaction fees (also paid in BTC) added by the users whose transactions were included in the new block.

Difficulty is a measure of how difficult it is to find a hash below a given target. The Bitcoin network has a global block difficulty that is recalculated every 2016 blocks. Because the desired rate of Bitcoin blocks is ten minutes, it would take two weeks to mine 2016 blocks. If it takes less than two weeks for 2016 new blocks, the difficulty will go up; if it takes more than two weeks for 2016 new blocks, the difficulty will go down. [6]

Bitcoin blocks are generally around 1 megabyte in size in 2021. Blocks include transaction data and also headers that contain metadata. There are 80 bytes or 640 bits in the header of a Bitcoin block. The output of the SHA256 (and SHA256d) function is a 256-bit number. This means that the chip to calculate Bitcoin's SHA256d hash function has 640 input wires and 256 output wires.

Mining bitcoin needs lots of electricity. Stoll et al. estimate "the annual electricity consumption of Bitcoin" in November 2018 to be 45.8 TWh and the annual carbon emissions range from 22.0 to 22.9 $MtCO_2$ [36]. For comparison, the use of electricity in Finland totalled 86.1 TWh in 2019 [15], the total energy consumption in Finland in 2019 was 1362 PJ or 378 TWh [9], and the total emissions of carbon dioxide (CO2 eq.) in Finland in 2020 was 48.3 million tonnes [5]. According to the Galaxy Digital Mining report from May 2021 [12], Bitcoin consumed 113.89 TWh of electricity annually, the gold industry used about 240.61 TWh of energy annually, and the banking industry consumed 263.72 TWh of energy annually. They compare Bitcoin's electricity usage to the global annual energy supply (1,458.2 times that of the Bitcoin network), the global annual electricity generation (234.7 times that of the Bitcoin network), the amount of electricity lost in transmission and distribution each year (19.4 times that of the Bitcoin network), and the energy footprint of "always-on" devices in American households (12.1 times that of the Bitcoin network). It is also useful to compare the bitcoin mining electricity usage to the electricity and energy usages of other IT industries' activities. PC gaming used about 75 TWh of electricity in 2012 according to Mills et al. [34] Facebook's global electricity consumption was 5.14 TWh in 2019 according to Alves [8]. The energy consumption of Google (Alphabet) was 12.7 TWh in 2019, according to Jaganmohan [1]. According to Alden [4], Bitcoin's energy usage is not a problem because the mining uses less than 0.1% of global energy and because a sizable portion of the energy used for mining would be otherwise stranded and wasted.

Bitcoin mining is based on a "Proof-of-Work" (PoW) mechanism, the idea that a miner needs to spend a sufficient amount of work to receive the compensation. In Bitcoin, it is implemented based on the principle that it is easy to validate the correctness of a cryptographic SHA256d hash given the input and the resulting hash, but it is very hard (or impossible) to find the input for the hash function from the particular output. Generally, to find an input value for a hash function given its output, one should brute force possible inputs. During the bitcoin mining process, miners compete in finding the *nonce*, a value that is along with details of new transactions and a link to the previous block, a

part of the input to the SHA256d functions. The goal is to find such a nonce that the number of leading zeros in the output would be greater than a certain threshold, set by the difficulty. The more leading zeros should be at the beginning of the output, the harder it is to find a suitable *nonce* value. By finding the nonce, new transactions are added into the blockchain, and modifications of the transactions in this block would require finding another nonce in the current and potential subsequent blocks. Thus, the bitcoin mining process consists of repeated calculations of SHA256d hashes and checking if they suit the difficulty constraint.

1.2 Reversible Computing

Almost all of the computing in the world today (including bitcoin mining) is irreversible. From the chip's output, the final state $f(x)$, it is difficult or impossible to figure out the intermediate states and the initial state x. Reversible computing is a computational model where the computational process can be reversed in time, i.e., its previous states can be reconstructed from its subsequent states. For example, specific inputs of logical exclusive OR (XOR) cannot be obtained from its output, as multiple different inputs may correspond to the output; however, the input of NOT operation can be determined based on its output. According to Frank [14], reversible computing refers to computing in a way that preserves signal energies and reuses them over multiple digital operations. Reversible computing focuses on achieving far greater energy efficiency and practical performance for all digital computing, rather than quantum speedups on relatively few specialized applications.

