On the insecurity of quantum Bitcoin mining

Or Sattath

Published online: 14 March 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract
Grover’s algorithm confers on quantum computers a quadratic advantage over classical computers for searching in an arbitrary data set, a scenario that describes Bitcoin mining. It has previously been argued that the only side effect of quantum mining would be an increased difficulty. In this work, we argue that a crucial argument in the analysis of Bitcoin security breaks down when quantum mining is performed. Classically, a Bitcoin fork occurs rarely, i.e., when two miners find a block almost simultaneously, due to propagation time effects. The situation differs dramatically when quantum miners use Grover’s algorithm, which repeatedly applies a procedure called a Grover iteration. The chances of finding a block grow quadratically with the number of Grover iterations applied. Crucially, a miner does not have to choose how many iterations to apply in advance. Suppose Alice receives Bob’s new block. To maximize her revenue, she should stop and measure her state immediately in the hopes that her block (rather than Bob’s) will become part of the longest chain. The strong correlation between the miners’ actions and the fact that they all measure their states at the same time may lead to more forks—which is known to be a security risk for Bitcoin. We propose a mechanism that, we conjecture, will prevent this form of quantum mining, thereby circumventing the high rate of forks.

Keywords  Bitcoin · Cryptocurrencies · Proof of work · Quantum computing · Quantum cryptography · Grover’s algorithm · Security · Post-quantum cryptography

1 Introduction
Bitcoin is an electronic payment system [20] with a prospective aspiration by its supporters of becoming a dominant form of payment. Likewise, many hope that quantum computers will become an important form of computation. Can these two developments proceed hand-in-hand? Specifically, do quantum computers pose a fundamental threat to Bitcoin’s success?

Previous studies that addressed this concern argued that there are no fundamental risks to Bitcoin due to quantum computers [1,6,35]. Their argument is implicitly based on the following false logic: Bitcoin uses two cryptographic primitives, digital signatures and hash functions, both of which have post-quantum variants, i.e., classical digital signatures and hash functions that are secure against quantum adversaries. An upgraded version of this protocol, which will use post-quantum cryptography, will be secure against quantum adversaries. In this work, we argue the opposite: A crucial part of Bitcoin’s security argument is rendered invalid once Bitcoin miners begin to primarily use quantum computers, even though the cryptographic primitives on which Bitcoin relies on are not broken. This is another instance of what Dominique Unruh calls the post-quantum fallacy [37].

The gist of the argument is as follows. Classically, the times at which miner A and miner B find a block are completely uncorrelated: Both continuously try to find a block, and the probability of success for each attempt by either miner is the same and independent of the other miners’ actions. This property is tightly related to the fact that proof-of-work is progress free [5]. In the quantum setting, this is not the case. Grover’s algorithm provides a quadratic advantage to miners, and therefore, the chances of finding a block grow quadratically with the number of Grover iterations (where each iteration takes some fixed time). Suppose Alice devoted 2 min to applying Grover’s algorithm, and now she receives a new block, mined by Bob. Although she could discard her computation and start mining on top of Bob’s block, that course of action would effectively waste 2 min of computational resources. Instead, she could immediately stop Grover’s algorithm and measure her quantum state. If she...
is lucky and her block is valid, and she also propagates her block to most other miners before Bob does, these other miners will mine on top of her block, and she, rather than Bob, will get the block reward. Therefore, it is more profitable for Alice to use the second strategy. We call this second strategy aggressive.

Importantly, as soon as one miner finds a block, all other miners will also measure their quantum states at roughly the same time. In other words, there is a strong correlation between the times at which the different miners measure their respective states and, therefore, the times at which blocks are found. This correlation, which only happens in the quantum setting due to the aforementioned aggressive strategy, leads to a higher fork rate (called the stale rate), which, in the classical setting, is a known security risk.

As a countermeasure, we propose to change the default behavior of miners. Currently, in the event of a tie between the longest chains, a miner mines on top of the tip which minimizes the time that the block was received. We propose to add a term to the tie-breaking rule so as to penalize the aggressive strategy. Our proposed default behavior for the miners is to mine on top of the block, which minimizes the time in which the block was received plus the absolute value of the difference between the time the block was received and its timestamp.

Greater detail can be found in the paragraph that immediately precedes Eq. (2). The timestamp must be chosen in advance—before the Grover iterations are applied. From the aggressive miner’s perspective, the time at which the competing block will be mined is highly uncertain. Therefore, the timestamps of a block that was mined in response to receiving a competitor’s block will not be close to that of the timestamp in its own block. The addition of a term in the default mining behavior will confer a significant penalty for any block created using this aggressive strategy and therefore prohibits the aggressive strategy usefulness.

1.1 Structure

First, we provide a general explanation of the cryptographic primitives used in Bitcoin and how the Bitcoin network works. Next the 51% attack, the stale rate, and the risks associated with a high stale-rate, are discussed. Readers who are familiar with Bitcoin can safely skip this general introduction. We then provide a basic description of the properties of Grover’s algorithm and its relevance to Bitcoin mining. We do not explain how Grover’s algorithm works, and no prior knowledge in quantum computing is required to follow the text. We then show that the stale rate increases dramatically when miners use a natural quantum strategy. Next, we present our countermeasure to prevent the high stale rate and follow this with our conclusions. Appendix A discusses other interesting quantum mining effects that have no implications for the Bitcoin network’s security, and Appendix B discusses another attempt at devising a countermeasure, but in this case, it was found to have a vulnerability.

2 Preliminaries

2.1 The cryptographic building blocks used in Bitcoin

Bitcoin uses two cryptographic primitives, the first of which is proof of work [7]. An ideal cryptographic hash function is modeled as a random function \( H : \{0, 1\}^* \mapsto \{0, 1\}^n \) (this is known as the random oracle model). The function used in Bitcoin (see, e.g., [22]) is \( H(x) = \text{SHA256}(\text{SHA256}(x)) \), for which \( n = 256 \). The challenge in a proof of work is as follows. Given a message \( m \) and a target \( k \), find an \( x \) such that \( H(m, x) \leq k \). Because we model \( H \) as a random function, the optimal classical algorithm to find \( x \) is by brute force.

The second cryptographic primitive is a digital signature scheme, which is less relevant to our discussion. For a formal definition of digital signatures, see [12,13,16]. A signer who wants to sign messages creates a secret and a public key pair, using a polynomial key-generation algorithm, \((sk, pk) \leftarrow \text{key-gen}(\lambda)\), where \( \lambda \) is the security parameter, and the signer publishes \( pk \). Using the secret key, the signer can generate a signature \( \sigma \) for a document \( \alpha \) by running the signing algorithm: \( \sigma \leftarrow \text{sign}_{sk}(\alpha) \). A signature thus generated passes verification, that is,

\[
\Pr(\text{verify}_pk(\alpha, \sigma) = \text{accept}) = 1.
\]

Note that verification can be done by anyone by using the public key of the signer. A forger cannot generate a valid signature in time polynomial in the security parameter \( \lambda \). In the chosen message attack, this is formalized by the following security game. The forger has access to a signing oracle \( \text{sign}_{sk}(\cdot) \), and the forger wins the security game if he can find a fresh document (i.e., a document that was not given to the signing oracle) and a signature that passes the verification. A signature scheme is unforgeable under a chosen message attack if for every polynomial adversary, the winning probability in this game is negligible, i.e., vanishes faster than \( 1/\text{poly}(\lambda) \).

