NEBULA-A CRYPTOCURRENCY WHOSE PROOF-OF-WORK PROBLEM, R5, WILL BUILD THE REVERSIBLE COMPUTER

JOSEPH VAN NAME

Abstract. By Landauer's principle, reversible computers are potentially many times more energy efficient than the maximum possible efficiency of an irreversible computer. The randomizing function for the proof-of-work problem for the cryptocurrency Nebula is designed to be computed by reversible circuits. This cryptocurrency will therefore prompt the development of energy efficient reversible computers in order to compute the randomizing function to mine Nebula in a profitable way.

1. Introduction

The most prominent issue with cryptocurrencies is that the POW problems require a large investment in computing machinery and energy but these problems do not provide anything of value besides securing the blockchain. Some have attempted to resolve this issue by replacing the computationally intensive proof-of-work (Abbreviated POW) problem with proof-of-stake which does require intensive computation (unlike POW problems, proof-of-stake cannot solve the problem of how to distribute newly minted currencies). Others have attempted to select a POW problem where the solution or the process of obtaining the solution advances science or mathematics in some way (for example, the POW problems for Primecoin and Gapcoin amount to finding peculiar prime numbers). The value however of the solution of these POW problems for Primecoin and Gapcoin is debated since no new theorems, conjectures, or platforms for cryptosystems have been obtained through mining these cryptocurrencies. Since POW problems for cryptocurrencies must satisfy some stringent requirements, so far there are no cryptocurrencies with a POW problem which has undisputed real-world applications.

Cryptocurrencies need to implement useful POW problems to secure a strong public image. If cryptocurrencies do not have a strong public image, then various organizations will be likely to attack the security of the cryptocurrencies, and governments will pass laws restricting cryptocurrencies.[1] As the market for cryptocurrencies grows, environmental advocates will more intensely condemn and lobby against cryptocurrencies for having wasteful POW problems unless their POW problems have some other practical use besides achieving decentralization.[2] Therefore, the best way for a cryptocurrency to maintain a strong public image and be in a good

[1]Some nations such as Ecuador and Bolivia have already banned or have placed severe restrictions on cryptocurrencies.

[2]Environmental groups currently have made little effort to advocate against cryptocurrency POW problems. For example, Greenpeace has accepted Bitcoin donations since 2014 and continues to accept Bitcoin donations. Furthermore, Greenpeace currently views cryptocurrencies favorably.

standing with the world's governments will be to select a suitable POW problem that advances science, technology, or mathematics.

Landauer's principle gives a theoretical limit to the efficiency of conventional computers. However, Landauer's principle only applies to irreversible operations, and reversible operations are not restricted by Landauer's principle. Reversible computers, which only implement reversible operations, are therefore potentially many times more efficient than conventional computers could ever become. Since reversible computers are potentially more efficient than conventional computers, reversible theoretically computers produce much less heat than conventional computers, and therefore reversible computers can potentially run at much higher speeds than conventional computers. Unfortunately, all of today's computers are irreversible, and there are currently no reversible computers which are more efficient than a conventional computer which simply simulates reversible operations. Any function which can be computed by a conventional computer could also be computed by a reversible computer. However, today there is little demand for and little motivation to develop reversible computers since reversible computers typically need to take more steps to compute a function than is needed for an conventional computer to compute the same function. The increased complexity of reversible computation stems from the fact that purely reversible computers are not allowed to delete any information since deleting information is an irreversible process. Reversible computers are not even necessary today since the efficiency of today's computers is still far below Landauer's limit. On the other hand, in the future, since reversible computation is not subject to Landauer's limit, computational machinery will consist mainly of reversible components since the efficiency gained by using reversible computation will eventually compensate for the need for the additional complexity which is required in reversible computation.