In 1961 Rolf Landauer [31] noticed that logically irreversible gate will dissipate heat to its environment according to the equation

$$E = k_{\mathrm{B}} T \ln(2). \tag{1}$$

In Equation (1), k_{B} is the Boltzmann constant, T is the temperature of the environment in kelvins, and $\ln(2)$ is the natural logarithm of 2.

With reversible computing, it would be possible to *uncompute* the final state $f(x)$ and go back all the way to the initial state x. By not wasting any information, reversible computing could be highly energy-efficient. Making computing reversible could reduce the excess generation of waste heat. Quantum computing is closely related to reversible computing. Frank et al. [24] note that (a) Landauer's Principle sets a strict lower bound on entropy generation in traditional non-reversible architectures for deterministic computing machines; and (b) reversible computing can potentially circumvent the Landauer limit with the potential of allowing the efficiency of future digital computing to improve indefinitely.

1.3 Generating Pseudorandom Numbers

Random numbers in classical computing systems are generally pseudorandom numbers because it is impossible to get truly random numbers from computers

considered deterministic. The big difference is quantum computing that makes true random number generation possible. For example, Heinonen [25] shows a simple example of how to generate a quantum program that generates true random numbers.

Here we consider classical computing systems, so we concentrate on the PRNGs (Pseudorandom Number Generators). There are PRNGs such as Blum Blum Shub [21], Yarrow [29], and Fortuna [22]. Fortuna is a modern and cryptographically secure PRNG. It is a family of secure PRNGs, and they consist of the following parts: (a) the generator, which once seeded will produce pseudorandom data; (b) the entropy accumulator, which collects random data from various sources and reseeds the generator when possible; (c) the seed file, which stores entropy for the computer to start generating random numbers after rebooting.

1.4 Literature review

We know from Stoll et al. [36] that bitcoin mining uses lots of energy and has a considerable carbon footprint. de Vries et al. [40] found that bitcoin mining generates lots of hardware waste or *e-waste*: 30.7 metric kilotons annually as of May 2021. de Vries [39] estimated mining equipment to become obsolete in roughly 1.5 years.

It is exciting that reversible computing is not a new invention, but it is still not used as of writing this article. Bennett [19] found already in 1973 that every classical computation can be turned into reversible form. Toffoli [38] invented a universal reversible logic gate in 1980. According to Frank [23], reversible computing could be from 1000 to 100,000 as cost-effective as irreversible computing in the 2050s. The IBM Q Experience quantum computing documentation has an excellent introduction to reversible computing [7].

We also know various consensus methods that have the potential to replace the energy-consuming Proof-of-Work consensus methods. For example, Ethereum developers are trying to replace Ethereum's Proof-of-Work with Proof-of-Stake (PoS). We know projects like Gridcoin [10], and Primecoin [30] do valuable science while securing the blockchains with their consensus methods. Bizzaro et al. [20] introduce Proof-of-Evolution (PoE) that keeps the security features of Proof-of-Work, and uses part of the mining computations for the execution of genetic algorithms (GAs). Miller et al. [33] try to repurpose Bitcoin work for data preservation. Manthey et al. [32] try to replace brute force mining algorithm with solving Boolean satisfiability problem (SAT).

Bitcoin's transaction fees are too low to motivate bitcoin miners, according to Kaşkaloğlu [28] and Cussen [17]. According to Alden [4], the Bitcoin network continues to be more energy-efficient each year due to the declining block rewards.

According to Taylor [37], bitcoin ASIC mining is proof that bespoke silicon (customized silicon) can be developed in small volumes. These devices outperform general-purpose SoCs developed by major multi-billion dollar companies.

Ferguson et al. [22] note that backups and virtual machines cause problems when reseeding PRNGs. The problem is that PRNG that loads the seed file from backups will be reseeded from the very same seed file. Until the accumulator has collected enough entropy, the PRNG will produce the same output after two reboots. They claim that there is no direct defense against this kind of attack.