The current digital signature scheme used in Bitcoin, the Elliptic Curve Digital Signature Algorithm (ECDSA), is not secure against quantum polynomial forgers [25, 28]. However, digital signature schemes that are designed (and therefore, conjectured) to be secure against quantum forgers—often called post-quantum digital signature schemes—already exist. Accordingly, the Bitcoin network should transition to a post-quantum scheme before quantum computers become developed enough to perform such
attacks [6,35]. Concrete estimates of the time in years before quantum computers will be capable of breaking the elliptical curve signature scheme used by Bitcoin were calculated by Aggarwal et al. [1].

2.2 A simplistic overview of the Bitcoin protocol

Every Bitcoin user creates a digital signature key-pair. Suppose Alice has 5 bitcoins that are already assigned to her secret key $sk_A$, and she wants to send 1 bitcoin to Bob. First, she asks for Bob’s Bitcoin address, which is his public key $pk_B$. To send 1 bitcoin to Bob, she creates a message $m = \text{“I wish to send 1 bitcoin to the address } pk_B\text{”}$ and publishes the transaction $tx = (m, \sigma)$, where $\sigma \leftarrow \text{sign}_{sk_A}(m)$. Alice’s message, by itself, does not guarantee that Bob will receive the bitcoin: She could create a contradicting message, called a double-spend, where she also sends all 5 of her bitcoins to Charlie.

To finalize transactions, the Bitcoin protocol provides incentives to miners—special nodes in the bitcoin network that use dedicated hardware to solve the proof of work—to create a valid block. If a miner manages to extend the longest (valid) block-chain, she gets a reward of $r$ bitcoins. A block consists of transactions, a hash of the previous block, the Bitcoin address of the miner who mined it, the timestamp (which will play an important role later) and a nonce $x$. A block is considered valid if (i) the hash of the block is at most $k$, where $k$ is determined by the blocks that preceded it, and (ii) the transactions are valid, i.e., the signatures pass the verification, and there are no double-spend.

Since each block contains a pointer to a parent, the structure of the block-chain is a (directed) tree, the root of which is called the genesis block. Honest miners try to mine on-top of the tip of the longest block-chain. But what happens if there are two (or more) longest block-chains? The honest (default) behavior is to mine on top of the tip they received first. Currently, these ties occur rarely, and the longest chain rule breaks these ties efficiently, and so, effectively, the graph is a chain (hence the name block-chain), with rare exceptions of short forks.

A miner tries to find a (valid) block by using the proof of work mechanism described above—the miner increments the nonce $x$ until she finds a valid block that satisfies $H(B, x) < k$. The target $k$ is adjusted by some mechanism (that we do not describe) to obtain that a block is mined every 10 min in expectation. Currently, the timestamp is part of the block mainly for this adjustment. A user (e.g., a merchant) will consider a received payment as finalized if the transaction in which the bitcoins were transferred is “buried” deeply enough inside the longest block-chain. To that end, 6 blocks built on top of the transaction is considered a high level of security.

2.3 The 51% attack

A dishonest miner who controls a fraction of the network can attempt to double-spend. Consider the simple attack in which a miner sends one of its coins to a merchant, and after the merchant sends the miner the goods in return, the miner starts mining on top of the first block that preceded the transaction in which those bitcoins were spent. The attack is considered successful if the longest chain becomes the side of the fork that does not contain the first transaction. The success probability of this attack diminishes exponentially fast as a function of the number of confirmations as long as $q$, the relative hash rate of the attacker, satisfies $q < \frac{1}{2}$. A 51% attack refers to the case where $q > \frac{1}{2}$, and here, the dishonest miner will eventually succeed with a probability of 1. This is one of the main reasons why it is important to maintain Bitcoin mining decentralized.

2.4 Stale blocks

There are also forks that occur naturally due to network effects, i.e., when two miners mine blocks at almost the same time. We call blocks that are outside the longest chain stale blocks. We define the stale rate $p_{\text{stale}}$ to be the ratio of the number of blocks outside the longest chain to the number of all blocks. The condition $\tau_{\text{prop}} \ll \tau_{\text{block}}$ is crucial to keep the stale rate small, where the propagation time $\tau_{\text{prop}}$ is the time it takes for a block to reach all the other miners after it is mined, and $\tau_{\text{block}}$ is the average time between blocks.

Decker and Wattenhofer [8] empirically observed a stale rate of 1.69% in 2013, and, more recently, Stifter et al. [31] used an improved technique (based on data from merged-mined cryptocurrencies) and estimated that the stale rate between 07/2016 and 07/2018 has dropped to $p_{\text{stale}} \approx 0.24\%$.

2.5 Concerns with a high stale rate

Two main concerns associated with a high stale rate were studied in the classical setting. The first is the increased security risk due to a 51% attack. A miner with less than 50% of the hashing power can double-spend by exploiting the 51%

---

1 We only describe features of the protocol that are relevant to our work. This description does not faithfully represent how Bitcoin actually works. Refer to Ref. [22] for a textbook that fully covers the subject. More concise surveys can be found in Refs. [36,38].

2 The reward $r$, originally set to 50 bitcoins, is cut in half every 4 years.

3 These blocks are sometimes called orphaned blocks, a term that we feel is confusing: Although these blocks do not have descendants, every block, by definition, has a parent indicated by the block to which it is pointing.
attack described above. Stale blocks are created due to propagation time effects. The attacker can put all of her miners in the same physical location and decrease the propagation time to essentially zero on her own competing chain (it is fairly trivial to arrange a high-bandwidth and low-latency condition, such as in standard data centers). The condition of this attack being successful (with probability 1) for a miner with a $q$ fraction of the total hashing power is given by:

$$q > (1 - p_{\text{stale}})(1 - q).$$

Here, the left hand side is the relative hashing power of the attacker, which is all translated to a chain without any forks. The right hand side is the effective hashing power of the competing (honest) side: Though it has a relative hashing power of $1 - q$, a $p_{\text{stale}}$ fraction of it constitutes stale blocks, and therefore, they do not contribute to the longest chain. Solving Eq. (1) gives the condition:

$$q > \frac{1 - p_{\text{stale}}}{2 - p_{\text{stale}}}.$$ 

For example, if the stale rate is $\frac{1}{2}$, a miner with more than 40% (instead of 50%) of the total hashing power can double-spend using this attack. By plugging in the recent observed stale rate by Stifter et al. [31] (i.e., 0.24%) gives the condition $q \gtrsim 0.4994$. This security risk is discussed both theoretically and empirically in more detail in [8,31].