A reversible computation optimized POW problem (abbreviated RCO-POW problem) is a proof-of-work problem which is designed to be efficiently solved by a reversible computer. Since reversible computation is more complex than conventional computation, there will be little commercial interest in reversible computation until reversible computational processes are several times as efficient as conventional computational processes. After all, there is no reason to construct a reversible circuit whose gates are twice efficient as a conventional computer if a conventional circuit that computes the same function uses a third as many gates. On the other hand, an optimized reversible computer will take just as many steps in each attempt to solve an RCO-POW problem as a conventional computer, and the reversible computer does not produce any unnecessary garbage information that must be erased by a conventional computer (though, the reversible computer must uncompute or erase information in order to rewrite the input for the next attempt to solve the RCO-POW problem). Therefore since RCO-POW problems are computationally intensive but are designed to be solved using a reversible computer as easily as they are solved using a conventional computer, computational machinery manufacturers will have an incentive to create reversible computers which are faster and more energy efficient than conventional computers in order for the cryptocurrency miners to save energy and to run their computational equipment faster.

2. Reversible computation

A computation is said to reversible if the computation does not erase or delete any information in any part of the computational process. Said differently, a computation is reversible if for every portion of the computation, the input can be recovered from the output by running the computation in reverse. For example, the AND gate is irreversible since it is impossible to determine the input of the AND gate when the output is FALSE. The OR gate is irreversible for the same reason. However, the NOT gate is reversible since one can recover the output from the input. Since a reversible gate is not permitted to erase any information, the output of a reversible gate must have the same number of bits as the input (the AND gate is irreversible since it has 2 inputs yet only one output).

The gate $T : \{0,1\}^3 \to \{0,1\}^3$ defined by $T(x,y,z) = (x,y,(x \wedge y) \oplus z)$ is known as the Toffoli gate. We shall call the gate $T^* : \{0,1\}^3 \to \{0,1\}^3$ defined by $T^*(x,y,z) = (x,y,(x \vee y) \oplus z)$ the Toffoli* gate. The gate $F : \{0,1\}^3 \to \{0,1\}^3$ defined by $F(0,y,z) = (0,y,z), F(1,y,z) = (1,z,y)$ is known as the Fredkin gate. The gate $C(x,y) = (x, x \oplus y)$ is known as the CNOT gate. The Toffoli, Fredkin, and CNOT gates are all reversible. No reversible gate on 2 bits is capable of universal computation, but the Toffoli gate is sufficient for universal computation and the Fredkin gate is also sufficient for universal computation (AND and OR gates can be simulated using either Toffoli gates with ancilla or Fredkin gates with ancilla). The gates in reversible circuits consist of reversible gates such as Toffoli, Fredkin, NOT, and CNOT gates.

Landauer's principle states that erasing a bit always costs a minimum of $k \cdot T \cdot \ln(2)$ energy where $k = 1.38064852 \cdot 10^{-23} J/K$ and where T is the temperature. At around room temperature of $300K$, by Landauer's principle, erasing a bit costs $2.8 \cdot 10^{-21} J$. Irreversible computation requires one to constantly erase information, so irreversible computation always costs energy. On the other hand, reversible computation does not allow any erasure of information, so the energy efficiency of reversible computation is not limited by Landauer's principle. There is theoretically no limit to the energy efficiency of reversible computation. Since reversible computation in-principle is more energy efficient than irreversible computation, reversible computers can potentially run at much higher frequencies and use much denser parallelism than is possible with an irreversible computer.

Since it is necessary for all computers to eventually erase information and accept new inputs, reversible computers must incorporate some irreversibility. On the other hand, much of the garbage information produced by a reversible computer can be removed by a process called uncomputation.

Suppose that $f : \{0,1\}^n \to \{0,1\}^n$ is a function that one wishes to compute using a reversible computer. Then there exists some bijective

$$C : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n \times \{0,1\}^m$$

easily computable by a reversible circuit along with some $G : \{0,1\}^m \to \{0,1\}^m$ such that $C(x,0) = (f(x), G(x))$ for all $x \in \{0,1\}^n$. While the reversible function C in-essence computes the function f, the function C also produces garbage $G(x)$. Fortunately, the following trick known as uncomputation allows one to remove the garbage information $G(x)$.

