LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

RESEARCH ARTICLE

Quantum Attacks on Bitcoin, and How to Protect Against Them

Divesh Aggarwal,*† Gavin Brennen,‡ Troy Lee,§ Miklos Santha,¶ Marco Tomamichel††

Abstract. The key cryptographic protocols used to secure the internet and financial transactions of today are all susceptible to attack by the development of a sufficiently large quantum computer. One particular area at risk is cryptocurrencies, a market currently worth over 100 billion USD. We investigate the risk posed to Bitcoin, and other cryptocurrencies, by attacks using quantum computers. We find that the proof-of-work used by Bitcoin is relatively resistant to substantial speedup by quantum computers in the next 10 years, mainly because specialized ASIC miners are extremely fast compared to the estimated clock speed of near-term quantum computers. On the other hand, the elliptic curve signature scheme used by Bitcoin is much more at risk, and could be completely broken by a quantum computer as early as 2027, by the most optimistic estimates. We analyze an alternative proof-of-work called Momentum, based on finding collisions in a hash function, that is even more resistant to speedup by a quantum computer. We also review the available post-quantum signature schemes to see which one would best meet the security and efficiency requirements of blockchain applications.

1. Introduction

Bitcoin is a decentralized digital currency secured by cryptography. Since its development by Satoshi Nakamato in 2008,[1] Bitcoin has proven to be a remarkably successful and secure system and has inspired the development of hundreds of other cryptocurrencies and blockchain technologies in a market currently worth over 100 billion USD.

The security of Bitcoin derives from several different features of its protocol. The first is the proof-of-work that is required to write transactions to the Bitcoin digital ledger. The work required to do this safeguards against malicious parties who possess less than 50% of the computational power of the network from creating an alternative history of transactions. The second is the cryptographic signature that is used to authorize transactions. Bitcoin currently uses a signature scheme based on elliptic curves.

The coming development of quantum computers poses a serious threat to almost all of the

* 3KNzjxAUuA199FbmWaA7ide4PvhVcKobCd

† D. Aggarwal (dcsdiva@nus.edu.sg) is an Assistant Professor in the Department of Computer Science and Principal Investigator at the Centre for Quantum Technologies at NUS, Singapore.

‡ G. K. Brennen (gavin.brennen@mq.edu.au) is an Associate Professor at Macquarie University.

§ T. Lee (troyjlee@gmail.com) is an Associate Professor at the University of Technology Sydney.

¶ M. Santha (miklos.santha@gmail.com) is Senior Researcher at the CNRS, IRIF, Université Paris Diderot and Principal Investigator at the Centre for Quantum Technologies at NUS, Singapore.

†† M. Tomamichel (marco.tomamichel@uts.edu.au) is Senior Lecturer in Quantum Information at the University of Technology Sydney.

cryptography currently used to secure the internet and financial transactions, and also to Bitcoin. The basic attack vectors on Bitcoin by quantum computers are known in the Bitcoin community.[2] Our contribution in this paper is to more precisely and quantitatively analyze these threats to give reasonable estimates as to when they might be viable. We find that the proof-of-work used by Bitcoin is relatively resistant to substantial speedup by quantum computers in the next 10 years, mainly because specialized ASIC miners are extremely fast compared to the estimated clock speed of near-term quantum computers. This means that transactions, once on the blockchain, would still be relatively protected even in the presence of a quantum computer.

The elliptic curve signature scheme used by Bitcoin is well-known to be broken by Shor's algorithm for computing discrete logarithms.[3] We analyse exactly *how long* it might take to derive the secret key from a published public key on a future quantum computer. This is critical in the context of Bitcoin as the main window for this attack is from the time a transaction is broadcast until the transaction is processed into a block on the blockchain with several blocks after it. By our most optimistic estimates, as early as 2027 a quantum computer could exist that can break the elliptic curve signature scheme in less than 10 minutes, the block time used in Bitcoin.

We also suggest some countermeasures that can be taken to secure Bitcoin against quantum attacks. We analyse an alternative proof-of-work scheme called Momentum,[4] based on finding collisions in a hash function, and show that it admits even less of a quantum speedup than the proof-of-work used by Bitcoin. We also review alternative signature schemes that are believed to be quantum safe.

2. Blockchain Basics

In this section we give a basic overview of how Bitcoin works, so that we can refer to specific parts of the protocol when we describe possible quantum attacks. We will keep this discussion at an abstract level, as many of the principles apply equally to other cryptocurrencies with the same basic structure as Bitcoin.

All Bitcoin transactions are stored in a public ledger called the *blockchain*. Individual transactions are bundled into blocks, and all transactions in a block are considered to have occurred at the same time. A time ordering is placed on these transactions by placing them in a chain. Each block in the chain (except the very first, or *genesis* block) has a pointer to the block before it in the form of the hash of the previous block's header.

Blocks are added to the chain by *miners*. Miners can bundle unprocessed transactions into a block and add them to the chain by doing a *proof-of-work* (PoW). Bitcoin, and many other coins, use a PoW developed by Adam Back called Hashcash.[5] The hashcash PoW is to find a well-formed *block header* such that $h(\text{header}) \leq t$, where h is a cryptographically secure hash function and header is the block header. A well-formed header contains summary information of a block such as a hash derived from transactions in the block,[6] a hash of the previous block header, a time stamp, as well as a so-called *nonce*, a 32-bit register that can be freely chosen. An illustration of a block can be found in Table 1. The parameter t is a target value that can be changed to adjust the difficulty of the PoW. In Bitcoin, this parameter is dynamically adjusted every 2016 blocks such that the network takes about 10 minutes on average to solve the PoW.