The second issue associated with a high stale rate and studied in the classical setting is fairness: An honest but large miner is systematically rewarded more than her fair share of the total mining revenue (both from the block reward and the fees). Fairness indirectly affects security: Unfairness contributes to centralization, and it is considered easier to attack a small group of miners than it is to attack a lot of them (for example, DOS attacks or regulatory changes in few countries) and for the miners to execute a 51% attack. These concerns are partially addressed in several works [9,19,29,33,34]. The author is not aware of any PoW-based solution that claims to simultaneously address fairness (of the block rewards and the revenue from fees) and security.

In the classical versus the quantum settings, stale blocks emerge for different reasons. In the quantum case, the effects that the stale rate has on security and fairness have not been elucidated. Nonetheless, we propose a mechanism that we conjecture resolves the high stale rate that may occur as a result of quantum mining. It is not clear, however, whether other concerns also need to be addressed in the quantum setting or whether the countermeasure for the high stale rate also addresses the concerns mentioned above regarding the insecurity and unfairness.

### 2.6 Grover’s algorithm

Consider an arbitrary function $f : \{1, \ldots, N\} \mapsto \{0, 1\}$. We call an item that is mapped to 1 by the function $f$ a marked item. If $f$ has $K$ marked items, it takes $\Omega(\sqrt{N})$ queries to $f$ to find a marked item with probability of at least $\frac{2}{3}$ using a classical random algorithm. Grover’s algorithm is a quantum algorithm that finds a marked item with a probability of at least $\frac{2}{3}$ using only $O\left(\frac{\sqrt{N}}{K}\right)$ queries [14] (see also the standard textbook [23]). Furthermore, $K$ need not be known in advance [4]. Grover’s algorithm is known to be optimal [3].

The above paragraph demonstrates the standard method of presenting Grover’s algorithm. Here, we offer a slightly stronger formulation. Suppose we allow only $Q$ queries to the function $f$. Crucially for this work, $Q$ may not be known to the algorithm in advance. The probability of finding a marked item using a classical randomized algorithm is $O\left(\frac{QK}{N}\right)$, whereas the success probability of Grover’s algorithm is $\Theta\left(\frac{QK}{N}\right)$ for $Q = O\left(\frac{\sqrt{N}}{K}\right)$. Grover’s algorithm applies a series of identical iterations and can be stopped after any number of iterations. The success probability of finding a marked element when the final state is measured (in the standard basis) is as given above. We emphasize that in the quantum setting, if the outcome is negative, the post-measurement state is of no use (due to the collapse of the quantum state), and one has to start from the beginning to find a marked element.

### 3 Risks of quantum Bitcoin mining

#### 3.1 Quantum mining

A miner that uses a quantum computer can use Grover’s quantum algorithm for the function

$$f(x) = \begin{cases} 1, & \text{if } H(B, x) \leq k \\ 0, & \text{otherwise.} \end{cases}$$

Consider a miner who receives a new block. In the classical setting, the miner already knows that the work that was invested during the time that elapsed between the previous block and the current block was essentially wasted. In the quantum setting, the miner has likely already applied several Grover iterations but has not measured the state. Though the miner could discard that computation, that action would be a waste of resources. Instead, the miner could immediately measure the state and test whether the computation was successful, following a strategy that we refer to as an aggressive quantum mining strategy (AQMS—should be pronounced a-qu-mess as in “a quantum mess”). In contrast, a peaceful
quantum mining strategy (PQMS) is one in which the state is discarded after receiving a valid block. These two mining strategies are described in more detail in Algorithm 1, where the difference can be seen in line 11. We use these terms since an aggressive strategy results in more forks that, in turn, lead to a higher stale rate.

Algorithm 1 Aggressive versus Peaceful Quantum Mining Strategies
1: procedure QUANTUM_MINING_STRATEGY 
2: \( B \leftarrow \) the tip of the unique longest block-chain, and \( k \leftarrow \) the current target 
3: \( \text{loop} \)
4: \( \text{Create a candidate template for a block } B_{\text{mine}} \text{ without a nonce, with the parent } B \)
5: \( \text{Define } f(x) = \begin{cases} 1, & \text{if } H(B_{\text{mine}}, x) \leq k \\ 0, & \text{otherwise.} \end{cases} \)
6: \( \text{Sample } Q \text{ according to some distribution} \)
7: for \( i = 1, \ldots, Q \) do 
8: \( \text{Apply 1 Grover iteration, with respect to the function } f \)
9: \( \text{if a new block } B_{\text{other}}, \text{which is the tip of the unique longest block-chain, is received then} \)
10: \( \text{set } B \leftarrow B_{\text{other}} \)
11: \( \text{if aggressive then} \)
12: \( \text{goto line 20} \)
13: \( \text{else if peaceful then} \)
14: \( \text{goto line 4} \)
15: \( \text{end if} \)
16: \( \text{end if} \)
17: \( \text{end for} \)
18: \( \text{Terminate Grover’s algorithm.} \)
19: \( \text{if Grover’s algorithm terminated with a successful output } x \) (i.e., an output \( x \) for which \( f(x) = 1 \), or alternatively, \( H(B_{\text{mine}}, x) \leq k \)) then 
20: \( \text{Set } B \leftarrow (B_{\text{mine}}, x) \)
21: \( \text{Propagate } B \text{ to all neighbors} \)
22: \( \text{end if} \)
23: \( \text{end loop} \)
24: \( \text{end procedure} \)

The analogous aggressive and peaceful classical mining strategies are shown in Algorithm 2. Classically, the difference between aggressive and peaceful strategies in Algorithm 2 only affects the result of one hash per miner per block. Insofar as the current speed of a typical mining device is 10 tera-hashes per second, this distinction between the aggressive and peaceful strategies is too negligible to have a noticeable effect.

3.2 Analysis of the stale rate in a simplistic model

We now provide a simplistic model with which we can analyze the stale rate. Suppose there are \( n \) symmetric miners (i.e., each miner has exactly the same hardware, software, network facilities, etc.) that are all interconnected with each other (the topology is a clique). Furthermore, we assume that the network is fully synchronous, and we ignore network effects.