Define a mapping

$$L : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^m$$

by letting $L(x,y,z) = (x \oplus y, y, z)$ (L is computable using n CNOT gates). Let $\mathrm{Id} : \{0,1\}^n \to \{0,1\}^n$ denote the identity function. Define

$$H = (\mathrm{Id} \times C)^{-1} \circ L \circ (\mathrm{Id} \times C).$$

Then H is easily computable by a reversible circuit, but

$$(\mathbf{0}, x, \mathbf{0}) = (f(x), x, \mathbf{0}).$$

While the function $\mathrm{Id} \times C$ computes the function f and outputs the garbage $G(x)$, the inverse $(\mathrm{Id} \times C)^{-1}$ uncomputes the function f and removes the garbage $G(x)$. Since the process of uncomputation is reversible, uncomputation removes the garbage $G(x)$ without being subject to Landauer's limit.

Today there are no commercially available reversible computers. Reversible computers are currently mainly of just an academic interest, and today most researchers are only interested in reversible computation since reversible computation is a major component of quantum computation. While reversible computers are not commercially available, there are currently reversible programming languages and reversible compilers.[3]

3. Hash functions and other POW problems are not designed for reversible circuits

There are currently no RCO-POW problems implemented for any cryptocurrency since a POW-problem will only be an RCO-POW problem if it were intentionally designed to be computed reversibly and because the POW problems for cryptocurrencies today are designed to be easily solved by conventional computers instead of reversible computers.

Many cryptocurrency POW-problems require one to find peculiar hashes, but ryptographic hash functions are not designed to be efficiently computed using a reversible computer. Since a cryptographic hash function maps data of an arbitrary length to data of a fixed length, cryptographic hash functions are by their very definition irreversible. Furthermore, reversibility conflicts with the security requirements of cryptographic hash functions, namely pre-image resistance. Many cryptographic hash functions in use today use modular addition (SHA-3 is an exception since SHA-3 does not use modular addition), but modular addition is difficult to implement on a reversible computer since modular addition requires extra ancilla bits and garbage bits. Since most current cryptographic hash functions extensively use modular addition, these hash-based POW problems are not suitable POW problems for Nebula.

While many cryptocurrencies use cryptographic hash functions as their POW problems, there is a movement away from standard hash-based POW problems in order to produce ASIC-resistant POW problems. On the other hand, ASIC resistance appears to be incompatible with reversible computation optimization.

4. The RCO-POW problem R5 for Nebula

R5 is a multi-algorithm RCO-POW problem consisting five different algorithms where the optimal design of the computational machinery of each of these five algorithms is radically different.

[3]The programming language Janus, created in 1982, is the first time-reversible programming language.

The objective of the problem R5 will be to find a 256 bit hash k (the hash function used to compute k will be a standard hash function like SHA-256) of the header for current block in the blockchain along with a 68 bit string x such that $f_i(k\#x) \leq \gamma_i$ (Problem i) for some $i \in \{1, ..., 5\}$ where $\#$ denotes concatenation and where γ_i is a number which is adjusted to tune the difficulty of Problem i in order to satisfy the following conditions:

(1) Each type of problem $1, ..., 5$ is solved for 20% of all blocks.
(2) The average amount of time to solve the next block remains constant.
(3) If Problem i is solved for a certain block in the blockchain, it will be twice as difficult to solve Problem i for the next block in the blockchain (this period where Problem i is more difficult than the other problems is called a recessionary period for Problem i).

The following chart shows how the functions $f_1, ..., f_5$ are computed.