In Bitcoin the hash function chosen for the proof of work is two sequential applications of

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

Version	0x20000012
Previous block header hash	00…0dfff7669865430b…
Merkle Root	730d68233e25bec2…
Timestamp	2017-08-07 02:12:18
Difficulty	860,221,984,436.22
Nonce	941660394
Transaction 1	
Transaction 2	
⋮	

Table 1. Illustration of a block. The data in the top constitutes the block header.

the SHA256 : $\{0,1\}^* \to \{0,1\}^{256}$ hash function, *i.e.* $h(\cdot) = \text{SHA256}(\text{SHA256}(\cdot))$. As the size of the range of h is then 2^{256}, the expected number of hashes that need to be tried to accomplish the hashcash proof of work with parameter t is $2^{256}/t$. Rather than t, the Bitcoin proof-of-work is usually specified in terms of the *difficulty* D where $D = 2^{224}/t$. This is the expected number of hashes needed to complete the proof of work divided by 2^{32}, the number of available nonces. In other words, the difficulty is the expected number of variations of transactions and time stamps that need to be tried when hashing block headers, when for each fixing of the transactions and time stamp all nonces are tried.

Miners can bundle unprocessed transactions into a block however they like, and are awarded a number of bitcoins for succeeding in the PoW task. The "generation" transaction paying the mining reward is also a transaction included in the block, ensuring that different miners will be searching over disjoint block headers for a good hash pre-image.

Once a miner finds a header satisfying $h(\text{header}) \leq t$, they announce this to the network and the block is added to the chain. Note that it is easy to verify that a claimed header satisfies the PoW condition — it simply requires one evaluation of the hash function.

The purpose of the PoW is so that one party cannot unilaterally manipulate the blockchain in order to, for example, double spend. It is possible for the blockchain to fork, but at any one time the protocol dictates that miners should work on the fork that is currently the longest. Once a block has k many blocks following it in the longest chain, a party who wants to create a longest chain not including this block would have to win a PoW race starting k blocks behind. If the party controls much less than half of the computing power of the network, this becomes very unlikely as k grows. In Bitcoin, a transaction is usually considered safe once it has 6 blocks following it.

The first question we will look at in Section 3.1 is what advantage a quantum computer would have in performing the hashcash PoW, and if it could unilaterally "come from behind" to manipulate the blockchain.

The second aspect of Bitcoin that is important for us is the form that transactions take. When Bob wants to send bitcoin to Alice, Alice first creates (an ideally fresh) private-public key pair. The public key is hashed to create an *address*. This address is what Alice provides to Bob as the destination to send the bitcoin. Bitcoin uses the hash of the public key as the address instead of the public key not for security reasons but simply to save space.[7] As we see later, this design choice does have an impact on the *quantum* security.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

To send bitcoin to Alice, Bob must also point to transactions on the blockchain where bitcoin was sent to addresses that he controls. The sum of bitcoin received to these referenced transactions must add up to at least the amount of bitcoin Bob wishes to send to Alice. Bob proves that he owns these addresses by stating the public key corresponding to each address and using his private key corresponding to this address to sign the message saying he is giving these bitcoins to Alice.

3. Quantum Attacks on Bitcoin

3.1. Attacks on the Bitcoin Proof-of-Work—In this section, we investigate the advantage a quantum computer would have in performing the hashcash PoW used by Bitcoin. Our findings can be summarized as follows: Using Grover search,[8] a quantum computer can perform the hashcash PoW by performing quadratically fewer hashes than is needed by a classical computer. However, the extreme speed of current specialized ASIC hardware for performing the hashcash PoW, coupled with much slower projected gate speeds for current quantum architectures, essentially negates this quadratic speedup, at the current difficulty level, giving quantum computers no advantage. Future improvements to quantum technology allowing gate speeds up to 100GHz could allow quantum computers to solve the PoW about 100 times faster than current technology. However, such a development is unlikely in the next decade, at which point classical hardware may be much faster, and quantum technology might be so widespread that no single quantum enabled agent could dominate the PoW problem.

We now go over these results in detail. Recall that the Bitcoin PoW task is to find a valid block header such that $h(\text{header}) \leq t$, where $h(\cdot) = \text{SHA256}(\text{SHA256}(\cdot))$. The security of the blockchain depends on no agent being able to solve the PoW task first with probability greater than 50%. We will investigate the amount of classical computing power that would be needed to match one quantum computer in performing this task.

We will work in the random oracle model,[9] and in particular assume that $\Pr[h(\text{header}) \leq t] = t/2^{256}$ where the probability is taken uniformly over all well-formed block headers that can be created with transactions available in the pool at any given time (such well-formed block headers can be found by varying the nonce, the transactions included in the block as well as the least significant bits of the timestamp of the header). On a classical computer, the expected number of block headers and nonces which need to be hashed in order to find one whose hash value is at most t is $D \times 2^{32}$ where D is the hashing difficulty defined by $D = 2^{224}/t$.[10]

For quantum computers in the random oracle model we can restrict our attention to the generic quantum approach to solving the PoW task using Grover's algorithm.[8] By Grover's algorithm, searching a database of N items for a marked item can be done with $O(\sqrt{N})$ many queries to the database (whereas any classical computer would require $\Omega(N)$ queries to complete the same task).