Wang et al. [41] present RandChain, a decentralized random beacon protocol designed to provide continuous randomness at regular intervals.

According to the literature research, we do not have solid answers to the following questions.

1. How to secure the Bitcoin blockchain without a huge carbon footprint and lots of mining hardware e-waste? There are consensus methods like Proof-of-Stake, but they are not ready to replace Proof-of-Work yet.
2. The information in reversible computing needs to be stored somewhere. Where and how will it be stored? Will it be stored locally or globally?
3. There seems to be not enough incentive to build reversible computers. How to stimulate the development of reversible computing hardware and software?
4. When there is not enough entropy available, how to seed PRNGs without using the same seed file during the computer startup process?
5. People who do not use bitcoin tend to state that bitcoin is not valuable. How to make Bitcoin more valuable and justified even for those who do not want to use the bitcoin cryptocurrency itself? One method to provide new value to the system is to solve science problems while securing the blockchain. There are inventions like Proof-of-Evolution, Primecoin, and Permacoin, but Bitcoin is not using their methods.

Research Question Our research question is: How to change bitcoin mining to use potentially less energy and do something valuable besides securing the Bitcoin blockchain?

1.5 Recycling Hashes from Reversible Bitcoin Mining to Seed Pseudorandom Number Generators

We try to answer our Research Question by introducing Recycling Hashes from Reversible Bitcoin Mining to Seed Pseudorandom Number Generators. Using reversible computing for bitcoin mining has been discussed on the Bitcoin Forum [13]. Seeding PRNGs with random data is a familiar concept, and methods like LavaRand use digitalized fresh images of lava lamps to seed PRNGs.

What kind of a chip would mine bitcoin using reversible computing? The exact number of input and output wires for the R-SHA256d chip is unknown because reversible computing architectures are still in the early stages. There will probably be more input and, especially, output wires for the reversible SHA256d chip than for the irreversible SHA256d chip.

Is not it impossible to reverse a secure hash function? Reversible computing is not breaking the secure hash functions (including SHA256). It will only echo the input wires x to output wires x, calculate the final state $f(x)$ and generate

some garbage data, intermediate states $g(x)$, from clean scratch memory 000... (L zeros). All it does is mapping x, 0^L to x, $g(x)$, and $f(x)$. It is impossible to use the output from SHA256 (or SHA256d) in R-SHA256 (or R-SHA256d) to figure out the input. The output of SHA256 (and SHA256d) is missing the x and $g(x)$ information that would be needed for going back to the initial state x.

The idea of using reversible bitcoin mining to generate random numbers did not come from reversible computing but from the need to find some usage of the billions of hashes generated during the mining process. There is the famous LavaRand method [35] to generate random numbers by taking digital pictures of lava lamps, converting the information to binary numbers, applying a cryptographic hash function, obtaining seed from the hash function, and feeding that seed to the PRNG. Our idea was to take the otherwise wasted hashes of bitcoin mining and feed them to the Bitcoin network users to seed their PRNGs. This idea was getting more justified in reversible computing. Erasing information means generating waste heat. The erasing of information can be avoided if the information is copied to *a clean auxiliary register* before uncomputing the solution $f(x)$ [7].

What if most or at least some of the otherwise wasted hashes of mining could be recycled somehow? Could they be stored onto the blockchain or sent securely to the Bitcoin network users so they can seed their PRNGs? The peer-to-peer network of Bitcoin (or the blockchain itself) could act as the auxiliary register to record the information before it gets uncomputed (and erased). The Fortuna PRNG has a problem with the seed files when using virtual machines or backups because the same seed file will be used. Our solution of using fresh seeds from the blockchain network's entropy pool could solve this problem. It will need an Internet connection to get fresh seeds from the blockchain network.

2 Methods

Bitcoin difficulty is a measure of the mining power available securing the Bitcoin blockchain. The Bitcoin difficulty changes every 2016 blocks (two weeks if there are 10 minutes between each block) to correspond to the changes in total hash rate. We got the Bitcoin difficulty data from Blockchain.com website [11] and a bitcoin miner's technical specs from the producer's website [2].