TABLE 1. An overview of the functions $f_1, ..., f_5$

	Computational machinery
f_1	A 2D reversible cellular automaton on an 18×18-square
f_2	A 2D reversible cellular automaton on an 18×18-torus
f_3	A random reversible circuit where the bits are arranged into an 18×18 square, each gate only acts on adjacent bits, and the circuit can be divided into well-defined layers
f_4	A random reversible circuit on 324 bits consisting of a sequence of 192 layers, each layer consists of 108 gates that partition the 324 bits, and the circuit completely changes annually (the new reversible circuits are known 6 months before the annual circuit update)
f_5	A random reversible circuit on 324 bits consisting of a random consecutive sequence of 16384 gates and where the circuit changes slightly after every block; the changes to the random reversible circuits depend on the blockchain and they cannot be calculated in advance

As $i \in \{1, ..., 5\}$ grows larger, it will be more difficult to manufacture a reversible computational device that calculates f_i. For instance, to compute f_1, one will simply need to reversibly embed the cellular automaton rule into an 18×18 square while to compute f_5 efficiently, one will need a reversible circuit that reconfigures one of its gates after every block.

The use of several functions $f_1, ..., f_5$ for our R5 will increase the decentralization of Nebula mining power and hence improve the security of Nebula since it will be difficult for any particular entity to maintain control of a significant portion of mining power. The use of several functions in our POW problems will also help the development of general purpose reversible computers since a variety of RCO-POW problems will prompt the development of a greater variety of reversible computational technologies.

Let $\sigma, \tau, \mu : \{0,1\}^{324} \to \{0,1\}^{324}$ be the mappings where

(1) $\sigma(x_n) = (y_n)_n$ precisely when $y_n = x_n$ whenever $162 \leq n < 324$ and where $y_n = x_n \oplus x_{n+162}$ whenever $0 \leq n < 162$,

(2) $\tau(x_n) = (y_n)_n$ precisely when $y_n = x_n$ whenever $81 \leq n < 324$ and where $y_n = x_n \oplus x_{n+81}$ whenever $0 \leq n < 81$, and
(3) $\mu(x_n) = (y_n)_n$ precisely when $y_n = x_n$ whenever $n \geq 256$ and where $y_n = x_n \oplus x_{256+m}$ whenever $0 \leq n < 256, n = m \mod 68$ and $0 \leq m < 68$.

The functions σ, τ, μ are computable simply by using CNOT gates.

For $i \in \{1, ..., 5\}$, let $f_i = \tau \circ \sigma \circ F_i \circ \mu$ where each F_i will be defined later.

The functions σ, τ, μ make sure that there are no short cuts in computing the function f_i. The function μ also helps increase the diffusion of the nonce for security.

Let $\Sigma : \{0,1\}^{324} \to \{0,1\}^{18\times 18}$ be the function defined by

$$\Sigma(x_n)_{n=0}^{324} = (x_{18i+j})_{0\leq i<18, 0\leq j<18}.$$

Problem 1: Problem 1 amounts to computing a 2D reversible cellular automaton on an 18×18 square.

Define $C_1 : \{0,1\}^{2\times 2} \to \{0,1\}^{2\times 2}$ to be the mapping where $C_1((x_{i,j})_{i,j\in\{0,1\}}) = (y_{i,j})_{i,j\in\{0,1\}}$ precisely when $x_{1,1} = y_{1,1}, x_{0,0} = y_{0,0}$, if $x_{1,1} = 0$, then $x_{0,1} = y_{0,1}, x_{1,0} = y_{1,0}$, and if $x_{1,1} = 1$, then $x_{0,1} = y_{1,0}, x_{1,0} = y_{0,1}$. The mapping C_1 is computed using one Fredkin gate.

Define $C_2 : \{0,1\}^{2\times 2} \to \{0,1\}^{2\times 2}$ to be the mapping where $C_2((x_{i,j})_{i,j\in\{0,1\}}) = (y_{i,j})_{i,j\in\{0,1\}}$ precisely when

$$y_{1,1} = x_{0,1} \oplus x_{0,0} \oplus x_{1,0},$$
$$y_{0,1} = x_{0,1} \oplus x_{0,0} \oplus x_{1,0} \oplus x_{1,1},$$
$$y_{0,0} = x_{0,0} \oplus x_{1,0} \oplus x_{1,1},$$
$$y_{1,0} = x_{1,0} \oplus x_{1,1}.$$

The mapping C_2 is computed using 4 CNOT gates.