Let $N = 2^{256}$ be the size of the range of h for the following. By our assumptions, with probability at least 0.9999 a random set of $10 \cdot N/t$ many block headers will contain at least one element whose hash is at most t. We can fix some deterministic function g mapping $S = \{0,1\}^{\lceil \log(10 \cdot N/t) \rceil}$ to distinct well-formed block headers. We also define a function f which

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

determines if a block header is "good" or not

$$f(x) = \begin{cases} 0 & \text{if} \quad h(g(x)) > t \\ 1 & \text{if} \quad h(g(x)) \leq t \end{cases}.$$

A quantum computer can compute f on a superposition of inputs, *i.e.* perform the mapping

$$\sum_{x \in S} \alpha_x |x\rangle \rightarrow \sum_{x \in S} (-1)^{f(x)} \alpha_x |x\rangle.$$

Each application of this operation is termed an *oracle call*. Using Grover's algorithm a quantum algorithm can search through S to find a good block header by computing $\#\mathscr{O} = \frac{\pi}{4}\sqrt{10 \cdot N/t} = \pi 2^{14}\sqrt{10 \cdot D}$ oracle calls. The Grover algorithm can be adapted to run with this scaling even if the number of solutions is not known beforehand, and even if no solutions exist.[11]

While the number of oracle calls determines the number of hashes that need to be performed, additional overhead will be incurred to compute each hash, to construct the appropriate block header, and to do quantum error correction. We now analyze these factors to determine a more realistic estimate of the running time in two ways. First, we estimate the running time based on a well studied model for universal quantum computers with error correction.

On a classical computer, a hash function such as SHA256 uses basic boolean gate operations, whereas on a quantum computer, these elementary boolean gates are translated into reversible logical quantum gates which introduces some overhead. There are a total of 64 rounds of hashing in the SHA256 protocol and each round can be done using an optimized circuit with 683 Toffoli quantum gates.[12] The Toffoli gate is a three qubit controlled-controlled not gate defined by its action on a bit string: $\texttt{Toffoli}|x_1\rangle|x_2\rangle|x_3\rangle = |x_1\rangle|x_2\rangle|x_3 \oplus x_1 x_2\rangle$. Most quantum error correction codes use T gates rather than Toffoli gates as the representative time consuming gate. The T gate is a single qubit gate defined by the action $T|x\rangle = e^{ix\pi/4}|x\rangle$. Like the Tofolli, the T gate is a so called non-Clifford gate which means, for most error correction codes, it is more resource demanding to implement fault tolerantly, requiring (for example) state distillation factories. A careful analysis of the cost to perform the SHA256 function call as well as the inversion about the mean used in the Grover algorithm finds a total T gate count of 474168 for one oracle call.[13] In that circuit decomposition, the T gates can only be parallelized by roughly a factor of three.

There is additional overhead needed by quantum computers to perform error correction. In deciding on a good quantum error correction code there are a variety of tradeoffs to consider: tolerance to a particular physical error model, versatility for subroutines, number of qubits used, logical gate complexity, and the amount of classical processing of error syndromes and feedback. Adopting the surface code, which has advantages of a relatively high fault tolerance error threshold and local syndrome measurements, we can adapt the analysis in Suchara (*et al.*) to estimate the total run time of the quantum algorithm.[13] The time needed to run the Grover algorithm and successfully mine a block is

$$\tau = \#\mathscr{O} \times \#G/s = \pi 2^{14}\sqrt{10 \cdot D} \times \#G/s,$$

where $\#G$ is the number of cycles needed for one oracle call, and s is the quantum computer clock speed. Using a surface code, where the dominant time cost is in distilling magic states to implement T gates, one finds

$$\#G = 297784 \times c_\tau(D, p_g),$$

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

where the first factor includes the logical T gate depth for calling the SHA256 function twice as required by Bitcoin PoW, and twice again to make the circuit reversible, as well as the inversion about the mean. The second factor, c_τ, is the overhead factor in time needed for quantum error correction. It counts the number of clock cycles per logical T gate and is a function of difficulty and the physical gate error rate p_g. For a fixed gate error rate, the overhead factor c_τ is bounded above by the cost to invert a 256 bit hash (maximum difficulty).

Because the quantum algorithm runs the hashing in superposition, there is no direct translation of quantum computing power into a hashing rate. However, we can define an effective hash rate, h_{QC}, as the expected number of calls on a classical machine divided by the expected time to find a solution on the quantum computer, viz.

$$h_{\mathrm{QC}} \equiv \frac{N/t}{\tau} = \frac{0.28 \times s\sqrt{D}}{c_\tau(D, p_g)}.$$

Because the time overhead is bounded, asymptotically the effective hashing rate improves as the square root of the difficulty, reflecting the quadratic advantage obtainable from quantum processors.

The Grover algorithm can be parallelized over d quantum processors. In the optimal strategy, each processor is dedicated to search over the entire space of potential solutions, and the expected number of oracle calls needed to find a solution is $\#\mathscr{O} = 0.39 \times \#\mathscr{O}/\sqrt{d}$.[14] This implies an expected time to find a solution is

$$\tau_{\|} = 0.39 \times \tau/\sqrt{d},$$

and the effective hash rate using d quantum processors in parallel is

$$h_{\mathrm{QC},\|} = 2.56 \times h_{\mathrm{QC}}\sqrt{d}.$$

The number of logical qubits needed in the Grover algorithm is fixed at 2402, independent of the difficulty. The number of physical qubits needed is