Algorithm 2 Aggressive versus Peaceful Classical Mining Strategies
1: procedure CLASSICAL_MINING_STRATEGY 
2: \( B \leftarrow \) the tip of the unique longest block-chain, and \( k \leftarrow \) the current target 
3: \( \text{loop} \)
4: \( \text{Create a candidate template for a block } B_{\text{mine}} \text{ without a nonce, with the parent } B \)
5: \( \text{Define } f(x) = \begin{cases} 1, & \text{if } H(B_{\text{mine}}, x) \leq k \\ 0, & \text{otherwise.} \end{cases} \)
6: \( \text{Sample } Q \text{ according to some distribution} \)
7: for \( i = 1, \ldots, Q \) do 
8: \( \text{Apply 1 Grover iteration, with respect to the function } f \)
9: \( \text{if a new block } B_{\text{other}}, \text{which is the tip of the unique longest block-chain, is received then} \)
10: \( \text{set } B \leftarrow B_{\text{other}} \)
11: \( \text{if aggressive then} \)
12: \( \text{goto line 18} \)
13: \( \text{else if peaceful then} \)
14: \( \text{goto line 4} \)
15: \( \text{end if} \)
16: \( \text{end if} \)
17: \( \text{end for} \)
18: \( \text{Terminate Grover’s algorithm.} \)
19: \( \text{if Grover’s algorithm terminated with a successful output } x \) (i.e., an output \( x \) for which \( f(x) = 1 \), or alternatively, \( H(B_{\text{mine}}, x) \leq k \)) then 
20: \( \text{if } y = 1 \) (i.e. we found a valid block with a nonce \( x \)) then 
21: \( \text{Set } B \leftarrow (B_{\text{mine}}, x) \)
22: \( \text{Propagate } B \text{ to all neighbors} \)
23: \( \text{end if} \)
24: \( \text{end loop} \)
25: \( \text{end procedure} \)

Note that in a fully synchronous setting with clique topology, there is no point in using an aggressive strategy, since by the time you hear about the other block, all the other nodes have already received it, and no other miner will mine on top of it because they would receive it only after the original block (recall the tie-breaking rule). Therefore, we can assume that in both the classical and the quantum settings, all miners use a peaceful strategy.

In the classical setting, let \( h \) be the hashes-per-minute that every miner applies, and \( p_{\text{success}} = \frac{1}{h} \) be the probability that the value of a hash would be at most \( k \) and, therefore, a successful guess. Since a block is mined (roughly) every 10 min in expectation, we deduce that \( p_{\text{success}} \cdot h \cdot n \approx \frac{1}{10} \). Suppose one miner finds a block. The probability that another miner will find a block in exactly the same round (and hence, a fork will occur) is at most \( n \cdot p_{\text{success}} \approx \frac{1}{10} \approx 1.6 \cdot 10^{-3} \), which can be safely approximated to 0.

Suppose that in the quantum setting, all miners choose the same \( Q \) in Algorithm 1, line 6 deterministically. Let us denote by \( t \) the number of minutes it takes to apply these \( Q \) iterations (i.e., if the time it takes to perform one iteration in the for loop is \( t_{\text{iteration}} \), then \( t = Q \cdot t_{\text{iteration}} \)). Every \( t \) min-
Therefore, for $0 < t < 10$,

$$\lambda(t) = \ln\left(\frac{10}{10 - t}\right)$$

$\mathbb{E}[\# \text{ blocks added per minute}] = \lambda(t)/t$

The stale rate is:

$$p_{\text{stale}}(t) = \frac{\mathbb{E}[\# \text{ blocks outside the longest chain added per minute}]\vspace{3pt}}{\mathbb{E}[\# \text{ blocks added per minute}]} = 1 - \frac{\mathbb{E}[\# \text{ blocks added to the longest chain per minute}]\vspace{3pt}}{\mathbb{E}[\# \text{ blocks added per minute}]} = 1 - \frac{1/10}{\lambda(t)/t} = 1 - \frac{t}{10 \ln\left(\frac{10}{10 - t}\right)}$$

Figure 1 presents $p_{\text{stale}}(t)$.

For example, the stale rate is $\approx 5\%$ for $t = 1$ minute and $50\%$ for $t \approx 7.96$ min.

It may be the case that a deterministic choice of $Q$ is not an equilibrium and that only mixed equilibria strategies (i.e., distributions over $Q$) exist. Note that there are two contradicting forces at work here: by investing more time a miner can increase her chances of finding a valid nonce quadratically (due to the Grover speed-up). On the other hand, the more time the miner invests in a particular computation, the risk that someone else will find a block before she does also increases, in which case the time she spent was wasted as she will have to terminate before completion. In other words, investing less time increases the chances that her computation will not be discarded. We restate that in the fully synchronous setting, the AQMS has no advantage over the PQMS: By the time a miner hears about another miner’s block, all other miners have also received that block, and therefore, there is no chance of winning such a block race.

### 3.3 The stale rate in a realistic setting

In more realistic scenarios, where propagation time effects are taken into account (i.e., the non-synchronous setting), we will see that it is more profitable for quantum miners to use aggressive rather than peaceful strategies. Fix a miner $M$, which has a very small (but nonzero) fraction of the hash-power. Following Eyal and Sirer and others [10,24], we denote by $\gamma$ the ratio of the network (in terms of hash-power) that will mine on $M$’s block when another miner $N$ have released a block at (approximately) the same time, resulting in an equal length fork. This parameter, which is called the propagation factor (or the influence), also played an important role in prior works, see, for example, [10,24,32]. Suppose an aggressive miner $M$ is notified about a block by some other miner $N$. In response, $M$ measures his own block by utilizing the aggressive strategy, and suppose this measurement yields a valid block. At his point, there are two blocks that were released approximately the same time. The probability that $M$ will win the block race (and $N$ will lose the block reward) is $\gamma$: Recall that according to the existing tie-breaking rule, honest miners will mine on top of the block which they receive first, and in this case, a $\gamma$ fraction of the miners will first receive $M$’s block, and $1 - \gamma$ will first receive $N$’s block. Therefore, as long as $\gamma > 0$, this strategy performs better than the peaceful one (which would simply discard the quantum state and lose the chance of getting the block reward when being notified about another block).

---

4 Despite the fact that it plays such an important role, as far as the author is aware, there are no empirical data on the propagation factor $\gamma$. We speculate that it is because it is hard to measure $\gamma$, directly, or indirectly. A related notion—the propagation time—was studied and measured extensively in Ref. [8].
The main observation is that it is more profitable to use aggressive strategies in a realistic setting (i.e., in the non-synchronous model). All miners will therefore do so, and when one miner finds a block, the rest of the miners will immediately measure their states. As a result, there is strong correlation between the states at which valid blocks are found. This outcome is in sharp contrast to the classical behavior, where the distribution of finding a block by two miners is statistically independent. The conclusion that can be drawn where the distribution of finding a block by two miners is correlated between the times at which valid blocks are found.

This outcome is in sharp contrast to the classical behavior, where the distribution of finding a block by two miners is statistically independent. The conclusion that can be drawn where the distribution of finding a block by two miners is correlated between the times at which valid blocks are found.