The Bitcoin network's total hash rate measures the number of hashes the miners worldwide are generating when mining bitcoin in one second. We got the Bitcoin network's total hash rate data from the Blockchain.com website [16].

We simulated mining Bitcoin's Genesis block with Python code to generate 10,000 hashes until the mining ended with finding the correct hash. We stored the hashes as binary numbers into a file sample.bin. The file contained 2,560,000 binary numbers (zeros and ones). We run the Fourmilab's Pseudorandom Number Sequence Test Program, ent, with the following command:

```
ent -c sample.bin > sample.bak
```

Table 1. Table showing the bit rate of the miner divided by the upload speed of the Internet connection. The slower speeds (Gbit/s) are the Internet upload speeds and the faster speeds (Pbit/s) are the bit rates of the miners.

	2.816 Pbit/s	28.160 Pbit/s	281.600 Pbit/s
0.1 Gbit/s	281,600,000	2,816,000,000	28,160,000,000
1.0 Gbit/s	28,160,000	*281,600,000*	2,816,000,000
10 Gbit/s	2,816,000	28,160,000	281,600,000

3 Results

In this section we introduce the results: difficulty and hash rate of Bitcoin over time, the total number of hashes generated in bitcoin mining, and our small pseudorandom number sequence test to check the occurrences of ones and zeros in the set of 10,000 hashes, the entropy of the data set and some other statistics generated by the ent program.

3.1 Difficulty, hash rate, and total number of hashes

We plotted the Bitcoin difficulty in function of time in Figure 1 and the Bitcoin network's total hash rate in function of time in Figure 2. We calculated the integral of the Bitcoin network's total hash rate (hashes per second) data, $H(t)$, over the time period of early 2009 to this date by using Python SciPy's trapezoid function and got the result of

$$\int_{t=T(\text{2009-01-02 23:00:00})}^{T(\text{2021-09-30 00:00:00})} H(t)\,\mathrm{d}t = 1.059466790224828 \cdot 10^{28} \text{ hashes} \approx 10^{28} \text{ hashes}. \tag{2}$$

The number of hashes in Equation (2) means that storing all of them would need storage of $2.560 \cdot 10^{30}$ bits.

According to [2] Antminer S19 Pro has a hash rate of 110 TH/s, so it can generate $110 \cdot 10^{12}$ SHA256d hashes per second. One SHA256d hash has 256 bits, so the bit rate of the miner is $28.16 \cdot 10^{15}$ bit/s or 28.160 Pbit/s. We calculated various different upload speeds and bitcoin miner's bit rates in Table 1.

3.2 Pseudorandom number sequence test

We used the program called *ent* to test our sequence of 10,000 hashes stored in a file that contained 2,560,000 zeros and ones. Table 2 shows the fractions of ones and zeros in our file with 10,000 simulated bitcoin hashes. The test results from the ent program were stored in a file sample.bak.

The entropy of the data set was 1.000000 bits per byte according to the ent program. Optimum compression would reduce the size of the 2560000-byte file by 87 percent. Chi-square distribution for 2560000 samples was 325120003.70 and

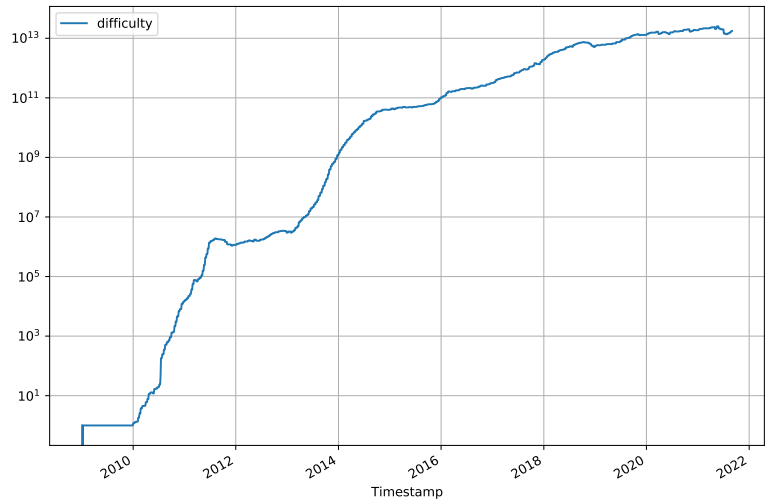

Fig. 1. The difficulty of Bitcoin during the years.