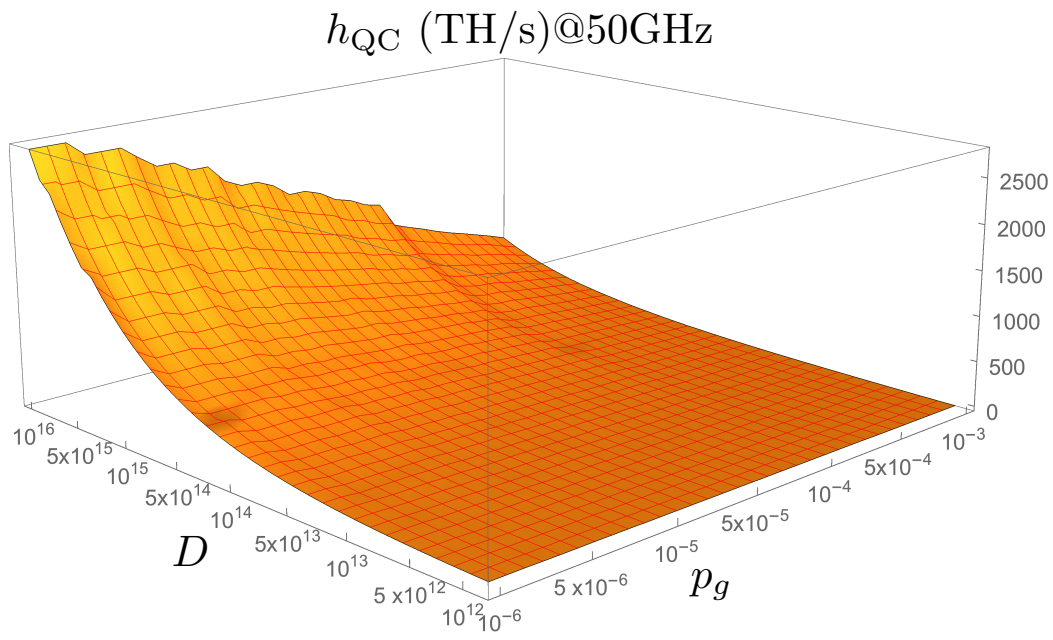
$$n_Q = 2402 \times c_{n_Q}(D, p_g),$$

where c_{n_Q} is the overhead in space, *i.e.* physical qubits, incurred due to quantum error correction, and is also a function of difficulty and gate error rate.

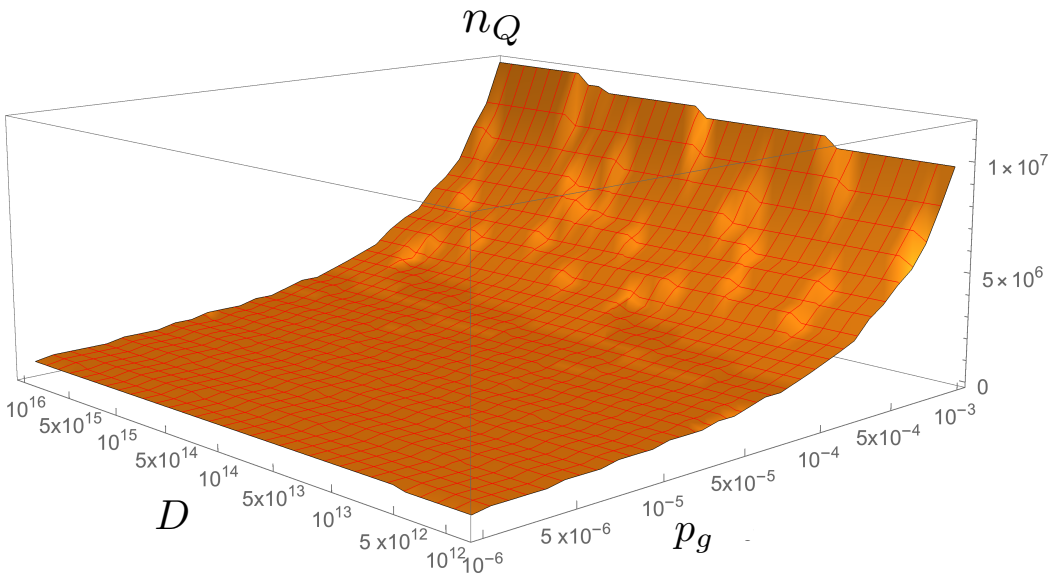
In Appendix 1 we show how to calculate the overheads in time and space incurred by error correction. The results showing the performance of a quantum computer for blockchain attacks are given in Figure 1. To connect these results to achievable specifications, we focus on superconducting circuits which as of now have the fastest quantum gate speeds amoung candidate quantum technologies and offer a promising path forward to scalability. Assuming maximum gate speeds attainable on current devices of $s = 66.7$MHz,[15] and assuming an experimentally challenging, but not implausible, physical gate error rate of $p_g = 5 \times 10^{-4}$, and close to current difficulty $D = 10^{12}$, the overheads are $c_\tau = 538.6$ and $c_{n_Q} = 1810.7$, implying an effective hash rate of $h_{\mathrm{QC}} = 13.8$GH/s using $n_Q = 4.4 \times 10^6$ physical qubits. This is more than one thousand times slower than off the shelf ASIC devices which achieve hash rates of 14TH/s;[16] the reason being the slow quantum gate speed and delays for fault tolerant T gate construction.

