Hybrid Mining

Exploiting Blockchain's Computational Power for Distributed Problem Solving

Krishnendu Chatterjee
IST Austria
Klosterneuburg, Austria
krishnendu.chatterjee@ist.ac.at

Amir Kafshdar Goharshady
IST Austria
Klosterneuburg, Austria
amir.goharshady@ist.ac.at

Arash Pourdamghani
Sharif University of Technology
Tehran, Iran
pourdamghani@ce.sharif.edu

ABSTRACT

In today's cryptocurrencies, Hashcash proof of work is the most commonly-adopted approach to mining. In Hashcash, when a miner decides to add a block to the chain, she has to solve the difficult computational puzzle of inverting a hash function. While Hashcash has been successfully adopted in both Bitcoin and Ethereum, it has attracted significant and harsh criticism due to its massive waste of electricity, its carbon footprint and environmental effects, and the inherent lack of usefulness in inverting a hash function. Various other mining protocols have been suggested, including proof of stake, in which a miner's chance of adding the next block is proportional to her current balance. However, such protocols lead to a higher entry cost for new miners who might not still have any stake in the cryptocurrency, and can in the worst case lead to an oligopoly, where the rich have complete control over mining.

In this paper, we propose Hybrid Mining: a new mining protocol that combines solving real-world useful problems with Hashcash. Our protocol allows new miners to join the network by taking part in Hashcash mining without having to own an initial stake. It also allows nodes of the network to submit hard computational problems whose solutions are of interest in the real world, e.g. protein folding problems. Then, miners can choose to compete in solving these problems, in lieu of Hashcash, for adding a new block. Hence, Hybrid Mining incentivizes miners to solve useful problems, such as hard computational problems arising in biology, in a distributed manner. It also gives researchers in other areas an easy-to-use tool to outsource their hard computations to the blockchain network, which has enormous computational power, by paying a reward to the miner who solves the problem for them. Moreover, our protocol provides strong security guarantees and is at least as resilient to double spending as Bitcoin.

CCS CONCEPTS

• **Applied computing** → **Electronic commerce**; **Secure online transactions**; • **Computing methodologies** → **Distributed computing methodologies**;

KEYWORDS

Blockchain, Proof of Work, Mining, Distributed Problem Solving, Collaborative Problem Solving

1 INTRODUCTION AND PRELIMINARIES

The aim of this paper is to develop a blockchain mining protocol that can be exploited for distributed problem solving. In this section, we review basics of blockchain, mining, and distributed problem solving.

1.1 Blockchain and Mining

Bitcoin. Bitcoin was the first cryptocurrency that provided a protocol to achieve consensus about the ownership of funds, without a need for relying on a central governing authority or bank [29]. This decentralization was very well-received and, at the time of writing, Bitcoin is the largest cryptocurrency in the world with a market cap of more than 100 billion dollars [10].

Double Spending. The most basic attack in every electronic monetary system is that of double spending. In the real world, units of currency cannot be duplicated and hence one can pay for goods and services by simply transferring a banknote or coin to the recipient. However, in a digital setting, a coin is just a file or a collection of data and one can copy it and send each copy to a different recipient, effectively creating money out of thin air. A solution to avoid this problem, as employed by Bitcoin, is to have every transaction broadcast to the entire currency network [29]. This way, every participant in the network would reject a transaction that is spending the same coin for a second time. However, the problem of double spending is still not solved entirely. If one creates two transactions, sending the same coin to different recipients, and publishes both at the same time in different parts of the world, then due to network latency, some nodes of the network would receive the first transaction and reject the second one as double spending, while other nodes might do the opposite. Bitcoin has an elegant way of ensuring that the entire network reaches a consensus about the results of transactions. This protocol is called the Blockchain [29].

Blockchain. In a cryptocurrency, a blockchain is a distributed ledger of transactions. The transactions are grouped into fixed-size blocks. The chain starts with a predefined block and each block has a pointer to its preceding block. Hence, a blockchain is effectively a singly-linked list of blocks [37]. Every node of the network keeps a local copy of the blockchain. To ensure that all local copies eventually agree on the contents of the blockchain, one cannot simply add new blocks as she wishes. Instead, adding blocks is regulated by a process called mining [29, 37].

Mining. Mining is the process by which a node of the network is selected and permitted to add a new block to the blockchain [19, 37]. The selected node forms a block using the valid transactions she has received. She then broadcasts the block to the entire network. Every other node of the network checks two conditions: (i) whether all the transactions in the broadcast block are valid, and (ii) whether the broadcaster had the right to add a new block according to the mining protocol. If both checks pass, then the node adds this block to the end of her local copy of the blockchain. If the mining protocol always chooses exactly one node for adding the new block, then there will be no conflicts and the entire network would be in consensus about the contents of the blockchain. However, in Bitcoin and other cryptocurrencies, the participants are anonymous, hence

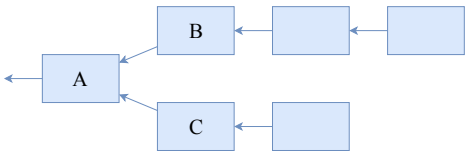

Figure 1: A Fork in the Blockchain: The two blocks B and C are both announced as valid successors to A. However, the chain containing B has become longer and is now the consensus chain. Hence, C and its successors will be dropped by the network nodes.

a protocol cannot simply point to the next person who is allowed to add a block.

Proof of Work. Bitcoin's solution to the problem above is to use a Proof of Work (PoW) protocol. In a PoW, a hard computational problem is chosen and one can add a new block as soon as she succeeds in solving the problem [4]. In case of Bitcoin, the used protocol is Hashcash [3], i.e. one has to find a nonce value, such that the result of applying the SHA256 hash function to a tuple consisting of her public key, the hash of the previous block, the hash of the current block that is being added, and this nonce value, is at most a predefined amount[1]. Given the one-way property of hash functions, the only strategy for finding a nonce, and hence getting the right to add a new block, is to repeatedly try randomly-generated nonces until one of them solves the problem. This process is called Proof of Work mining and the nodes taking part in finding nonces are called miners. When a miner finds the right nonce, she broadcasts the new block, together with the nonce. Then, every other node of the network can perform the two previously-mentioned checks and add the block to her local copy of the blockchain [29].