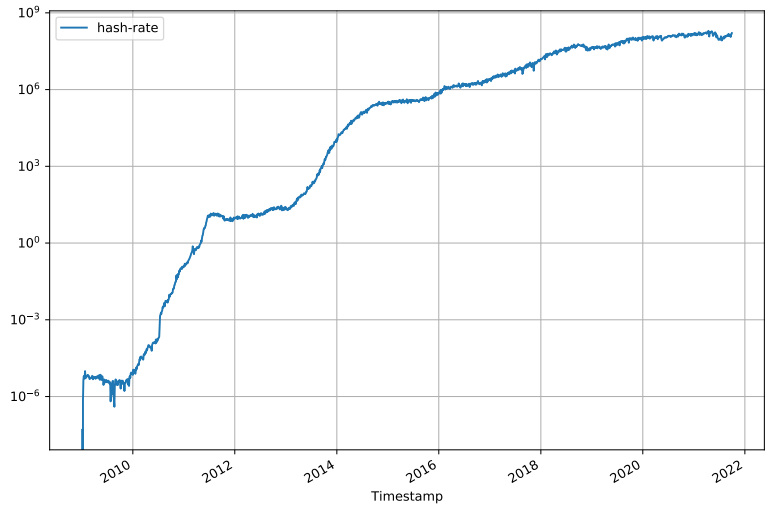

Fig. 2. The total hash rate (H/s) of Bitcoin network during the years.

Table 2. Table showing the ASCII values of the characters, their occurrences and fractions of the whole data set.

ASCII value	Character	Occurrences	Fraction
48	0	1280136	0.500053
49	1	1279864	0.499947
Total		2560000	1.000000

randomly would exceed this value less than 0.01 percent of the time. The arithmetic mean value of the data bytes was 48.4999 (127.5 = random). Monte Carlo value for Pi was 4.000000000 (error 27.32 percent). Serial correlation coefficient was 0.000944 (totally uncorrelated = 0.0).

4 Discussion

In this section, we discuss the huge number of hashes generated by bitcoin mining, the speed of Internet connections, our proposal of a two-coin model to incentivize the usage of bitcoin miners that would not be profitable with the current one-coin model of deflationary bitcoin (BTCd). We also discuss further research.

4.1 The number of hashes and the speed of Internet connections

According to our calculation in Equation (2), the total number of hashes generated by bitcoin mining since the beginning of Bitcoin is 10^{28} hashes. When writing this article, only 703,364 of those hashes have been used to add a new block onto the Bitcoin blockchain.

The Antminer S19 Pro miner will generate $281.6 \cdot 10^6$ as many hashes as it is possible to transfer through the Internet connection as seen in the middle of Table 1. Most of these hashes will probably be erased, so they will contribute to heat generation. What will be the bit rate of a realistic reversible bitcoin miner? We cannot be sure because our understanding of reversible computing principles is minimal.

It was stated in IBM's documentation [7] that one would never use the method described in the documentation for reversible computations since it requires too large a scratch memory. According to the documentation, some proposed optimization methods exist to uncompute partial results and reuse scratch memory bits.

A realistic Internet connection in the consumer market is 100 Mbit/s and small data centers could have a connection of 1 Gbit/s. If a bitcoin mining data center has ten Antminer S19 Pro miners and a 1 Gbit/s Internet connection, then the bit rate of the miners is 2,816,000,000 times the speed of the Internet connection. This would mean that

$$\frac{281,600,000\,\mathrm{Gbit/s} - 1\,\mathrm{Gbit/s}}{281,600,000\,\mathrm{Gbit/s}} \cdot 100\% = 99.9999996448863636\ldots\%$$

of the generated hashes will be destroyed and only $0.0000003551136\ldots\%$ of the generated hashes will be recycled. Even if only 0.000000355% of the hashes can be recycled, it would still mean that $0.0000000355 \cdot 10^{28} = 355 \cdot 10^{18}$ hashes (355 EH) would have been recycled since the beginning of Bitcoin!