Quantum technologies are poised to scale up significantly in the next decades with a quantum version of Moore's law likely to take over for quantum clock speeds, gate fidelities, and qubit

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127



(a)



(b)

Fig. 1. Performance of a single quantum computer for blockchain attacks as a function of physical gate error rate p_g, which is an internal machine specification, and mining Difficulty D, which is set by the blockchain protocol. (a) Effective hash rate h_{QC} for a quantum computer operating at 50GHz clock speed which is at the optimistic limit of foreseeable clock speeds. The hash rate increases as the square root of difficulty (note the log scale). For d quantum computers acting in parallel the effective hash rate increases by a factor of $2.56 \times \sqrt{d}$.(b) Number of physical qubits n_{Q} used by the quantum computer.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

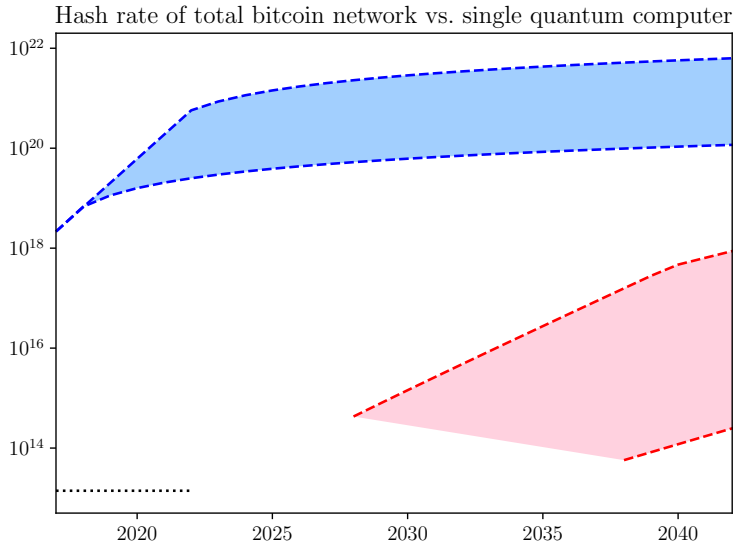


Fig. 2. This plot shows two estimates of the hashing power (in hashes per second) of the Bitcoin network (blue striped curves) vs. a single quantum computer (red striped curves) as a function of time for the next 25 years. We give more and less optimistic estimates and uncertainty regions (blue and orange area). The model is described in detail in Appendices 2 and 3. Prior to 2028 (in the more optimistic estimate) there will not be any quantum computer with sufficiently many qubits to implement the Grover algorithm. For comparison, the black dotted line shows the hash rate of a single ASIC device today.

number. Guided by current improvements in superconducting quantum circuit technology, forecasts for such improvements are given in Appendices 2 and 3. This allows us to estimate of the power of a quantum computer as a function of time as shown in Figure 2. Evidently, it will be some time before quantum computers outcompete classical machines for this task, and when they do, a single quantum computer will not have a majority of hashing power.

Nonetheless, certain attacks become more profitable for an adversary armed with quantum computers with even modest hashing power advantage over classical miners. One example is a mining pool attack wherein a malicious outside party pays pool members to withhold their valid block solutions.[17] This reduces the effective mining power of the pool and increases the relative power of the adversary. Smart contracts can be added to the blockchain to enforce the attacker's bribes and the pool members compliance if they agree to withhold. Remarkably, such an attack is profitable even when the hashing power of the attacker is well below half of the entire network. For example, an attacker with 0.1% of the total network hashing power could, with only a small bribe, cause pool revenue to decrease by 10%. This level of quantum hashing power could be realized by an adversary controlling 20 quantum computers running in parallel with specifications at the minimum of the optimistic assumptions outlined in Appendix 1 where the effective hash rate scales like $h_{\mathrm{QC}} = 0.04 \times s\sqrt{D}$, assuming difficulty $D = 10^{13}$ and clock speed $s = 50\mathrm{GHz}$.

3.2. Attacks on Signatures—Signatures in Bitcoin are made using the Elliptic Curve Digital Signature Algorithm based on the secp256k1 curve. The security of this system is based on

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

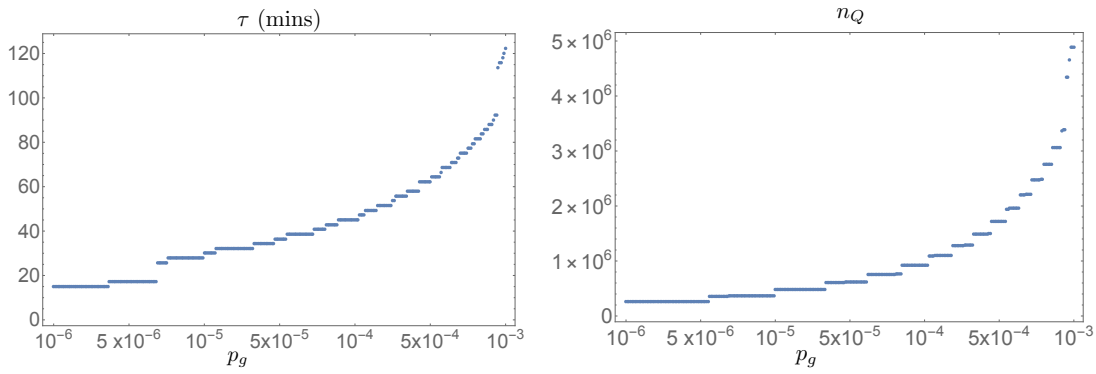


Fig. 3. Performance of a quantum computer operating at 10GHz clock speed for attacks on digital signatures using the elliptic curve digital signature algorithm. (a) Time in minutes to break the signature as a function of physical gate error rate p_g. (b) Number of physical qubits used by the quantum computer.

the hardness of the Elliptic Curve Discrete Log Problem (ECDLP). While this problem is still believed to be hard classically, an efficient quantum algorithm to solve this problem was given by Shor.[3] This algorithm means that a sufficiently large universal quantum computer can efficiently compute the private key associated with a given public key rendering this scheme completely insecure. The implications for Bitcoin are the following:

(1) (Reusing addresses) To spend bitcoin from an address the public key associated with that address must be revealed. Once the public key is revealed in the presence of a quantum computer the address is no longer safe and thus should never be used again. While always using fresh addresses is already the suggested practice in Bitcoin, in practice this is not always followed. Any address that has bitcoin and for which the public key has been revealed is completely insecure.