Suppose

(1) $C = C_1 \circ C_2$,
(2) $C_{0,0}, C_{0,1}, C_{1,0}, C_{1,1} : \{0,1\}^{2\times 2} \to \{0,1\}$ are the mappings where
$$C((x_{i,j})_{i,j}) = (C_{i,j}((x_{r,s})_{r,s}))_{i,j},$$
and
(3) $E, O : \{0,1\}^{18\times 18} \to \{0,1\}^{18\times 18}$ are the mappings defined by $E((x_{i,j}) = (y_{i,j})_{i,j}$ and $O((x_{i,j})_{i,j}) = (z_{i,j})_{i,j}$ where
$$y_{2i+r,2j+s} = C_{r,s}((x_{2i+u,2j+v})_{0\leq u<2, 0\leq v<2})$$
for $r, s \in \{0,1\}, 0 \leq i < 9, 0 \leq j < 9$ and
$$z_{2i+r+1,2j+s+1} = C_{r,s}((x_{2i+u+1,2j+v+1})_{0\leq u<2, 0\leq v<2})$$
for $r, s \in \{0,1\}, 0 \leq i < 8, 0 \leq j < 8$, and $z_{p,q} = x_{p,q}$ whenever $\{p, q\} \cap \{0, 17\} \neq \emptyset$.

Then define $F_1 = \Sigma^{-1} \circ (E \circ O)^{64} \circ \Sigma$.

Problem 2: Problem 2 requires one to compute a 2D cellular automaton on an 18×18 torus. Problem 2 will be more difficult to compute using a reversible device than Problem 1 since such a device will likely be shaped as a torus instead of a square or at least be a torus folded over itself into a square.

$D = D_1 \circ D_2$. Here $D_2 = C_2$. Define $D_1 : \{0,1\}^{2\times 2} \to \{0,1\}^{2\times 2}$ to be the mapping where $D_1((x_{i,j})_{i,j\in\{0,1\}}) = (y_{i,j})_{i,j\in\{0,1\}}$ precisely when $y_{0,1} = x_{0,1}, y_{1,0} = x_{1,0}, y_{0,0} = x_{0,0}$ and $y_{1,1} = (x_{0,1} \vee x_{1,0}) \oplus x_{1,1}$. The mapping D_1 is computed using one Toffoli* gate.

Suppose

(1) $D = D_1 \circ D_2$,

(2) $D_{0,0}, D_{0,1}, D_{1,0}, D_{1,1} : \{0,1\}^{2\times 2} \to \{0,1\}$ are the mappings where
$$D((x_{i,j})_{i,j}) = (D_{i,j}((x_{r,s})_{r,s}))_{i,j},$$
and

(3) $E', O' : \{0,1\}^{18\times 18} \to \{0,1\}^{18\times 18}$ are the mappings defined by
$$E'((x_{i,j}) = (y_{i,j})_{i,j}, O'((x_{i,j}) = (z_{i,j})_{i,j}$$
where
$$y_{2i+r,2j+s} = C_{r,s}((x_{2i+u,2j+v})_{0\leq u<2,0\leq v<2})$$
and
$$z_{2i+r+1,2j+s+1} = C_{r,s}((x_{2i+u+1,2j+v+1})_{0\leq u<2,0\leq v<2})$$
for $r, s \in \{0,1\}$, where i, j are integers, and where addition is taken modulo 18.

Now define $F_2 = \Sigma^{-1} \circ (E' \circ O')^{64} \circ \Sigma$.

Problem 3:

Suppose that $\alpha_{n,i,j} \in \{0,1,2,3,4,5\}$ for $0 \leq n < 128, 0 \leq i < 9, 0 \leq j < 9$ are pseudorandomly generated and $\beta_{n,i,j} \in \{0,1,2,3,4,5\}$ for $0 \leq n < 128, 0 \leq i < 8, 0 \leq j < 8$ are also pseudorandomly generated.