Ensuring Consensus. It is possible that two miners succeed in solving the Hashcash problem at approximately the same time. If this happens, then two new blocks would be broadcast, and network latency leads to a fork in the blockchain, i.e. a situation in which some of the nodes have added the first new block to their local copy, and others have added the second one. In such cases, the blockchain protocol considers both chains to be valid and miners can choose the chain they want to extend. However, as soon as one of the chains becomes longer than the other ones, the protocol dictates that the longest chain is the consensus chain and all other chains will be dropped by everyone in the network [29] (See Figure 1). The probabilistic nature of the mining process guarantees that one of the chains would eventually get longer [18], and therefore the whole network would reach consensus after a few blocks[2].

Incentives for Mining. PoW mining requires considerable computational power and electricity usage. Hence, the process is not free for the miners. Therefore, there should be compensations in place to ensure that network participants are incentivized to mine. Bitcoin uses two incentives for the miners [29]:

- *Block reward.* Any miner who successfully adds a block to the blockchain gets a predefined reward. The reward is currently set at 12.5 bitcoins, which is roughly equal to 75000 dollars at the time of writing. Hence, in Bitcoin, block rewards are also used as the mechanism for creating new units of currency.
- *Transaction Fees.* While block rewards incentivize miners to add new blocks to the blockchain, the miners might choose to add empty blocks, i.e. blocks that contain no transactions. There should be an additional incentive for the miners in order to ensure that a transaction is eventually picked up by the miners and added to the blockchain. Bitcoin achieves this using transaction fees. Each transaction in Bitcoin can set a reward, or transaction fee, that is paid to the miner who puts the transaction in a block and adds it to the blockchain.

[1]In Bitcoin, this predefined amount is set in a manner that ensures a new block is added roughly every 10 minutes.

[2]In Bitcoin, and consequently in our approach, the length of a chain is not the number of blocks in that chain, but the sum of the hardnesses of the PoW problems that have been solved in that chain.

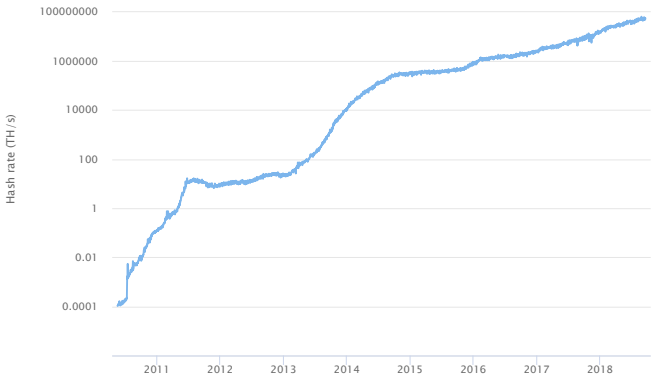

Figure 2: Bitcoin network's estimated hash rate (in trillions of hashes per second). The data is provided in [34, 36]. Note that the graph is in logarithmic scale.

Computational Power of the Bitcoin Network. The rising price and popularity of Bitcoin has turned mining into a lucrative and highly-specialized industry. The computational (mining) power of the Bitcoin network has been increasing exponentially since its inception. Figure 2 shows the growth in the overall hashrate of the Bitcoin network, i.e. the estimated number of hashes that are computed every second. Note that this figure is in logarithmic scale. Already in 2013, Bitcoin network's computational power was reported to be more than 250 times the computational power of the world's 500 fastest supercomputers combined [9].

Specialized Mining Hardware. While anyone can take part in mining using their personal computer, the overall computational power of the bitcoin network is vast and the chances of finding a new block using everyday hardware are slim. Therefore, many specialized hardware modules have been designed for mining. These hardware modules are optimized for the single task of inverting SHA256 hash functions and are extremely efficient [35].

Pool Mining. Given the hardness of mining, and in order to decrease the variance in their revenue, miners often collaborate in *pools* [7, 33]. A pool is basically a group of miners who solve PoW problems using the same public key and share the rewards among themselves. Usually, each pool has a manager who receives all the rewards and divides them among the miners based on their overall contribution. The contribution of a miner is often assessed by partial proofs of work, i.e. proofs of work on the same problem, but with a lower difficulty.

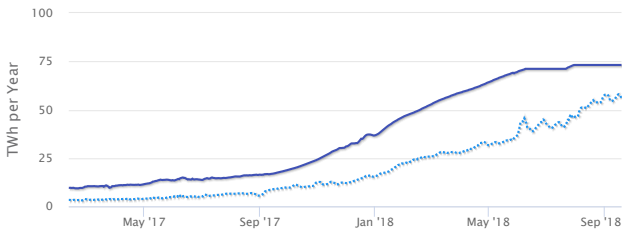

Figure 3: The growing energy consumption of Bitcoin mining. The solid line shows the estimated energy consumption and the dotted line is the minimum possible consumption. The graph above is adapted from [12, 13].

Criticisms and Alternatives. There are reports that the Bitcoin PoW mining uses more electricity than the entire country of Ireland [30]. Moreover, Bitcoin mining's energy consumption has been steadily increasing [12]. See Figure 3. On the other hand, the task of inverting a hash function is inherently useless, and there is no justification for it other than the incentives given by the Bitcoin protocol. Moreover, PoW has considerable environmental consequences and a huge carbon footprint [17] and has been called an environmental disaster by the Bank for International Settlements [6]. Therefore, the process of mining in Bitcoin has been harshly criticized and many other alternative mining protocols have been suggested in order to reduce the energy usage and carbon footprint of blockchains [16, 39]. Two of the most notable examples are proof of stake [21] and proof of space [14]. In proof of stake, one's chance of getting selected as the miner to add the next block is proportional to the number of cryptocurrency units in her possession, and proof of space replaces energy-consuming computations in PoW with memory-consuming computations.