(2) (Processed transactions) If a transaction is made from an address which has not been spent from before, and this transaction is placed on the blockchain with several blocks following it, then this transaction is reasonably secure against quantum attacks. The private key could be derived from the published public key, but as the address has already been spent this would have to be combined with out-hashing the network to perform a double spending attack. As we have seen in Section 3.1, even with a quantum computer a double spending attack is unlikely once the transaction has many blocks following it.

(3) (Unprocessed transactions) After a transaction has been broadcast to the network, but before it is placed on the blockchain it is at risk from a quantum attack. If the secret key can be derived from the broadcast public key before the transaction is placed on the blockchain, then an attacker could use this secret key to broadcast a new transaction from the same address to his own address. If the attacker then ensures that this new transaction is placed on the blockchain first, then he can effectively steal all the bitcoin behind the original address.

We view item (3) as the most serious attack. To determine the seriousness of this attack it is important to precisely estimate how much time it would take a quantum computer to compute the

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

ECDLP, and if this could be done in a time close to the block interval. For an instance with an n bit prime field, a recently optimized analysis shows a quantum computer can solve the problem using $9n+2\lceil\log_2(n)\rceil+10$ logical qubits and $(448\log_2(n)+4090)n^3$ Toffoli gates.[18] Bitcoin uses $n=256$ bit signatures so the number of Toffoli gates is 1.28×10^{11}, which can be slightly parallelized to depth 1.16×10^{11}. Each Toffoli can be realized using a small circuit of T gate depth one acting on 7 qubits in parallel (including 4 ancilla qubits).[19]

Following the analysis of Sec. 3.1, we can estimate the resources needed for a quantum attack on the digital signatures. As with block mining, the dominant time is consumed by distilling magic states for the logical T gates. The time to solve the ECDLP on a quantum processor is

$$\tau = 1.28\times10^{11}\times c_\tau(p_g)/s,$$

where the time overhead c_τ now only depends on gate error rate, and s is again the clock speed. The number of physics qubits needed is

$$n_Q = 2334\times c_{n_Q}(p_g),$$

where the first factor is the number of logical qubits including 4 logical ancilla qubits, and c_{n_Q} is the space overhead.

The performance of a quantum computer to attack digital signatures is given in Figure 3. Using a surface code with a physical gate error rate of $p_g=5\times10^{-4}$, the overhead factors are $c_\tau=291.7$ and $c_{n_Q}=735.3$, and the time to solve the problem at 66.6 MHz clock speed is 6.49 days using 1.7×10^6 physical qubits. Looking forward to performance improvements, for 10GHz clock speed and error rate of 10^{-5}, the signature is cracked in 30 minutes using 485550 qubits. The latter makes the attack in item (3) quite possible and would render the current Bitcoin system highly insecure. An estimate of the time required for a quantum computer to break the signature scheme as a function of time is given in Figure 4, based on the model described in Appendices 2 and 3.

3.3. Future Enhancements of Quantum Attacks—We have described attacks on the Bitcoin protocol using known quantum algorithms and error correction schemes. While some of the estimates for quantum computing speed and scaling may appear optimistic, it is important to keep in mind that there are several avenues for improved performance of quantum computers to solve the aforementioned problems.

First, the assumed error correction code here is the surface code which needs significant classical computational overhead for state distillation, error syndrome extraction, and correction. Other codes which afford transversal Clifford *and* non-Clifford gates could overcome the need for slow state distillation.[20] In fact the slow down from classical processing for syndrome extraction and correction could be removed entirely using a measurement free protocol.[21] Recent analysis of measurement free error correction using the surface code finds error thresholds only about 6 times worse than the measurement based approach.[22] This could potentially dramatically improve overall speed of error correction.

Second, reductions in logical gate counts of the quantum circuits are possible as more efficient advanced quantum-computation techniques are being developed. For example, using a particular large-size example problem (including oracle implementations) that was analyzed in a previous work,[23] a direct comparison of the concrete gate counts, obtained by the software package Quipper, has been achieved between the old and the new linear-systems solving quantum