Let $L_0 = C_2, L_1 = C_2^{-1}, L_2 = D_1$. Let $L_3, L_4, L_5 : \{0,1\}^{2\times 2} \to \{0,1\}^{2\times 2}$ be the mappings where
$$L_3((x_{i,j})_{i,j\in\{0,1\}}) = (y_{i,j})_{i,j\in\{0,1\}}$$
precisely when
$$y_{0,1} = x_{0,1}, y_{0,0} = x_{0,0}, y_{1,1} = x_{1,1}, y_{1,0} = (x_{0,0} \vee x_{1,1}) \oplus x_{1,0}$$

$$L_4((x_{i,j})_{i,j\in\{0,1\}}) = (y_{i,j})_{i,j\in\{0,1\}}$$
precisely when
$$y_{0,1} = x_{0,1}, y_{1,0} = x_{1,0}, y_{1,1} = x_{1,1}, y_{0,0} = (x_{0,1} \vee x_{1,0}) \oplus x_{0,0},$$
and
$$L_5((x_{i,j})_{i,j\in\{0,1\}}) = (y_{i,j})_{i,j\in\{0,1\}}$$
precisely when
$$y_{1,1} = x_{1,1}, y_{0,0} = x_{0,0}, y_{1,0} = x_{1,0}, y_{0,1} = (x_{0,0} \vee x_{1,1}) \oplus x_{0,1}.$$
The functions L_2, L_3, L_4, L_5 are each computed using a Toffoli*-gate.

For $i, j \in \{0,1\}, k \in \{0,1,2,3,4,5\}$, let $L_{i,j,k} : \{0,1\}^{2\times 2} \to \{0,1\}$ be the mapping where
$$L_k(x_{r,s})_{r,s\in\{0,1\}} = (L_{i,j,k}((x_{r,s})_{r,s\in\{0,1\}}))_{i,j\in\{0,1\}}.$$
For $i \in \{0, ..., 127\}$, let $R_i : \{0,1\}^{18\times 18} \to \{0,1\}^{18\times 18}$ be the mapping where
$$R_i((x_{i,j})_{0\leq i<18,0\leq j<18}) = (y_{i,j})_{0\leq i<18,0\leq j<18}$$
precisely when
$$y_{2r+i,2s+j} = L_{i,j,\alpha_{r,s}}((x_{2r+u,2s+v})_{0\leq u<2,0\leq v<2})$$
for $0 \leq r < 9, 0 \leq s < 9, i, j \in \{0,1\}$.

For $i \in \{0,...,127\}$, let $S_i : \{0,1\}^{18\times 18} \to \{0,1\}^{18\times 18}$ be the mapping where

$$S_i((x_{i,j})_{0\leq i<18,0\leq j<18}) = (z_{i,j})_{0\leq i<18,0\leq j<18}$$

precisely when

$$z_{2r+1+i,2s+1+j} = L_{i,j,\beta_{r,s}}((x_{2r+1+u,2s+1+v})_{0\leq u<2,0\leq v<2})$$

for $0 \leq r < 8, 0 \leq s < 8, i, j \in \{0,1\}$ and where $z_{i,j} = x_{i,j}$ whenever $\{i,j\} \cap \{0,17\} \neq \emptyset$.

For $i \in \{0,...,127\}$, let $\Delta_i = R_i \circ S_i$.

Let $F_3 = \Sigma^{-1} \circ \Delta_{127} \circ ... \circ \Delta_0 \circ \Sigma$.

Problem 4: The function F_4 consists of computing a random circuit composed into 192 distinct layers. The function F_4 will change annually and the new function will be announced 6 months in advance.