Obstacles to Adoption. There has been significant resistance against adoption of new mining protocols, especially in Bitcoin [15]. There are two major reasons for this: (i) some protocols, such as proof of stake, favor the richer members of the network [32] and are considered to be more prone to becoming centralized, i.e. leading to a situation in which a few members of the network have complete control over mining, and (ii) many miners have already invested significant amounts of funds into acquiring specialized PoW mining hardware [35] and are not willing to suddenly switch away from it to other specialized hardware, e.g. for proof of space.

1.2 Distributed Problem Solving

In this section, we provide a short review of distributed problem solving approaches.

Distributed Problem Solving. There are many intractable problems in various fields of science, for example computational biology [11], that lack efficient algorithms and finding their solutions takes decades of time on a normal computer. A very costly and non-scalable approach to solving such problems is to use supercomputers with more computing resources. However, many of these problems can also be solved using a new paradigm, which is called Distributed Problem Solving (DPS). In DPS, many computers on a network attempt to solve a single problem. Hence, the computational power that might otherwise be dormant in machines all around the world is effectively used to solve an important real-world problem.

Examples. One of the most well-known examples of DPS is the Great Internet Mersenne Prime Search project [40], in which the problem of finding huge prime numbers of the form $2^p - 1$ is solved in a distributed manner. Another example is folding@home [25], in which the computationally hard problem of protein folding, which is of great interest to biologists, is tackled by DPS. However, these projects do not provide incentives for contributors and most of the computation is carried out by volunteers, hence the overall computational power of the their networks is far less than that of blockchains. Another notable project is Foldit [11], in which the problem of finding protein structures is translated to a game, hence using the enjoyment of gaming as an incentive for solving a real-world scientific problem. For more examples of DPS see [1, 24]. In this paper, we propose a new mining protocol that provides Bitcoin-like incentives for solving such real-world problems, thereby allowing the DPS networks to grow beyond their volunteer community.

Relationship to Pool Mining. Note that pool mining can itself be considered a special case of DPS, given that miners in the pool are trying to solve an instance of the Hashcash problem in a distributed manner. This observation directly leads to the main goal of this paper, i.e. using the same mechanisms and incentives as mining for distributed solution of real-world practical problems.

NP Problems. The real-world problems mentioned above, that are solved by DPS, belong to the class NP. The class NP represents problems that can be solved in non-deterministic polynomial time, which means that if a correct answer is guessed/provided, it can be verified in polynomial time. However, finding the answer might be a much more challenging problem. The class of problems that can be solved in polynomial time is called P. It is a major open question whether P=NP, and it is widely believed that P$\neq$ NP [2].

NP-completeness. The class of NP-complete problems represents the core difficult problems of NP. Informally, NP-complete problems (i) belong to the class NP and (ii) are as hard as any other problem in NP. The notion of as hard as any other problem is captured by showing that there is a polynomial-time reduction from any other NP problem to an NP-complete problem [20]. This implies that an efficient (polynomial-time) algorithm for an NP-complete problem would imply P=NP, which is unlikely, and these NP-complete problems are considered computationally intractable [2]. Many of the DPS problems above are known to be NP-complete.

Choosing a Representative Problem. Given an NP-complete problem A, any other problem in NP, including other NP-complete problems, can be reduced to A in polynomial time. Hence, instead of creating distinct DPS networks for each problem, we can simply choose a representative problem A that the whole network tries to solve, and ask owners of real-world problems, such as computational biologists, to translate (reduce) their problem into A before submitting it to the network for solution.

Satisfiability. Any NP-complete problem can be used as a representative for DPS. In this paper, we use the boolean satisfiability (SAT) problem as the representative. However, our approach is not dependent on this and SAT can be replaced by any other NP-complete problem. We chose SAT because there has been ample research into solving real-world instances of this problem [8]. An

instance of SAT is a propositional logic formula φ in conjunctive normal form, consisting of boolean variables $x_1, x_2, \ldots, x_n$. The SAT problem asks whether one can find true/false assignments to the variables such that the formula φ is satisfied. The size $|\varphi|$ of the formula is called the length of the SAT instance, and n is its order.

1.3 Our Contribution

In this paper, we present Hybrid Mining: a new mining protocol that combines solving real-world instances of NP-complete problems with Hashcash mining. In our protocol, any node in the network can submit a problem, together with a reward for its solution. Then, every other node of the network can add a new block to the blockchain by either solving a submitted problem or taking part in Hashcash PoW. Our approach has the following benefits:

(i) It provides an easy-to-use protocol for DPS, where people can outsource their problems to be solved using the enormous computational power of the blockchain.
(ii) It gives miners a strong incentive to solve useful real-world problems, hence decreasing the waste in energy usage and carbon footprint of PoW.
(iii) It gives current Bitcoin miners the choice of reverting back to PoW in case they cannot solve any of the submitted problems, hence making use of the already purchased resources and specialized hardware.
(iv) On the other hand, new miners who have not invested in such hardware can join by using software and hardware technologies for solving NP-hard problems, especially SAT. This can in turn lead to financial incentives for research and development in solving such problems and gaining new insights to them.
(v) It is at least as secure, and resilient to centralization, as PoW mining.

To the best of our knowledge, this is the first mining protocol that incentivizes solving real-world hard problems that are of immediate practical interest to researchers in various fields.

1.4 Related Work

There has been ample research in finding alternative mining protocols, or as they are commonly called, proofs of *useful* work. We list some of these approaches below.

- *Number Theoretic Approaches.* Primecoin [22] and Gapcoin [23] are two approaches to mining that use alternative number-theoretic problems, instead of hash inversion, for PoW. Primecoin PoW asks for chains of prime numbers with a specific property, and Gapcoin PoW searches for large prime gaps.
- *Approaches using NP-completeness.* Using NP-complete problems for mining has been studied extensively and there are a variety of mining protocols using different NP-complete problem instances as their PoW. For example, see [5, 28, 31].