The conjecture in Open Problem 1 means that there are no Nash equilibria when the players can choose any strategy from $\mathcal{P}$, and $N E_{P \cup A}$ is the set of all mixed Nash equilibria when the players can choose any strategy from $\mathcal{P} \cup A$.

In all of the open questions, we are interested in the Nash equilibrium in the setting where the propagation delay is finite but goes to 0, all miners are symmetric and quantum, and the number of players $n \to \infty$.

1. Characterize the Nash equilibria in $N E_{P \cup A}$. What are their stale rates? Specifically, are all stale rates $\Omega(1)$? We conjecture that the answer is yes.

2. Characterize the Nash equilibria in $N E_{P \cup A}$. What are their stale rates? Specifically, are all stale rates $o(1)$? We conjecture that the answer is yes.

The conjectures in Open Problem 1 mean that there are no low-stale-rate equilibria if AQMSs are allowed and that when only PQMSs are considered, there exists (where here we use Nash’s theorem [21]) a low-stale-rate equilibrium. Therefore, these conjectures suggest that one way to resolve the high stale rate is to change the Bitcoin protocol such that AQMSs are prohibited. In the next section, we discuss such a countermeasure.

In a recent paper that was published after the first version of this work, Ray, Lee and Santha [18] proved a weak variant of our second conjecture. They assume that miners only use the PQMS, an assumption that they justify by exploiting the countermeasure discussed below—see Sect. 4. They define a game that models some of the central aspects of quantum Bitcoin mining while ignoring others. They analyze two games, called a quantum race and a stingy quantum race. In the latter case, in the case of a fork, all parties loose. In the (non-stingy) quantum race, all the miners that generated the fork split the revenue from the block reward. For technical reasons, the stingy game is easier to analyze, yet, the non-stingy race presents a better model for Bitcoin. They show that when the stingy quantum race is in equilibrium, the probability of a fork goes to 0 as the hashing power increases—see [18, Theorem 27 and Theorem 8]. They also show that the same strategy is an approximate Nash equilibrium in the (non-stingy) quantum race. Since this is the same strategy, it shows that there exists an approximate Nash equilibrium in which the stale rate goes to 0. This is in line with our conjecture.

4 A countermeasure to the aggressive strategy

4.1 The new tie-breaking rule

To prohibit use of the AQMS, consider the following proposal. Every Bitcoin block already contains a timestamp. Currently, the timestamp is used mainly to calculate the new target $k$ every 2016 blocks ($\approx 2$ weeks). We propose to change the default tie-breaking rule. Suppose these latest blocks (each one is the tip of a longest chain) have timestamps $s_1, \ldots, s_n$, and they were first received at the times $t_1, \ldots, t_n$. For each tip, let $\Delta_t \equiv |s_i - t_i|$ and let the penalty be defined as $p_i \equiv t_i + \Delta_t$. The current default strategy is to mine “on-top” of the tip that was received first, i.e., the block that minimizes $t_i$ (see, for example, [22, Chapter 5.5]). Instead, we propose that the default strategy is to mine “on-top” of the tip that minimizes the penalty $p_i$, i.e., add the term $\Delta_t$ to the penalty calculation. To conclude, the default mining strategy in the event of ties between longest chains should be changed to mining on top of the tip that achieves:

$$\min_i p_i \equiv \min_i t_i + |s_i - t_i|$$

(2)

Note that this change does not require users to upgrade all—only miners would be asked to upgrade. Furthermore, this upgrade does not require a hard-fork and not even a soft-fork (as no damage occurs to non-upgrading miners). For more detail on these two types of forks, see [22, Chapter 3].

4.2 The honest mining strategy

An honest miner knows in advance how many Grover iterations it plans to apply, and under the (very plausible) assumption that she knows how long it will take to complete these iterations, she can set the timestamp to be that (future) time. For example, suppose Alice starts the Grover

---

5 Specifically, they ignore the fact that once Grover’s algorithm fails, the miner restarts and that after a Bitcoin block is found, the process is repeated—an aspect which we also ignore in our conjecture since the repeated game is probably much harder to analyze.
algorithm at 12:00:00, each Grover iteration takes $3 \times 10^{-7}$
seconds, and she plans running $10^9$ Grover iterations. In this
case, she will set the timestamp in her attempt to find a block
to 12:05:00. Suppose Bob receives Alice’s block at 12:05:03.
The penalty which Bob will calculate is 12:05:06. In general,
the tip of an honest miner will have a penalty $p_i$ that is $t_i + \Delta_i$,
where $\Delta_i$ will be the propagation time—which is an order of
seconds in practice [8,31].

4.3 Security arguments supporting our
countermeasure

We first show why the AQMS strategy will not be beneficial
according to the new tie-breaking rule. Consider the example
given above. Suppose another miner, Bob, starts mining
with Alice (at 12:00:00), but Bob uses the AQMS. Bob plans
to do $5 \times 10^9$ iterations, where each iteration takes the same
time as Alice. Therefore, Bob’s timestamp will be 12:08:20.
Recall that Alice finds a valid block at 12:05:00 with timestamp
12:05:00. Bob receive here block a few seconds later.
In case Bob measures after Alice and is lucky enough so that
he successfully creates a block, his penalty will be at least
12:08:20, where Alice’s penalty is much lower: in our example,
12:05:00 + $\Delta_A$ where $\Delta_A$ is Alice’s propagation time,
we assumed is a few seconds. This means Bob will have a
higher penalty and therefore will not be followed by other
miners using our new tie breaking rule in this example. Bob
could win the block race if he planned, in advance, to gen-
erate a block with a timestamp which is between 12:05:00
and 12:05:00 + $\Delta_A$.

We conjecture that there is high uncertainty regarding the
creation time of blocks, assuming our countermeasure is used. This is supported by the results in Ref. [18]—see the discussion at the end of Sect. 3. Therefore, we argue that the chances two miners will successfully create blocks with serial numbers $s_1$ and $s_2$ such that $s_1 - s_2 < \Delta$ (where $\Delta$ is the maximal propagation time of the too) is very small.
This is the only case where there is a difference between the
PQMS and AQMS. This intuition is formalized in the following conjecture:

Conjecture 2 (Informal) If our proposed countermeasure is deployed, the AQMS becomes ineffective: In the setting as in
Open Problem 1, PQMS with $t$ iterations weakly dominates
AQMS with $t$ iterations.

In conjunction with the second conjecture in Open Prob-
lem 1, these conjectures imply that the effective stale rate
will be $o(1)$.

6 The specific choice depends on the Nash equilibrium, which depends
on the parameters of the network, such as the number of miners and
their respective hash power.