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

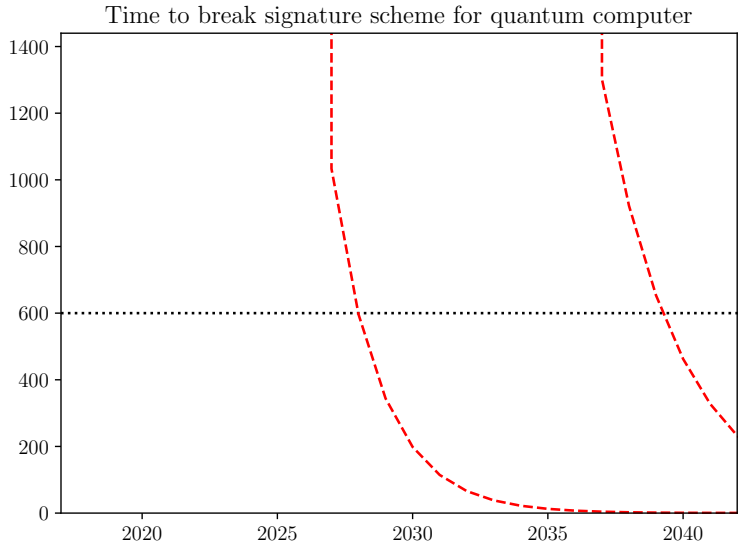


Fig. 4. This plot shows two estimates of the time (in seconds) required for a quantum computer to break the signature scheme (red curves) as a function of time for the next 25 years. We give more and less optimistic estimates (red striped lines). The models are described in detail in Appendix 3. According to this estimate, the signature scheme can be broken in less than 10 minutes (600 seconds, black dotted line) as early as 2027.

algorithms,[24, 25] showing an improvement of several orders of magnitude.[26] Given that the quantum Shor and Grover algorithms have been well studied and highly optimized, one would not expect such a dramatic improvement, nonetheless it is likely some improvement is possible.

Third, different quantum algorithms might provide relative speedups. Recent work by Kaliski,[27] presents a quantum algorithm for the Discrete Logarithm Problem: find m given $b = a^m$, where b is a known target value and a is a known base, using queries to a so called "magic box" subroutine which computes the most significant bit of m. By repeating queries using judiciously chosen powers of the base, all bits of m can be calculated and the problem solved. Problem queries can be distributed to many quantum computers to solve in parallel. While each such query would requires a number of logical qubits and gates comparable to solving the entire problem, there may be some overall speedup since the number of measurements at the end is reduced and required precision of logical gates may be less meaning lower overheads for fault tolerant implementation.

4. Countermeasures

4.1. Alternative Proofs-of-Work—As we have seen in the last section, a quantum computer can use Grover search to perform the Bitcoin proof-of-work using quadratically fewer hashes than are needed classically. In this section we investigate alternative proofs-of-work that might offer less of a quantum advantage. The basic properties we want from a proof-of-work are:

(1) (Difficulty) The difficulty of the problem can be adjusted in accordance with the computing power available in the network.
(2) (Asymmetry) It is much easier to verify the proof-of-work has been successfully com-

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

pleted than to perform the proof-of-work.

(3) (No quantum advantage) The proof-of-work cannot be accomplished significantly faster with a quantum computer than with a classical computer.

The Bitcoin proof-of-work accomplishes items (1) and (2), but we would like to find an alternative proof of work that does better on (3).

Similar considerations have been investigated by authors trying to find a proof-of-work that, instead of (3) look for proofs-of-work that cannot be accelerated by ASICs. An approach to doing this is by looking at memory intensive proofs of work. Several interesting candidates have been suggested for this such as Momentum,[4] based on finding collisions in a hash function, Cuckoo Cycle,[28] based on finding constant sized subgraphs in a random graph, and Equihash,[29] based on the generalized birthday problem. These are also good candidates for a more quantum resistant proof-of-work.

These schemes all build on the hashcash-style proof-of-work and use the following template. Let $h_1 : \{0,1\}^* \to \{0,1\}^n$ be a cryptographically secure hash function and $H = h_1(\text{header})$ be the hash of the block header. The goal is then to find a nonce x such that

$$h_1(H \parallel x) \leq t \text{ and } P(H,x) \ ,$$

for some predicate P. The fact that the header and nonce have to satisfy the predicate P means that the best algorithm will no longer simply iterate through nonces x in succession. Having a proof-of-work of this form also ensures that the parameter t can still be chosen to vary the difficulty.

In what follows, we will analyse this template for the Momentum proof-of-work, as this can be related to known quantum lower bounds. For the momentum proof of work, let $h_2 : \{0,1\}^* \to \{0,1\}^\ell$ be another hash function with $n \leq \ell$. In the original Momentum proposal h_1 can be taken as SHA-256 and h_2 as a memory intensive hash function, but this is less important for our discussion. The proof-of-work is to find H, a, b such that

$$h_1(H \parallel a \parallel b) \leq t \text{ and } h_2(H \parallel a) = h_2(H \parallel b) \text{ and } a,b \leq 2^\ell \ . \tag{1}$$

First let's investigate the running time in order to solve this proof-of-work, assuming that the hash functions h_1, h_2 can be evaluated in unit time. Taking a subset $S \subset \{0,1\}^\ell$ and evaluating $h_2(H \parallel a)$ for all $a \in S$, we expect to find about $|S|^2/2^\ell$ many collisions. Notice that by using an appropriate data structure, these collisions can be found in time about $|S|$.

One algorithm is then as follows. For each H, we evaluate h_2 on a subset S and find about $|S|^2/2^\ell$ many pairs a,b such that $h_2(H \parallel a) = h_2(H \parallel b)$. For each collision we then test $h_1(H \parallel a \parallel b) \leq t$. In expectation, we will have to perform this second test $2^n/t$ many times. Thus the number of H's we will have to try is about $m = \max\{1, \frac{2^{n+\ell}}{t|S|^2}\}$, since we have to try at least one H. As for each H we spend time $|S|$, the total running time is $m|S|$. We see that it is the smallest when $|S| = \sqrt{2^{n+\ell}/t}$, that is when $m = 1$, and we just try one H. This optimal running time is then $T = \sqrt{2^{n+\ell}/t}$, and to achieve it we have to use a memory of equal size to the running time, which might be prohibitive. For some smaller memory $|S| < \sqrt{2^{n+\ell}/t}$ the running time will be $\frac{2^{n+\ell+1}}{t|S|}$.