Note that the approaches mentioned above have some important limitations, which are not present in Hybrid Mining. Here are some examples:

- Most of these approaches are limited to finding solutions for a specific problem, such as prime gap search. These problems are not necessarily NP-complete. Our approach allows practical scientists to post any problem by simply translating it to SAT.
- These approaches are often referred to as proofs of *useful* work. However, the word "useful" should be taken with a grain of salt. While the problems solved for mining new blocks are indeed useful, the instances that are being solved are not real-world instances. Instead, they are artificially generated, either randomly or by a process that ensures they have a certain difficulty. This means that the resulting solutions are useful in the sense that they provide new insights into the hardness of the problem at hand, but they do not solve a real-world problem. On the other hand, our approach uses real-world instances that are submitted by and of immediate practical value to their owners, who can be scientists in other fields, including computational biology.
- Following the previous point, these approaches cannot be used for DPS, because the instances that are being solved are not proposed by an external entity, but generated by the protocol itself. A major novelty of our work is that we combine DPS with mining, hence creating new incentives for taking part in DPS.
- Virtually all of these approaches, with the exception of [28], are incompatible with currently-used PoW hardware and therefore alienate a majority of miners. In contrast, our Hybrid Mining protocol welcomes current miners and provides an easy way for them to use their current hardware, while also creating incentives for new types of mining.

2 PROBLEM STATEMENT AND REQUIREMENTS

We now define the problem we are aiming to solve and the requirements we expect of the final solution. We are aiming for a blockchain mining protocol that combines Hashcash with Distributed Problem Solving (DPS). In order to be useful, such a protocol must meet the following criteria:

(1) Functional Requirements:
  (1.a) Owners of real-world problems should be able to publish their problems for distributed solution by the network, i.e. the computing power of the network should be usable for DPS.
  (1.b) Conversely, miners should be able to add new blocks to the blockchain by solving the problems published by others. Hence, creating an incentive for them to solve real-world useful problems.
  (1.c) Miners should have the option to use Hashcash for adding a new block, e.g. in cases when they are not able to solve any of the published real-world problems. This condition also ensures that current miners who have invested in dedicated Hashcash PoW hardware can continue to mine in the new protocol, and that new blocks will be added even if there are no remaining problems to solve.
(2) Security Requirements:
  (2.a) The protocol must remain resilient to double spending.
  (2.b) The protocol must ensure that a miner is rewarded for solving a published problem, and that no miner can steal solutions found by another miner.
  (2.c) The protocol must be resilient to spamming and publication of fake or long problems that aim to thwart the network.
  (2.d) The protocol must ensure that no party, neither a problem proposer, nor a miner, can benefit from dishonest behavior that is detrimental to any other party.

3 HYBRID MINING

In this section, we propose a new mining protocol, called Hybrid Mining, and show the details of its operation. In the next section, we will show that Hybrid Mining satisfies all the required criteria mentioned above. As expected, in our approach, anyone can propose a new problem in the form of a SAT instance, and one can add a new block to the blockchain by either solving Hashcash or a previously-unsolved problem on the blockchain. We now provide detailed protocols for each of these actions.

3.1 Problem Proposal Protocol

In our system, proposing a problem is considered as a special type of transaction. Consider a problem owner Owen, who likes to broadcast his problem to the network, so that miners solve it for him. To do so, he creates a transaction consisting of the following parts and broadcasts it to the network:

- Owen's public key,
- The problem instance φ,
- An expiration date, setting the maximum amount of time Owen is willing to wait for the problem to be solved by the network[3],
- Four distinct payments as follows:
  - A constant problem proposal fee of f_1 units,
  - A problem storage fee of $|\varphi| \cdot f_2$ units,
  - A reward value r, chosen by Owen, which will be paid to the first miner who successfully solves the problem, and
  - A transaction fee t, also chosen by Owen.
- Owen's signature proving that he is the one broadcasting the transaction.

We call this problem proposal transaction a *Type 1* message. Note that Owen has to translate (reduce) his problem to a SAT instance φ, before publishing it on the network. We now explain the reason behind each of the payments:

- The constant problem proposal fee f_1, which is paid by Owen and is the same as the block reward fee, ensures that one cannot overwhelm the network with many spam problems. Also, it ensures that one cannot add new blocks by adding simple problems and solving them on his own. This money is burnt if the problem is solved, but it will be refunded to Owen if the problem remains unsolved until its expiry time.
- The second payment is correlated with the length of the problem instance. Every node of the network has to save the problem instance φ. Hence, in order to avoid unnecessarily long instances that aim to thwart the network, one has to pay a "storage fee" that correlates with the length of the problem. This idea is similar to the concept of Gas in Ethereum [38].
- In order to incentivize miners to solve the problem, Owen has to set a reward r, which will be paid to the miner that provides a solution to φ.
- As with any other transaction, Owen should pay a transaction fee t, in order to have his transaction processed and added to a block by a miner.

Note that transactions are processed in essentially the same way as Bitcoin. If the problem is not solved until its expiration time, then Owen can create another transaction that refunds the payment to him. We call such a transaction a *Type 2* message. The Type 2 transaction would be rejected if it is broadcast prematurely, i.e. before the expiry time of the problem.

[3]This time limit can be set by fixing the maximum number of Hashcash blocks that can be added to the chain without expiring the problem.

3.2 Mining Protocol

Consider a miner Mindy, who has formed a block of valid transactions and wants to add it to the blockchain. She has two choices for mining. She can either take part in Hashcash mining or solve one of the currently-unsolved problems on the blockchain. If she solves a Hashcash problem, everything will be similar to Bitcoin. She receives a block reward of f_1 and transaction fees from the transactions she has included in her block. As in Bitcoin, this is also how new coins are created in Hybrid Mining.

On the other hand, if she solves a previously-unsolved problem that has been proposed on the blockchain, she does not receive a block reward, but instead receives the solution reward r which was previously paid by the owner of the solved problem. In any case, Mindy can also claim her transaction fees. The exact protocol for adding a block by solving problems is explained in the next section.

Consensus Chain. As in Bitcoin, we consider the longest chain in the network to be the consensus chain. However, given that the difficulty of problem instances is not well-defined, we consider the length of a chain to be the sum of difficulties of those blocks in the chain that are mined by Hashcash.

Number of Blocks of Each Type. Note that each block that is mined by Hashcash creates f_1 new units of currency. On the other hand, one has to pay a proposal fee of f_1 to add a problem and this fee is burnt in case the problem gets solved and used for adding a new block to the chain. Hence, it is guaranteed that, at any point in time, there are at least as many blocks mined by Hashcash, as by solving proposed problems.