For $i \in \{1,...,192\}$, let

$$((r_{i,1,1}, r_{i,1,2}, r_{i,1,3}), ..., (r_{i,108,1}, ..., r_{i,108,3}))$$

be a pseudorandomly generated partition of $\{0,...,323\}$. The partitions of the form

$$((r_{i,1,1}, r_{i,1,2}, r_{i,1,3}), ..., (r_{i,108,1}, ..., r_{i,108,3}))$$

will completely change every year and the new partitions will be announced 6 months in advance of their implementation. The new partitions will be constructed pseudorandomly from data from the blockchain.

Now for $i \in \{0,...,191\}$, let $\Gamma_i : \{0,1\}^{324} \to \{0,1\}^{324}$ be the mapping where $\Gamma_i(x_n)_{n=0}^{323} = (y_n)_{n=0}^{323}$ precisely when $y_{r_{i,j,1}} = \neg x_{r_{i,j,1}}$ and if $x_{r_{i,j,1}} = 0$, then $y_{r_{i,j,2}} = x_{r_{i,j,2}}$ and $y_{r_{i,j,3}} = x_{r_{i,j,3}}$ and if $x_{r_{i,j,1}} = 1$, then $y_{r_{i,j,2}} = x_{r_{i,j,3}}$ and $y_{r_{i,j,3}} = x_{r_{i,j,2}}$.

Define $F_4 = \Gamma_{191} \circ \ldots \circ \Gamma_0$.

Problem 5: The function F_5 is computed by a constantly changing random circuit consisting of 16384 random gates. Since the function computing F_5 is constantly changing, it will be more difficult to construct a reversible device that computes F_5 than it will to construct a reversible device for solving problems 1-4.

Let $a_{i,1}, a_{i,2}, a_{i,3} \in \{0,...,323\}$ be randomly selected such that $|\{a_{i,1}, a_{i,2}, a_{i,3}\}| = 3$ for $0 \leq i < 16384$. Let $\ell_i : \{0,1\}^{324} \to \{0,1\}^{324}$ be the mapping such that $\ell_i(x_n)_{n=0}^{323} = (y_n)_{n=0}^{323}$ precisely when $x_n = y_n$ whenever $n \neq a_{i,3}$ and $y_{a_{i,3}} = (x_{a_{i,1}} \vee x_{a_{i,2}}) \oplus x_{a_{i,3}}$. Let $F_5 = \ell_{16383} \circ ... \circ \ell_0$.

After every block for which Problem 5 is solved, we randomly (by randomly, we mean that the selection depends on data from the blockchain) select some $i_0 \in \{1,....,128\}, j_0 \in \{0,1,2\}$ and replace a_{i_0,j_0} with another random number in $\{0,...,323\}$ subject to the condition that $|\{a_{i_0,1}, a_{i_0,2}, a_{i_0,3}\}| = 3$.

5. Discussion

5.1. **Possible technologies for reversible computation.** There are several possible avenues for constructing reversible computers including adiabatic circuits, quantum dot cellular automata [2], mechanical nano-computers [1], and superconducting computers [3]. Reversible quantum dot cellular automata are best suited for solving Problems 1-2. Mechanical nano-computers are best suited for solving Problems 1-3, but it will be difficult for a mechanical nano-computer to solve Problem 4-5 since the gates in Problems 4-5 will be physically distant from each other.

5.2. **Other cryptocurrencies.** We expect that in the future, other cryptocurrencies will use RCO-POW problems to secure their blockchains. While Nebula may not give enough incentive to initially develop the reversible computer, these other cryptocurrencies together with Nebula should provide enough incentive to develop the reversible computer. We recommend for other cryptocurrencies to use R5 or some minor variant thereof as their RCO-POW problem to make it easier for computational machinery manufacturers to produce reversible computers that can efficiently solve these cryptocurrency problems. If there are too many RCO-POW problems in the cryptocurrency ecosystem, then reversible machinery manufacturers must develop too many different kinds of reversible devices in order to solve these RCO-POW problems.