Storing all the hashes would mean storing $2.560 \cdot 10^{30}$ bits, but it is not feasible at the moment. According to Barnett [18], in 2016, the whole Internet traffic generated one zettabyte or about $8 \cdot 10^{21}$ bits of information.

Our simulation of 10,000 hashes showed, in Table 2, that the occurrences of zeros and ones in bitcoin hashes are almost 50% and 50%, so it is probably an encouraging finding for seeding the PRNGs.

4.2 Two-coin model

In this work, we proposed a second coin for the Bitcoin blockchain, an inflationary coin with a different currency unit (BTCi), to motivate the entropy providers to keep the old mining hardware online. The second coin might keep Bitcoin's security model safe in the future when the deflationary bitcoin (BTC or XBT or BTCd) block reward is becoming too low. The deflationary bitcoin coin (BTCd) comes with the famous cap of 21 million coins in total, but the inflationary bitcoin coin (BTCi) does not necessarily have any cap at all.

Having inflationary coins in the same blockchain ecosystem could also provide a solution to the problem of coin hoarding, holding, or "hodling". Inflationary coins would motivate (inflationary) bitcoin users to spend their money because inflation would eventually decrease the second coin's monetary value.

There are at least two different reasons why inflationary coin would solve the problem of "low mining rewards": (a) The inflationary bitcoin coin, which is given as a reward to the entropy providers (especially to the old mining hardware users), would probably motivate to keep on mining because the BTCi coin would have a monetary value even if it was not as expensive as the BTCd coin; and (b) the inflationary coin would probably raise the number of transactions in a block because the inflationary nature of BTCi coin would make people to use it more frequently than they use the deflationary BTCd coin. The more transactions are included in a block, the higher are the total transaction fees per block.

4.3 Further research

Further research would include using real bitcoin miners to generate seeds for PRNGs. It would be interesting to know if this could become a practical way to generate good quality random numbers in the future.

There needs to be more research on reversible computing principles. It would be interesting to know if quantum computing groups could also do more research on reversible (classical) computing because reversible computing and quantum computing are closely related.

There must also be more research on many-coin cryptoeconomies. How would the bitcoin economy change if a hard fork introduces a second coin into the

blockchain, for example, the inflationary BTCi coin? In the Ethereum ecosystem, the ether coin (ETH) and thousands of smart contract tokens are mainly running without any significant issues. Heinonen et al. [27] found some differences in behaviour between the ERC-20 (ERC means Ethereum Request for Comments) tokens and stockmarket. Heinonen [26] introduced the two-money cryptoeconomy of money and antimoney.

5 Conclusion

Our research question was: How to change bitcoin mining to use potentially less energy and do something valuable besides securing the Bitcoin blockchain?

Assuming the difficulty of Bitcoin will stay around 10^{13}, we found out that even with a reversible bitcoin miner, lots of heat will probably be generated because most of the generated hashes (information) will be erased in a way or another. The good side is that recycling hashes from bitcoin mining to PRNGs provides new value to the Bitcoin network. This entropy pool service could be available even for those who do not do bitcoin mining nor use bitcoin cryptocurrency nor the Bitcoin blockchain at all.

There may be breakthroughs in Internet connection speeds, mass storage, and reversible computing principles to overcome these issues. Still, it is challenging not to waste any energy during blockchain operations. Even if there are no breakthroughs in these technologies, our finding that

$$\begin{aligned}\text{hashes accepted (current block height)} &\lll \text{hashes potentially recycled}\\ &\ll \text{hashes generated}\end{aligned}$$

still motivates to pursue hash recycling.

Our proposal could be a solution for the problem of bitcoin mining hardware e-waste. One could use one's old (reversible/irreversible) ASIC bitcoin miner to generate hashes for the Bitcoin entropy pool even though the miner device is too old to create profitable deflationary bitcoin coins (BTCd) anymore. The incentive for mining with old hardware could come from the inflationary bitcoin coins (BTCi).