3.3 Solution Announcement and Reward Collection Protocol

If a miner Mindy finds a solution to Hashcash, she proceeds with announcing it in the same manner as in Bitcoin. No other miner can claim the rewards for this solution, because Mindy's identity is part of the Hashcash problem.

However, if she wants to add a block by solving a proposed problem, she has to publish the solution. This is problematic, because Mindy's identity is not part of the problem, and another miner, Manny, might see the solution, form another block with the same solution and publish it shortly after Mindy. Then network latency can lead to a situation where part of the network believes Manny was the one who solved the problem first.

To avoid this problem, our protocol requires Mindy to announce the block in a two-step manner, as described below.

In the first step, Mindy chooses a secret value s, and then announces the block she has created, which consists of the following parts:

- A block header, showing that the block was created by solving a problem and pointing to the problem that is being solved.
- Mindy's public key.
- A hash value h obtained by hashing a tuple consisting of the solution to the problem, the secret value s, and Mindy's public key.
- A deposit of $2 \cdot f_1 + |\varphi| \cdot f_2$ currency units.

- A storage fee of $n \cdot f_2$ units, which is a compensation for the whole network having to store the solution. Note that n is the number of variables in the problem instance.
- A list of transactions that are being included in the new block.
- A pointer to the previous block in the blockchain.
- A signature by Mindy, proving that she is the one who broadcast this block.

We call this announcement a *Type 3* message. Note that the block will be rejected by the network if another block in the blockchain is already claiming to have solved the same problem. By providing h, Mindy is committing to her solution without actually publishing it, i.e. she is promising that she has obtained a solution that, together with the secret value s and her public key, can produce the hash value h. After publishing this block, which includes paying a fee as a deposit for this promise, the block will be added to the blockchain by every node of the network. However, no reward is paid to Mindy until she provides the full solution, i.e. she neither receives her reward r, nor her transaction fees.

In the second step, Mindy has to publish a transaction, claiming the rewards of the block she added to the chain, as well as her deposit. We call this a *Type 4* message. Mindy can opt to wait for the chain that includes her block to get longer, so that when she publishes the full solution she is already ensured that the chain will not revert and no one else can claim her rewards. However, there is a certain fixed length limit k, and Mindy has to publish her solution before the length of the chain containing her block is increased by k units. If she fails to do so, she will not receive her solution reward, transaction fees or deposit. Instead, the problem owner, Owen, will be refunded, i.e. he gets back all of his payments that were due to this problem.

Concretely, in the second step, when Mindy is already ensured that her block is going to remain in the blockchain, she can create a specific type of transaction for claiming her money. This transaction has the following parts:

- Mindy's public key,
- A pointer to the block whose rewards are being claimed,
- The full solution to the problem,
- The secret value s, and
- A transaction fee.

When this transaction is published, other nodes of the network validate it by checking that (i) the solution is correct, (ii) the problem has not expired, (iii) the length of the chain has increased by at most k since the block was added, (iv) the hash of the tuple consisting of the solution, the secret value s, and Mindy's public key is the promised value h. If all the checks pass, Mindy's deposit will be returned back to her, and she will receive the problem reward r and the transaction fees in her block.

Note that all transactions are permanently saved on the blockchain and the second step transaction above contains the answer to the problem posed by Owen. So Owen can simply read the answer from the blockchain.

3.4 Overview of the System

We now summarize all parts of the Hybrid Mining protocol. Any aspect of the protocol that is not mentioned here follows the Bitcoin protocol as in [29]. Table 1 provides a summary of constants that we use in Hybrid Mining. The values of these constants can be set at the time of implementation. In order to support mining by problem solving, we introduced 4 new types of messages that are not present in Bitcoin. Table 2 summarizes the new message types. Figure 4 shows a diagram of the behavior of an honest problem owner, Owen, and Figure 5 shows the behavior of an honest miner, Mindy. The circled numbers in these figures denote the type of messages that are broadcast or waited for.

In our protocol, a network node works in exactly the same manner as in Bitcoin. It listens for transactions and blocks that are broadcast on the network. If it receives a valid new transaction, it broadcasts it to the rest of the network. Similarly, if it receives a valid new block, it adds the block to its local copy of the blockchain and broadcasts it (see Figure 6). Moreover, if a longer chain is formed, the node will adopt the new chain and broadcast it. Note that the length of a chain is defined in the same manner as Bitcoin and is the sum of difficulties of Hashcash problems solved in that chain. The only difference with Bitcoin comes from the additional checks

Constant	Description
f_1	The block reward for mining a Hashcash block, and also the problem proposal fee for adding a new problem
f_2	The storage fee for storing one byte of data, either as part of a problem or its solution
k	The maximum allowed waiting time between claiming a solution and announcing it. The time is measured by the increase in chain length.
m	The rate at which new Hashcash blocks are added to the blockchain, i.e. a new Hashcash block should be mined roughly every m minutes.

Table 1: Summary of Implementation-dependent constants in Hybrid Mining

Type	Description	Issuer	Effect
1	Problem Proposal Transaction	Problem Owner (Owen)	Adds a new unsolved problem to the Blockchain.
2	Refund Transaction	Problem Owner (Owen)	Refunds Owen if his problem has expired or claimed but not solved.
3	Committing to a Solution and Announcing a New Block	Miner (Mindy)	The block is added to the Blockchain, but Mindy's deposit and rewards are not paid until she publishes the entire solution, hence proving her claim.
4	Solution Announcement	Miner (Mindy)	If the solution is correct, Mindy's deposit is paid back to her, together with the reward for solving the problem and transaction fees in the resulting block.