We note that it is not clear whether the security risks that
were discussed due to a high stale rate, namely a reduced
threshold for 51% attacks and unfairness, are relevant in the
quantum mining setting. The reasons for stale blocks in the
classical and quantum settings (with or without the counter-
measure) are different: In the classical setting, a miner can
reduce its stale rate on its side of the chain by putting all of
its mining equipment in the same physical location. Yet, in
the quantum setting, a single miner might also suffer from a
high stale rate. Therefore, even if the conjecture above does
not hold, nevertheless, it is possible that neither fairness nor
security will be affected by quantum mining.

We leave the issue of whether security or fairness can be
compromised with our proposed countermeasure as an open
question. We emphasize that this is a very different question
than Conjecture 2 and that it is harder to formalize for various
reasons: In reality, miners are not restricted to the AQMS or
PQMS, but rather, they can use other strategies; indeed, their
goal may be to cause damage to others rather than to increase
their own utility (see the example in Appendix B); Bitcoin
mining is a repeated game, played for each block, and long
range attacks, such as selfish mining which is discussed in the
following paragraph should also be considered. We leave the
problem of modeling and addressing these concerns to future
work.

4.4 Selfish mining

Selfish mining is a strategy that allows a miner (or a mining
pool) to obtain more than its fair share [10], see also [24,32].

The intuition behind these attacks requires basic under-
standing of the Bitcoin protocol, but is essentially simple.
Bitcoin mining is a zero-sum game. By reducing the other
miners weight in the longest chain, a miner can increase its
own revenue. This is achieved in the following may. A miner
(especially a large one) who finds a block does not propagate
it to others. Suppose the miner is lucky again and now has an
advantage of two blocks. At this point, all other miners still
mine on top of the old block. Suppose a competing miner
now publishes a block. The selfish miner can then reveal his
two blocks that he mined and win the fork race.

Several proposals were made to mitigate selfish mining
attacks. Some proposals only change the tie-breaking rule—
[11,15,39], though they do so in a different way than ours.
Other mitigations change make more fundamental changes
to the validation rules, so these proposals are not backward
compatible—see Refs. [2,30].

The rationale behind these proposals is to disincentive
publications of the blocks which the selfish miner kept
secret—so that the overall probability of winning the race
would decrease. Note that these are necessarily old blocks—
and at the time these are mined the selfish miner has a lot
of uncertainty when they will be released. This is similar to
the challenge raised in this work: how can we rule out the AQMS strategy.

It would be convenient if there was a way to mitigate both the problems of selfish mining and the AQMS by using the same mechanism. None of the mitigation proposals cited above disqualifies the AQMS, and our tie-breaking rule is incompatible with theirs. Nevertheless, we are optimistic that our proposal is better than the current tie-breaking rule as a countermeasure to selfish mining, though probably it is not the best one.

More importantly, algorithms that evaluate modifications of the protocol (by finding the optimal attack) were developed and analyzed by Sapisrstein, Sompolinsky and Zohar [32] and Zhang and Preneel [39]. Naturally, the analysis in both works assumes that the miners are classical.

It is beyond the scope of this work to address the issue of selfish mining and its variants in the quantum setting.

4.5 Drawbacks

Note that Eq. (2) asks the miner to behave somewhat irrationally: A miner who began to run Grover’s algorithm on one tip is asked to move to another tip if the latter has a lower penalty. This means that the miner loses the computational power invested between the time she started and the time she moves to a new tip. Since the AQMS will accrue a high penalty on average, we expect this to be a rare event, similar to the stale rate today. Even if it occurs, the time spent mining the wrong tip is short (upper-bounded by the propagation time \( \Delta_t \), which is in the order of a second), so the damage incurred from moving to the other tip is relatively small. Based on the stale rate in 2017, this translates into potential damage of roughly one minute every year. This is in contrast to damage of following the PQMS without our proposed countermeasure, which amounts to minutes per block and sums up to several months per year: Without our countermeasure, the computational resources spent between the previous measurement and the time a block is received are discarded in the PQMS, without any justification.

Another drawback of our countermeasure is that it opens new attack vectors, such as those based on timing attacks (for example, attacks on time servers), since the strategy depends on having an accurate clock to calculate \( p_1, \ldots, p_n \). As mentioned before, the current Bitcoin protocol barely uses time, and therefore, the Bitcoin network is currently less prone to such attacks, even.

An alternative countermeasure approach would be to find a PoW mechanism for which quantum computers have no advantage over classical computers. Aggarwal et al. suggested an alternative proof-of-work function, called Momentum, for which the quantum advantage is claimed to be smaller than the quadratic advantage for double-SHA256 (which is the one used in Bitcoin) [1]. However, their construction is insufficient to completely resolve the security risk we discussed, since there is still a polynomial quantum speedup. Other approaches try to replace proof of work with an entirely different procedure—one notable attempt is called proof of stake, which is not prone to the vulnerability discussed in this work. More details on proof of stake can be found in [17] and references therein.

5 Conclusions

The security concern raised in this work is only a potential problem for the long term. Long term since the quantum computers predicted to be available at least until 2028 would not have enough qubits to run the described Grover’s algorithm—see [1]. The concern is only a potential threat, since we demonstrated that the classical security argument does not hold in the presence of quantum miners. Perhaps security and fairness can be shown by using a different, more elaborate argument or by deploying our proposed countermeasure.

The goal of this work was to establish several facts: (i) There is strong evidence to suggest that, under current Bitcoin rules, quantum mining would cause a high stale rate. (ii) We propose a simple countermeasure that prevents the high stale rate by prohibiting the AQMS. (iii) Unlike the classical setting, it is not clear what strategy should be suggested as the default (honest) behavior for quantum miners and, more specifically, how many Grover iterations they should apply with our countermeasure. A partial answer to this third question has already been given in [18].

We would also like to explain what was not shown and why. This work did not formalize quantum mining as a one-shot game or some other simple model and analyzed the Nash equilibria of quantum mining. Additionally, this work did not try to relate between the real world and such a model. To formalize such a model and argue about its properties, it is important to first establish and debate the three facts mentioned in the preceding paragraph to ensure that the model or game that is under analysis captures the desired properties.

The main issues that remain to be addressed in terms of Bitcoin and quantum mining are to further understand the equilibria strategies in the presence of quantum miners, beyond the results of [18] and to better understand how other aspects of Bitcoin, such as fairness in the contexts of pool mining [26,27], infiltration attacks [11] and selfish mining and its extensions [10,24], are affected by quantum mining. We comment that the algorithms that analyze the optimal attacks [32,39] also need to be the adapted so that they could take quantum mining into account.

\footnote{For this reason alone, there is no need to consider some sort of responsible disclosure.}
Acknowledgements The author is grateful for his fruitful discussions with Yotam Ashkenazi, Ittay Eyal, Robin Kothari, Troy Lee, Maharshi Ray, Yonatan Sompolinsky and Aviv Zohar. He would also like to thank Troy Lee for noticing a vulnerability in the countermeasure presented in an earlier version of this manuscript—see Appendix B.