We hope that our concept of Recycling Hashes from Reversible Bitcoin Mining to Seed Pseudorandom Number Generators could:

1. Jump-start bespoke silicon for reversible computing.
2. Open up the possibility of Bitcoin's Proof-of-Work to be less energy-consuming in the future.
3. Provide scientific value or new services, in the form of entropy pool or random numbers, to Internet users while still achieving the security level of Bitcoin of today.
4. Decrease the old mining hardware e-waste by using them to recycle hashes to the entropy pool.
5. Solve the problem of low mining rewards.

Acknowledgements

We thank Professor Pekka Neittaanmäki and Professor Timo Hämäläinen for discussions and feedback. Henri thanks Liikesivistysrahasto (200092) for support.

References

1. Alphabet (google): energy consumption 2019 — statista. https://web.archive.org/web/20211029095928/https://www.statista.com/statistics/788540/energy-consumption-of-google/, accessed: 2021-11-08
2. Antminer s19 pro - the future of mining. https://web.archive.org/web/20210906102302/https://shop.bitmain.com/release/AntminerS19Pro/overview, accessed: 2021-09-06
3. Bitcoin: A peer-to-peer electronic cash system. https://web.archive.org/web/20211103223918/https://bitcoin.org/bitcoin.pdf, accessed: 2021-11-04
4. Bitcoin's energy usage isn't a problem. here's why. https://web.archive.org/web/20211103232331/https://www.lynalden.com/bitcoin-energy/, accessed: 2021-11-08
5. Carbon dioxide emissions - motiva. https://web.archive.org/web/20201030003703/https://www.motiva.fi/en/solutions/energy_use_in_finland/carbon_dioxide_emissions, accessed: 2021-10-26
6. Difficulty - bitcoin wiki. https://web.archive.org/web/20210813113701/https://en.bitcoin.it/wiki/Difficulty, accessed: 2021-09-29
7. Docs and resources - ibm quantum experience - shor's algorithm. https://web.archive.org/web/20201101072900/https://quantum-computing.ibm.com/docs/iqx/guide/shors-algorithm, accessed: 2021-09-06
8. Facebook electricity usage globally 2019 — statista. https://web.archive.org/web/20210818230043/https://www.statista.com/statistics/580087/energy-use-of-facebook/, accessed: 2021-11-08
9. Final consumption of energy - motiva. https://web.archive.org/web/20211026171442/https://www.motiva.fi/en/solutions/energy_use_in_finland/final_consumption_of_energy, accessed: 2021-10-26
10. Gridcoin white paper - the computation power of a blockchain driving science and data analysis. https://web.archive.org/web/20210815003224/https://gridcoin.us/assets/docs/whitepaper.pdf, accessed: 2021-11-04
11. Network difficulty - a relative measure of how difficult it is to mine a new block for the blockchain. https://www.blockchain.com/charts/difficulty, accessed: 2021-09-03
12. On bitcoin's energy consumption: A quantitative approach to a subjective question. https://web.archive.org/web/20211108150128/https://docsend.com/view/adwmdeeyfvqwecj2, accessed: 2021-11-08
13. Re: Theoretical minimum of logic operations to perform double iterated sha256? https://web.archive.org/web/20210906102310/https://bitcointalk.org/index.php?topic=1029536.msg11145144, accessed: 2021-09-06
14. Reversible computing: The only future for general digital computing. https://web.archive.org/web/20210401031527/https://cfwebprod.sandia.gov/cfdocs/CompResearch/docs/LPS21-talk-v5.pdf, accessed: 2021-10-01