Table 2: Summary of the 4 new types of messages in Hybrid Mining

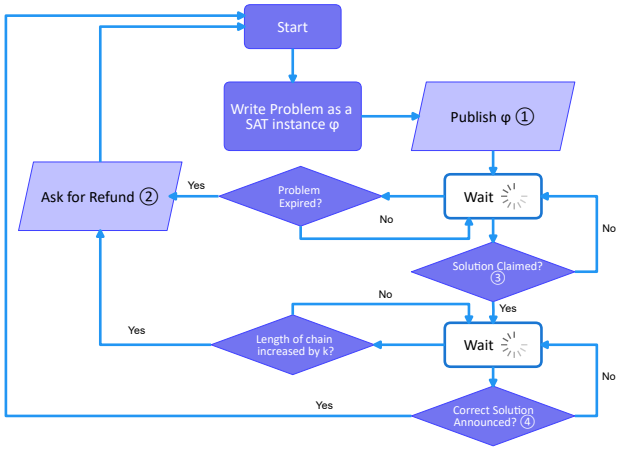

Figure 4: The behavior of an honest problem owner, Owen, in Hybrid Mining. The circled numbers denote the type of messages that are broadcast or expected.

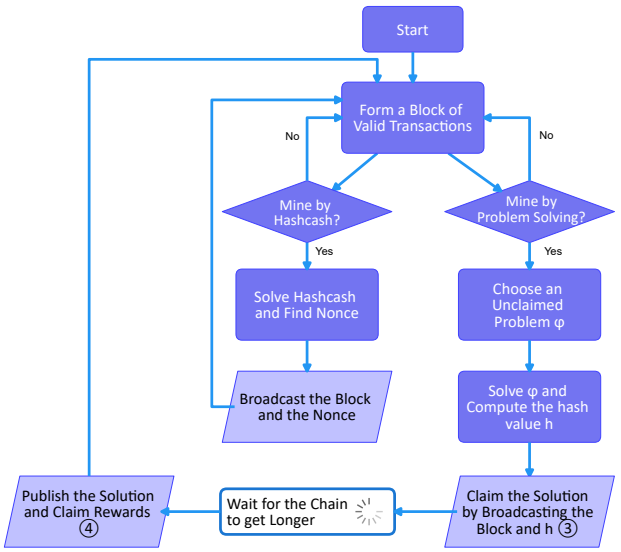

Figure 5: The behavior of an honest miner, Mindy, in Hybrid Mining. The circled numbers denote the type of messages that are broadcast.

that are performed for deciding whether blocks and transactions resulting from the 4 new types of messages are valid.

4 REQUIREMENTS ANALYSIS

In this section, we show how our Hybrid Mining protocol satisfies the requirements enumerated in Section 2. We deal with functional and security requirements separately.

4.1 Functional Requirements Analysis

Satisfaction of the functional requirements directly follows from the modeling and design of our protocols. The Problem Proposal Protocol of Section 3.1 allows all nodes to submit new problems for

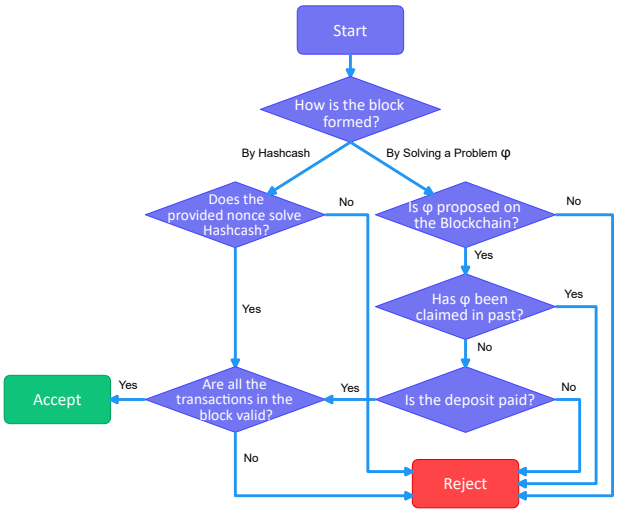

Figure 6: The process used by a network node to check validity of a newly broadcast block.

solution by the network. Hence, Hybrid Mining satisfies requirement (1.a). As illustrated in Section 3.2, miners can choose to mine using Hashcash or solving previously-submitted problems. The former is done in exactly the same manner as in Bitcoin and Section 3.3 provides a protocol for the latter. Hence, our approach satisfies requirements (1.b) and (1.c), as well. Therefore, we have the following proposition:

Proposition 1. *Hybrid Mining satisfies all the functional requirements listed in Section 2, i.e. it allows problem owners to submit their problems for solution in the same manner as in DPS, and allows miners to mine either by Hashcash or by solving problems.*

4.2 Security Requirements Analysis

In this section, we analyze the security of Hybrid Mining against several attacks and show that it satisfies the security requirements of Section 2.

Resilience to Double Spending. Given that at any point, at least half of the blocks are mined by Hashcash, and that a new Hashcash block is added roughly every m minutes[4], our protocol is at least as resilient to forks and double spending as Bitcoin, and the same arguments for Bitcoin's resilience, as in [29], can be applied to our setting without any change. Note that we consider the longest chain to be the consensus chain, and we define the length of a chain as the sum of difficulties of Hashcash blocks in the chain, ignoring the blocks obtained by problem solving. Hence, one cannot create a long chain and double spend by simply solving (possibly easy) problems. Therefore, Hybrid Mining is resilient to double spending and satisfies requirement (2.a).

Analysis of Incentives for Honest Behavior. In several points, our protocol relies on honest behavior of participants, including miners and problem owners. We now show that such behavior is always incentivized, and that the system is resilient to the following attacks:

[4]This constant time is set to 10 minutes in Bitcoin, but can be changed in implementation.

- *Falsely Claiming Solutions.* In order to claim a solution, one has to pay a deposit of $2 \cdot f_1 + |\varphi| \cdot f_2$. If the solution is incorrect or not provided later, $f_1 + |\varphi| \cdot f_2$ units of this deposit is used to refund the problem owner, who might be the same person who published the false solution claim. However, at least f_1 units of currency will be burnt. Hence, adding a block by falsely claiming a solution has a cost of at least f_1. Note that this means anyone can pay f_1 units just to add a new block, but this is not a rational strategy and does not have any other effect on the blockchain.
- *Invalidating a Problem.* Another reason for falsely claiming solutions might be to invalidate a problem. For example, if a miner Mindy wants to hurt the problem owner Owen, she can publish a false solution. This leads to a refund for Owen, except for the transaction fees he paid for adding the problem. However, as mentioned before, such behavior will cost Mindy at least f_1 units, which is equal to a full block reward. A block reward is often much more than a transaction fee. Hence, this is not a rational attack for Mindy and has negligible effect on Owen.