Funding This research was supported by ERC Grant 280157, by the Israel Science Foundation (ISF) Grant Nos. 682/18 and 2137/19 and by the Cyber Security Research Center at Ben-Gurion University.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by the author.

A Subtle implications of quantum mining

Quantum mining has some subtle implications, which are discussed below.

A.1 Effects on the confirmation time

The confirmation time for a transaction is defined as the time it takes for the transaction to be included in a block after being broadcast to the network by the user. Classically, it takes a miner a fraction of a second for each attempt to solve a proof-of-work puzzle. Upon receiving a transaction, a miner can include the transaction in the next attempt to solve the proof-of-work puzzle. Therefore, a block typically contains the transactions paying with the highest fees (measured in bitcoin per byte) that fit into a block at the time that the block was mined. For a user who is willing to pay enough, under normal circumstances (i.e., assuming that the user has Internet connection, miners are rational, no denial-of-service attack, etc.), she can guarantee that her transaction will be confirmed in the next block by offering a high enough fee (for example, a higher fee rate than all others).

A quantum miner can only update her block after a full run of the Grover algorithm. This condition holds with or without our proposed selection rule. For example, if the number of Grover algorithm iterations by all miners is set to 2 min, a user who offers a high enough fee can only guarantee her inclusion in blocks created 2 min after the transaction is broadcast. In a more realistic scenario, where the number of Grover iterations is chosen according to some distribution, users who pay high enough fees can only be guaranteed inclusion in the next two blocks (rather than in the next block, which is the current state of affairs).

Apparently, the reduced ability to guarantee inclusion is relatively unimportant since the block creation process is already unpredictable: There are no guarantees regarding the time it will take for the next block to be mined (only the expected time is guaranteed). Quantum mining will only increases the unpredictability. More precisely, although classically, a user could guarantee that a transaction would get included in the next block, she could not guarantee the more important property, i.e., when it would get included. Because this property cannot be guaranteed, the loss of the guarantees on the less relevant property is of secondary importance.

A.2 Economy of mining equipment

Suppose there are two classical mining devices with hash rates of $x$ and $2x$. Other things being equal (such as electricity consumption), we would expect the second device to cost twice as much as the first, since classically, the revenue from Bitcoin mining is linear in the hash rate. For quantum devices, the quadratic speedup renders a different scenario: One would expect, at least naively, that the cost of the second device will be quadruple that of the first.

Were such a cost difference the case, quantum Bitcoin mining hardware manufacturers would be more strongly motivated to improve the hash rate (analogous to CPU speed), a scenario that may also exist in other markets affected by quantum speedups. This example is illuminating, since the connection between computational power and revenue is direct and can be calculated easily.

A.3 Finding the equilibrium and barrier of entry

The current strategy for classical miners is extremely simple: mine on top of the tip of the longest chain as fast as possible. It is plausible that in the PQMS equilibrium, the distribution over the number of Grover iterations $Q$ the miner should apply (see Algorithm 1, line 6) in a PQMS would depend on the properties of the other miners (most importantly, the number and hash rates of the mining devices in each pool). To see a concrete example of such a dependence, see [18]. This information may not be accessible to all miners, in which case equilibrium would not be achieved.

Outside equilibrium, miners with more information about the strategies of the other miners would realize higher profits. This may lead to a high barrier to entry, which does not exist now.

---

8 The size of a block used to be 1 MB and then the calculation was trivial. Recently, Bitcoin has upgraded to a Segregated Witness, but the property that a miner selects the transactions that maximize her revenue is still part of the protocol.
B An unsuccessful countermeasure

Equation (2) provides a new default tie-breaking rule as a countermeasure—a mechanism that we conjecture prevents the AQMS. In an earlier version of this manuscript, we provided a different, older rule that, it turns out, prevents the AQMS. But in so doing, it introduces a new vulnerability, which was discovered by Troy Lee. The goal of this Appendix is to explain the earlier countermeasure and the resulting vulnerability.

The old tie-breaking rule was as follows: Suppose the tips of the longest chains have timestamps $s_1, \ldots, s_n$, and they were first received at the times $t_1, \ldots, t_n$. Let $I_{\text{min}} = \min_{i \in [n]} t_i$ and let the penalty of each of these tips be defined as $p_i = s_{\text{min}} - s_i$. The (old) default strategy was to mine “on-top” of the block that had the lowest penalty $p_i$. Though the (old) tie-breaking rule seems to efficiently prevent the AQMS strategy, consider Mallory, the malicious miner who wants to harm Alice, an honest miner. We assume Mallory has an complete knowledge of the network and its properties, and we present a strategy that can be executed at no cost.

Consider the following example. Let us assume that a block sent by Alice is received by all the other miners after exactly 1 second and that Mallory is also aware of Alice’s activity. Mallory waits for Alice to create a valid block. Suppose that a block has a timestamp $s_A$ and that it was received by all the other miners at time $t_A = s_A + 1$. Mallory then starts mining a block with the timestamp $s_M = s_A + 1$. Suppose that Mallory finds a block after running Grover’s algorithm for 100 seconds. As her block will be received much later than Alice’s block—roughly, $t_A + 100$, and therefore, $I_{\text{min}}$ will remain $t_A$. Note that Mallory’s penalty in this case is $|s_{\text{min}} - s_M| = |s_{\text{min}} - s_A + 1| = 0$, whereas Alice’s penalty will be $|s_{\text{min}} - s_A| = 1$, and therefore, Mallory will minimize the penalty and win the fork race. As a result, other miners will start to mine on the top of Mallory’s block rather than on Alice’s block. In terms of costs—since all miners will mine the top of Mallory’s block if she succeeds—the strategy used by Mallory entails no risk to her, but it may cause Alice to incur significant damage. Insofar as Bitcoin mining is a zero-sum game, it may even be indirectly beneficial to use this strategy not just to attack others. We emphasize that Mallory does not need to be well connected to the other nodes. Rather, she needs to know Alice’s propagation time. We are not aware of similar classical or quantum attacks in the context of mining. This attack does not work if the default tie-breaking rule is as defined in Eq. (2). Note that in this example, $|s_M - t_M|$ is roughly 100 seconds, and basically, because of that, Mallory’s penalty will be higher than that of Alice. Therefore, this specific attack will not work.