15. Statistics finland - energy supply and consumption. https://web.archive.org/web/20210414035155/https://www.stat.fi/til/ehk/2019/ehk_2019_2020-12-21_tie_001_en.html, accessed: 2021-11-08
16. Total hash rate (th/s) - the estimated number of terahashes per second the bitcoin network is performing in the last 24 hours. https://www.blockchain.com/charts/hash-rate, accessed: 2021-10-03
17. Turning off bitcoin's inflation funded security model - wishful thinking? https://web.archive.org/web/20211012055718/https://www.onionfutures.com/turning-off-bitcoins-inflation, accessed: 2021-10-26
18. The zettabyte era officially begins (how much is that?). https://web.archive.org/web/20210813122554/https://blogs.cisco.com/sp/the-zettabyte-era-officially-begins-how-much-is-that, accessed: 2021-10-04
19. Bennett, C.H.: Logical reversibility of computation. IBM Journal of Research and Development **17**(6), 525–532 (Nov 1973). https://doi.org/10.1147/rd.176.0525
20. Bizzaro, F., Conti, M., Pini, M.S.: Proof of evolution: leveraging blockchain mining for a cooperative execution of genetic algorithms. In: 2020 IEEE International Conference on Blockchain (Blockchain). pp. 450–455. IEEE (2020)
21. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on computing **15**(2), 364–383 (1986)
22. Ferguson, N., Schneier, B., Kohno, T.: Cryptography engineering: design principles and practical applications. John Wiley & Sons (2011)
23. Frank, M.P.: Nanocomputer systems engineering. CRC Press (2006)
24. Frank, M.P., Shukla, K.: Quantum foundations of classical reversible computing. Entropy **23**(6), 701 (2021)
25. Heinonen, H.: Katsaus kvanttilaskentateknologiaan ja sen sovelluksiin. Informaatioteknologian tiedekunnan julkaisuja (88) (2021)
26. Heinonen, H.T.: On creation of a stablecoin based on the morini's scheme of inv&sav wallets and antimoney (2021), accepted to IEEE Workshop on Blockchain Security, Application, and Performance (BSAP-2021)
27. Heinonen, H.T., Semenov, A., Boginski, V.: Collective behavior of price changes of erc-20 tokens. In: International Conference on Computational Data and Social Networks. pp. 487–498. Springer (2020)
28. Kaskaloglu, K.: Near zero bitcoin transaction fees cannot last forever (2014)
29. Kelsey, J., Schneier, B., Ferguson, N.: Yarrow-160: Notes on the design and analysis of the yarrow cryptographic pseudorandom number generator. In: International Workshop on Selected Areas in Cryptography. pp. 13–33. Springer (1999)
30. King, S.: Primecoin: Cryptocurrency with prime number proof-of-work. July 7th **1**(6) (2013)
31. Landauer, R.: Irreversibility and heat generation in the computing process. IBM journal of research and development **5**(3), 183–191 (1961)
32. Manthey, N., Heusser, J.: Satcoin–bitcoin mining via sat. SAT COMPETITION 2018 p. 67 (2018)
33. Miller, A., Juels, A., Shi, E., Parno, B., Katz, J.: Permacoin: Repurposing bitcoin work for data preservation. In: 2014 IEEE Symposium on Security and Privacy. pp. 475–490. IEEE (2014)
34. Mills, N., Mills, E.: Taming the energy use of gaming computers. Energy Efficiency **9**(2), 321–338 (2016)
35. Noll, L.C., Mende, R.G., Sisodiya, S.: Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system (Mar 24 1998), uS Patent 5,732,138

36. Stoll, C., Klaaßen, L., Gallersdörfer, U.: The carbon footprint of bitcoin. Joule **3**(7), 1647–1661 (2019)
37. Taylor, M.B.: Bitcoin and the age of bespoke silicon. In: 2013 international conference on compilers, architecture and synthesis for embedded systems (CASES). pp. 1–10. IEEE (2013)
38. Toffoli, T.: Reversible computing. In: International Colloquium on Automata, Languages, and Programming. pp. 632–644. Springer (1980)
39. de Vries, A.: Renewable energy will not solve bitcoin's sustainability problem. Joule **3**(4), 893–898 (2019)
40. de Vries, A., Stoll, C.: Bitcoin's growing e-waste problem. Resources, Conservation and Recycling **175**, 105901 (2021)
41. Wang, G., Nixon, M.: Randchain: Practical scalable decentralized randomness attested by blockchain. In: 2020 IEEE International Conference on Blockchain (Blockchain). pp. 442–449. IEEE (2020)