In both cases above, as well as in the other attacks mentioned below, the misbehaving party is never benefiting from his dishonest behavior. Rather, such dishonest actions lead to significant adverse effects and losses for the party that takes them. Hence, there is no incentive for such behavior and our approach satisfies requirement (2.d). Also note that, even if the attackers are assumed to be irrational and perform an attack despite the losses it causes them, none of these attacks lead to significant losses, i.e. losses beyond a small transaction fee, for any other party.

Ensuring Miners are Rewarded for Solving a Problem. A miner who solves a problem can claim the reward set by the problem owner. We now show that our system is resilient to attacks that aim to deprive a miner from her problem solving reward or to steal the reward and claim it for another miner.

- *Not Processing Another Miner's Transaction.* This line of attack is between miners. Suppose Mindy has solved a problem and published her first step block. After a while, she publishes the transaction for claiming the rewards. Now suppose that the next block is mined by Manny. Manny can opt not to include Mindy's transaction in his block. However, this means that Manny loses the transaction fee. Hence, this is not a rational move for Manny. The only way that withholding transactions, as described above, can make sense for Manny, is if he expects to be able to claim the reward himself. However, this is never possible, because after the first step block is published by Mindy, either she gets the rewards or everything is refunded to the problem owner. Hence, Manny cannot steal the solution.
- *Replication Attack.* Another type of attack between miners is as follows: when Mindy publishes her first step block, Manny can copy the block and replace Mindy's public key with his own. Then, part of the network would add Manny's block to their local copy of the blockchain, while the rest will add Mindy's. Then, when Mindy publishes her solution in the second step, Manny can copy the same solution and publish it as his own. Hence, at least part of the network would consider Manny's block as valid and if the chain containing Manny's block eventually happens to get longer, the attack has succeeded. However, this attack is not rational and is avoided by the fact that Mindy's public key is included in the hash value h. Hence, Manny cannot claim the solution for himself and get back his deposit and rewards, i.e. his second step transaction fails the hash check. Therefore, such an attack would have the same consequences as falsely claiming a solution.

Given the discussion above, Hybrid Mining satisfies requirement (2.b).

Resilience to Spamming. A problem owner has to pay a problem proposal fee, a storage fee, a transaction fee and a reward for every problem he proposes. Therefore, proposing long spam problems is costly and not rational. Similarly, proposing a problem and then solving it on your own leads to a loss of the f_1 units of proposal fee. Hence, one would not rationally propose a problem that is already solved. Therefore, the system is safe against such attacks. Hence, Hybrid Mining satisfies requirement (2.c).

Given the discussion above, we have the following proposition:

Proposition 2. *Hybrid Mining satisfies all the security requirements listed in Section 2, i.e. it is resilient to attacks such as double spending, spamming, depriving a miner from her reward or stealing it, and other dishonest behavior mentioned above.*

5 EXPERIMENTAL RESULTS

We implemented our approach in Java, and simulated both Hybrid Mining and classic Hashcash Mining for comparison.

Experiment Details. For classic Hashcash mining, we simulated a blockchain network with 10 miners. In our experiment, in order to speed up the simulation, a new Hashcash block is added roughly every 10 seconds, i.e. $m = 1/6$. Similarly, for Hybrid Mining, we simulated a network consisting of 10 miners and one problem proposer. The miners randomly choose to attempt either Hashcash or a problem that has not been claimed by another miner. For problems, we use a family of industrial benchmarks from the SAT 2009 competition, which originated from real-world computational biology problems [27]. For mining by problem-solving, the simulated miners run an extremely efficient SAT solver called SAT4J [26].

Experiment Machine. The experiment was conducted on an Intel Xeon E5-1650 (12 Threads, 3.8 GHz, 12 MB Cache) machine running 64-bit Debian 8. The machine had a total memory of 128 GB and each network node was given a single thread.

Experimental Results. In our experiment, using Hybrid Mining increases the block mining rate by 87%, in comparison with classic Hashcash, without changing the computational power or energy consumption of the network. Figure 7 shows the number of blocks mined in each approach as time goes by. Hence, Hybrid Mining not only uses the computational power of blockchain for solving useful problems, but also helps the blockchain become more scalable. In practice, when this approach is used in a real-world blockchain, the increase in scalability would be dependent on the relative hardness of the submitted problems and Hashcash, and also the rate at which new problems are submitted to the network.

6 CONCLUSION

In this paper, we introduced Hybrid Mining: a new mining protocol that can exploit the vast computational power of the blockchain network for solving real-world useful problem instances by incentivizing the miners to solve them, while remaining compatible with the traditional Hashcash method of mining. We implemented our

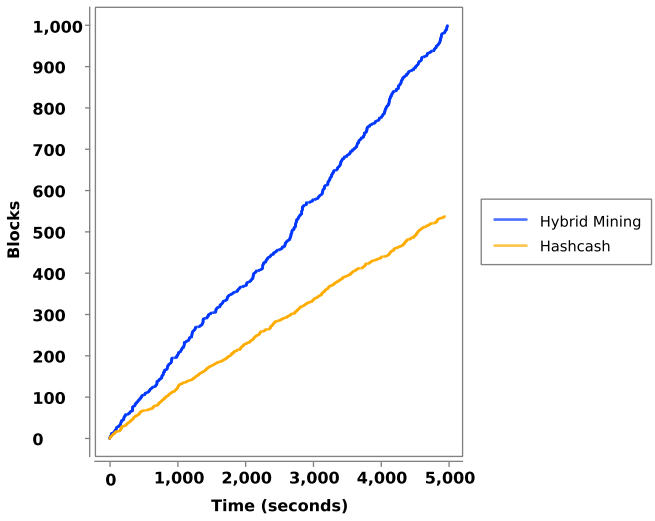

Figure 7: Scalability comparison of Hybrid Mining and Hashcash (using the same computational resources). The x axis is the time, and the y axis is the number of mined blocks.

approach and simulated it, using problems from computational biology, showing that in addition to usefulness in DPS, Hybrid Mining can also make blockchains more scalable. We also analyzed the security of the new protocol and showed that it is at least as resilient to double spending as Bitcoin, while also providing several other security guarantees.