While R5 will motivate the construction of several different kinds of design for reversible computational machinery, other RCO-POW problems will incentivize the construction of different technologies for reversible computation than are incentivized by R5. Such other RCO-POW problems include problems that require the computation of 1D reversible cellular automata (1D reversible cellular automata have the disadvantage that one will need to run the cellular automaton for too many generations in order to obtain a good level of security), 3D reversible cellular automata, and random reversible circuits that need to be completely reconfigured every 2^{32} attempts (such a POW problem will not only spur the development of reversible computers but also of automatically reconfigurable reversible computers). However, we chose not to include these types of RCO-POW problems in R5 because having too many RCO-POW problems will reduce the incentive for constructing a reversible device for solving any particular RCO-POW problem.

5.3. **Reversible computers may pave the way for quantum computers.** There are many similarities between reversible computation and quantum computation, and the construction of a reversible computer may facilitate the production of quantum computers.[4] After all, reversible computers can be thought of as quantum computers except for the fact that with reversible computers the individual bits are not entangled. Since the bits in reversible computing devices are not entangled with each other, reversible computing devices will be much easier to construct than quantum computers. Therefore, reversible computing should be considered as a stepping stone between conventional computing and quantum computing. The best way to construct large scale quantum computers will therefore likely be to first construct energy efficient reversible computers and then use the technology behind reversible computers to construct quantum computers.

5.4. **Nebula in a post-reversible world.** After reversible computers are commercially available and the demand for reversible computers can be easily sustained without cryptocurrency POW problems, there would be little need for Nebula to use POW problems designed to be solved by a reversible computer. At this point

[4]Mechanical reversible computers are not very likely to lead to the development of quantum computers.

which will likely be many years in the future, the POW problem for Nebula should be replaced by some other problems with scientific value[5].

5.5. **Decentralization and security.** Since the main distinguishing feature between Nebula and the other cryptocurrencies is its POW problem, other cryptocurrencies can easily incorporate Nebula's POW problem into their protocols. The POW problem for Nebula satisfies all of the desired properties that a hash-based POW problem satisfies. We do not see any clear disadvantage for using a POW problem suitable for reversible computers over a hash-based POW problem.

Since Nebula and similar cryptocurrencies will be the first market for reversible computing machinery, reversible computing machinery manufacturers will have a strong incentive to ensure that these cryptocurrencies remain secure. These manufacturers will therefore refrain from launching attacks against these cryptocurrencies. Furthermore, these manufacturers will expend a great effort to make sure that these cryptocurrencies remain secure. In particular, these businesses will make a strong effort to ensure that no particular party obtains a large share of the Nebula mining power. These businesses will also promote and use Nebula in order to increase the market cap for Nebula in order to more easily develop reversible computing devices.

5.6. **Concluding remarks.** Some have attempted to justify the energy and resource usage of cryptocurrency mining in various ways. Some state that the energy used to mine cryptocurrencies is used to heat homes. However, without data about how much cryptocurrency mining is actually purposely used to heat homes in the Winter, one must dismiss this argument until data is used to prove this point.

Some state that the expense is justified since the expense is required to secure the cryptocurrency and to distribute newly minted coins. Others have stated that gold and fiat currencies require much more energy and resources than cryptocurrencies. While these arguments do justify the use of a POW problem even if it expends much energy, these arguments do not justify the use of a hash-based POW problem over a useful POW problem.

Today, many people are concerned about the environmental impact of cryptocurrency mining. However, an RCO-POW problem could be considered environmentally friendly in the long-term since reversible computers will be more energy efficient than conventional computers.

References

1. http://www.imm.org/Reports/rep046.pdf
2. http://accentsjournals.org/PaperDirectory/Journal/IJACR/2012/3/14.pdf
3. https://link.springer.com/chapter/10.1007/978-3-319-08494-7_2

E-mail address: jvanname@mail.usf.edu

[5]While there are no good useful POW problems found in other cryptocurrencies, since cryptocurrencies are still quite new, people will develop many useful POW problems for cryptocurrencies in the future. Furthermore, the stringent requirements for a POW problem for a cryptocurrency are partially relaxed when there are several different kinds of POW problems instead of just one and when there is an algorithm that removes broken POW problems.