References

1. Aggarwal, D., Brennen, G., Lee, T., Santha, M., Tomamichel, M.: Quantum attacks on bitcoin and how to protect against them. Ledger 3 (2018). arXiv:1710.10377
2. Bahack, L.: Theoretical Bitcoin attacks with less than half of the computational power (draft) (2013). arXiv:1312.7013
3. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.V.: Strengths and weaknesses of quantum computing. SIAM J. Comput. 26(5), 1510–1523 (1997). arXiv:quant-ph/9701001
4. Boyer, M., Brassard, G., Høyer, P., Tapp, A.: Tight bounds on quantum searching. Fortschr. Phys. 46(4–5), 493–505 (1998). arXiv:quant-ph/9605034
5. Biryukov, A., Khovratovich, D.: Equihash: asymmetric proof-of-work based on the generalized birthday problem. Ledger 2, 1–30 (2017)
6. Buterin, V.: Bitcoin is not quantum-safe, and how we can fix it when needed. https://bitcoinmagazine.com/articles/bitcoin-is-not-quantum-safe-and-how-we-can-fix-1375242150. http://www.webcitation.org/6wDiIPU3l (2013)
7. Dwork, C., Naor, M.: Pricing via processing or combatting junk mail. In: Proceedings of Advances in Cryptology—CRYPTO’92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16–20, 1992, pp. 139–147 (1992)
8. Decker, C., Wattenhofer, R.: Information propagation in the Bitcoin network. In: Proceedings of 13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013, Trento, Italy, September 9–11, 2013, pp. 1–10 (2013)
9. Eyal, I., Gencer, A.E., Sirer, E.G., van Renesse, R.: Bitcoin-NG: a scalable blockchain protocol. In: 13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, pp. 45–59 (2016). arXiv:1510.02037
10. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Financial Cryptography and Data Security—18th International Conference, FC 2014, Christ Church, Barbados, March 3–7, 2014, Revised Selected Papers, pp. 436–454 (2014). arXiv:1311.0243
11. Eyal, I.: The miner’s dilemma. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17–21, 2015, pp. 89–103 (2015). arXiv:1411.7099
12. Goldwasser, S., Micalli, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)
13. Goldreich, O.: The Foundations of Cryptography. Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
14. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22–24, 1996, pp. 212–219 (1996). arXiv:quant-ph/9605043
15. Heilman, E.: One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract). In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) Financial Cryptography and Data Security—FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers, volume 8438 of Lecture Notes in Computer Science. Springer, pp. 161–162 (2014). IACR Cryptogo ePrint Archive: report 2014/007
16. Katz, J., Lindell, Y.: Introduction to Modern Cryptography, 2nd edn. CRC Press, Boca Raton (2014)
17. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Advances in Cryptology—CRYPTO 2017, Part I, volume 10401 of Lecture Notes in Computer Science. Springer, pp. 357–388 (2017). Cryptology ePrint Archive: report 2016/689
18. Lee, T., Ray, M., Santha, M.: Strategies for quantum races. In: Blum, A. (ed.) 10th Innovations in Theoretical Computer Science Conference, ITCS 2019, volume 124 of LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 51:1–51:21 (2019). arXiv:1809.03671
19. Lewenberg, Y., Sompolinsky, Y., Zohar, A.: Inclusive block chain protocols. In: Financial Cryptography and Data Security—19th International Conference, FC 2015, San Juan, Puerto Rico, January 26–30, 2015, Revised Selected Papers, pp. 528–547 (2015)
20. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
21. Nash, J.F.: Equilibrium points in n-person games. Proc. Natl. Acad. Sci. 36(1), 48–49 (1950)
22. Narayanan, A., Bonneau, J., Felten, E.W., Miller, A., Goldfeder, S.: Bitcoin and Cryptocurrency Technologies—a Comprehensive Introduction. Princeton University Press, Princeton (2016)
23. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th edn. Cambridge University Press, New York (2011)
24. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: IEEE European Symposium on Security and Privacy, EuroS&P 2016, pp. 305–320 (2016). Cryptology ePrint Archive: report 2015/796
25. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.E.: Quantum resource estimates for computing elliptic curve discrete logarithms. In: Advances in Cryptology—ASIACRYPT 2017, Part II, pp. 241–270 (2017). arXiv:1706.06752
26. Rosenfeld, M.: Analysis of Bitcoin pooled mining reward systems (2011). arXiv:1112.4980
27. Schrijvers, O., Bonneau, J., Boneh, D., Roughgarden, T.: Incentive compatibility of Bitcoin mining pool reward functions. In: Financial Cryptography and Data Security—20th International Conference, FC 2016, Revised Selected Papers, volume 9603 of Lecture Notes in Computer Science. Springer, pp. 477–498 (2016)
28. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484–1509 (1997). arXiv:quant-ph/9508027
29. Sompolinsky, Y., Lewenberg, Y., Zohar, A.: SPECTRE: a fast and scalable cryptocurrency protocol. IACR Cryptology ePrint Archive, vol. 2016, p. 1159 (2016)
30. Solat, S., Potop-Butucaru, M.: Brief announcement: ZeroBlockd: timestamp-free prevention of block-withholding attack in Bitcoin. In: Lecture Notes in Computer Science. Springer International Publishing, pp. 356–360 (2017). arXiv:1605.02435
31. Stifter, N., Schindler, P., Judmayer, A., Zamyatin, A., Kern, A., Weippl, E.R.: Echoes of the past: recovering blockchain metrics from merged mining. IACR Cryptology ePrint Archive, vol. 2018, p. 1134 (2018)
32. Sapirstein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in Bitcoin. In: Grossklags, J., Preneel, B. (eds.) Financial Cryptography and Data Security, volume 9603 of Lecture Notes in Computer Science. Springer, pp. 515–532 (2016). arXiv:1507.06183
33. Sompolinsky, Y., Zohar, A.: Secure high-rate transaction processing in Bitcoin. In: Financial Cryptography and Data Security—19th International Conference, FC 2015, Revised Selected Papers, pp. 507–527 (2015). Cryptology ePrint Archive: report 2013/881
34. Sompolinsky, Y., Zohar, A.: PHANTOM: a scalable BlockDAG protocol. IACR Cryptology ePrint Archive, vol. 2018, p. 104 (2018)
35. Tessler, L., Byrnes, T.: Bitcoin and quantum computing (2017). arXiv:1711.04235
36. Tschorsch, F., Schueermann, B.: Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. 18(3), 2084–2123 (2016). Cryptology ePrint Archive: report 2015/464
37. Unruh, D.: Formal verification of quantum cryptography [Youtube video] (2016). https://kodu.ut.ee/~unruh/publications/2016-09-15%20-%20Verification%20of%20Quantum%20Cryptography%20-%20%20Quantum%20Invited%20talk.pptx
38. Zohar, A.: Bitcoin: under the hood. Commun. ACM 58(9), 104–113 (2015)
39. Zhang, R., Preneel, B.: Publish or perish: a backward-compatible defense against selfish mining in Bitcoin. In: Handschuh, H. (ed.) Topics in Cryptology—CT-RSA 2017—The Cryptographers’ Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings, volume 10159 of Lecture Notes in Computer Science. Springer, pp. 277–292 (2017)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.