A natural direction of future work would be to deploy this protocol in a large scale by creating a new cryptocurrency. This can lead to a much higher total computational power than what was available to us for the experiments, and can be used for solving harder and more important practical problems, specifically in computational biology. An example of such problems is protein folding.

ACKNOWLEDGMENTS

The research was partially supported by Vienna Science and Technology Fund (WWTF) Project ICT15-003, Austrian Science Fund (FWF) NFN Grant No S11407-N23 (RiSE/SHiNE), ERC Starting Grant (279307: Graph Games), and the IBM PhD Fellowship program.

REFERENCES

[1] C Adam-Bourdarios, D Cameron, A Filipčič, E Lancon, Wenjing Wu, ATLAS Collaboration, et al. 2015. ATLAS@ Home: harnessing volunteer computing for HEP. In *Journal of Physics: Conference Series*, Vol. 664. IOP Publishing, 022009.
[2] Sanjeev Arora and Boaz Barak. 2009. *Computational complexity: a modern approach*. Cambridge University Press.
[3] Adam Back. 1997. Hashcash. http://hashcash.org/.
[4] Adam Back et al. 2007. Hashcash – a denial of service counter-measure, 2002.
[5] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. 2017. Proofs of Useful Work. *IACR Cryptology ePrint Archive* 2017 (2017), 203.
[6] Bank for International Settlements. 2018. *Cryptocurrencies: looking beyond the hype*. Technical Report. Bank for International Settlements. https://www.bis.org/publ/arpdf/ar2018e5.pdf
[7] Krishnendu Chatterjee, Amir Kafshdar Goharshady, Rasmus Ibsen-Jensen, and Yaron Velner. 2018. Ergodic Mean-Payoff Games for the Analysis of Attacks in Crypto-Currencies. In *CONCUR 2018*. 11:1–11:17.
[8] Koen Claessen, Niklas Een, Mary Sheeran, and Niklas Sorensson. 2008. SAT-solving in practice. In *International Workshop on Discrete Event Systems*. IEEE, 61–67.
[9] Reuven Cohen. 2013. Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers Combined. In *Forbes (23 Nov 2013)*.
[10] CoinMarketCap. 2018. Cryptocurrency Market Capitalizations. https://coinmarketcap.com/.
[11] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756.
[12] Alex de Vries. 2018. Bitcoin's Growing Energy Problem. *Joule* 2, 5 (2018), 801–805.
[13] Digiconomist.net. 2018. Bitcoin Energy Consumption Index. https://digiconomist.net/bitcoin-energy-consumption.
[14] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2015. Proofs of space. In *Annual Cryptology Conference*. Springer, 585–605.
[15] Pasquale Forte, Diego Romano, and Giovanni Schmid. 2015. Beyond Bitcoin-Part I: A critical look at blockchain-based systems. *IACR Cryptology ePrint Archive* 2015 (2015), 1164.
[16] Pasquale Forte, Diego Romano, and Giovanni Schmid. 2016. Beyond Bitcoin-Part II: Blockchain-based systems without mining. *IACR Cryptology ePrint Archive* 2016 (2016), 747.
[17] Spyros Foteinis. 2018. Bitcoin's alarming carbon footprint. *Nature* 554, 7691 (2018), 169–169.
[18] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 281–310.
[19] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 51–68.
[20] Richard M Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer, 85–103.
[21] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, 357–388.
[22] Sunny King. 2013. Primecoin: Cryptocurrency with prime number proof-of-work. https://bravenewcoin.com/assets/Whitepapers/primecoin-paper.pdf.
[23] Sunny King. 2014. Gapcoin. http://gapcoin.org.
[24] Eric Korpela, Dan Werthimer, David Anderson, Jeff Cobb, and Matt Lebofsky. 2001. SETI@ HOME–massively distributed computing for SETI. *Computing in science & engineering* 3, 1 (2001), 78–83.
[25] Stefan M Larson, Christopher D Snow, Michael Shirts, and Vijay S Pande. 2009. Folding@ Home and Genome@ Home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866* (2009).
[26] Daniel Le Berre and Anne Parrain. 2010. The SAT4J library, system description. *Journal on Satisfiability, Boolean Modeling and Computation* 7 (2010), 59–64.
[27] Daniel Le Berre and Olivier Roussel. 2009. International SAT Competition. http://www.satcompetition.org/2009/.
[28] Angelique Faye Loe and Elizabeth A Quaglia. 2018. Conquering Generals: an NP-Hard Proof of Useful Work. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*. ACM, 54–59.
[29] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system.
[30] Karl J O'Dwyer and David Malone. 2014. Bitcoin mining and its energy footprint. In *25th IET Irish Signals & Systems Conference*.
[31] Carlos G Oliver, Alessandro Ricottone, and Pericles Philippopoulos. 2017. Proposal for a fully decentralized blockchain and proof-of-work algorithm for solving NP-complete problems. *arXiv preprint arXiv:1708.09419* (2017).
[32] Andrew Poelstra et al. 2014. Distributed consensus from proof of stake is impossible. (2014).
[33] Meni Rosenfeld. 2011. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980* (2011).
[34] Peter Smith et al. 2018. Hash Rate: The estimated number of tera hashes per second (trillions of hashes per second) the Bitcoin network is performing. https://www.blockchain.com/charts/hash-rate?scale=1&timespan=all.
[35] Michael Bedford Taylor. 2017. The evolution of bitcoin hardware. *Computer* 9 (2017), 58–66.
[36] Quandl Team. 2018. Quandl: Financial, Economic and Alternative Data. https://www.quandl.com/data/BCHAIN/HRATE-Bitcoin-Hash-Rate.
[37] 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.
[38] Fabian Vogelsteller, Vitalik Buterin, et al. 2014. Ethereum whitepaper.
[39] Harald Vranken. 2017. Sustainability of bitcoin and blockchains. *Current opinion in environmental sustainability* 28 (2017), 1–9.
[40] George Woltman. 1996. GIMPS: the Great Internet Mersenne Prime Search. https://www.mersenne.org/.