Now let us look at the running time on a quantum computer. On a quantum computer we can do the following. Call H *good* if there exists $a,b \in S$ such that $h_1(H \parallel a \parallel b) \leq t$ and $h_2(H \parallel a) = h_2(H \parallel b)$. Testing if an H is good requires finding a collision, and therefore necessitates at

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

least $|S|^{2/3}$ time by the quantum query lower bound of Aaronson and Shi.[30] Note that this lower bound is tight as finding such a collision can also be done in roughly $|S|^{2/3}$ time using Ambainis's element distinctness algorithm.[31] We have argued above that a set of size $m = \max\{1, \frac{2^{n+\ell}}{t|S|}\}$ is needed to find at least one good H. By the optimality of Grover search we know that we have to perform at least $\sqrt{m}$ many tests to find a good H.[32] As testing if an H is good requires time $|S|^{2/3}$, the total running time is at least $\sqrt{m}|S|^{2/3}$. As the classical running time is $m|S|$, we see that unlike for the current proof of work in Bitcoin, with this proposal a quantum computer would not be able to achieve a quadratic advantage as soon as S is more than constant size. In particular, since $\sqrt{m}|S|^{2/3}$ is minimized also when $S = \sqrt{2^{n+\ell}/t}$, the running time of even the fastest quantum algorithm is at least $T^{2/3}$, which is substantially larger than $T^{1/2}$.

4.2. Review of Post-Quantum Signature Schemes—Many presumably quantum-safe public-key signature schemes have been proposed in the literature. Some examples of these are hash-based signature schemes (LMS,[33] XMSS,[34] SPHINCS,[35] and NSW [36]), code-based schemes (CFS [37] and QUARTZ [38]), schemes based on multivariate polynomials (RAINBOW [39]), and lattice-based schemes (GPV,[40] LYU,[41] BLISS,[42] ring-TESLA,[43] DILITHIUM,[44] and NTRU [45]). Each of these cryptosystems have varying degree of efficiency. For a comparison in terms of signature size and key size, see Table 2.

In the blockchain context the most important parameters of a signature scheme are the signature and public key lengths, as these must be stored in some capacity to fully verify transactions, and the time to verify the signature. Looking at Table 2, with respect to the sum of signature and public key lengths, the only reasonable options are hash and lattice based schemes.

Hash based schemes like XMSS have the advantage of having provable security, at least assuming the chosen hash function behaves like a random oracle. The generic quantum attack against these schemes is to use Grover's algorithm which means that their quantum security level is half of the classical security level. In contrast, the best known quantum attack against DILITHIUM at 138 bit classical security level requires time 2^{125}. Thus at the same level of *quantum* security, lattice based schemes have some advantage in signature plus public key length.

Although the lattice based scheme BLISS has the shortest sum of signature and public key lengths of all the schemes in Table 2, there are some reasons not to choose BLISS in practice. The security of BLISS relies on hardness of the NTRU problem and the assumption that solving this problem is equivalent to finding a short vector in a so-called NTRU lattice. It has been shown recently that this assumption might be too optimistic, at least for large parameters.[46] Moreover, there is a history of attacks on prior NTRU-based signature schemes.[47,48] Perhaps most fatally, BLISS is difficult to implement in a secure way as it is very susceptible to side channel attacks. The production grade strongSwan implementation of BLISS has been attacked in this way by Pessl (*et al.*),[49] who showed that the signing key could be recovered after observing about 6000 signature generations.

Acknowledgement

MT and GB would like to thank Michael Bremner for initial discussions. TL would like to thank John Tromp for helpful comments and discussions about proof-of-work and Ronald de Wolf for conversations about parallel quantum search. This material is based on work supported in part by the Singapore National Research Foundation under NRF RF Award No. NRF-NRFF2013-13.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

type	name	classical security (bits)	quantum security (bits)	PK length (kb)	signature length (kb)	total length (kb)
	ECDSA	127	0	0.3	0.5	0.8
I.1	GPV[50]	100		300	240	540
I.2	LYU[50]	100		65	103	168
I.3	BLISS[42]	128		7	5	12
I.4	FALCON-512*[51]	114	103	7.2	4.9	12.1
I.5	ring-TESLA[43]	128		26.6	11.9	38.5
I.6	qTESLA-128*[52]	128		23.8	21.7	45.4
I.7	DILITHIUM*[44]	138	125	11.8	21.6	33.4
II.1	RAINBOW[53]	160		305	0.2	305.2
III.1	LMS[54]	256	128	0.8	22.6	23.4
III.2	XMSS[34]	196	93	13.6	22.3	35.9
III.3	SPHINCS[35]	256	128	8.4	328	336.4
III.4	NSW[36]	128		0.3	36	36.3
IV.1	CFS[37]	83		9216	0.1	9216.1
IV.2	QUARTZ[38]	80		568	0.1	568.1

Table 2. Comparison of the public key (PK) and signature lengths of post-quantum signature schemes in kilobits (kb). As a reference, the parameters for ECDSA are also given. The security level given is against classical and quantum (where available) attacks. Type I are lattice based, type II based on multivariate polynomials, type III hashing based, and type IV code based. An asterisk indicates schemes that have been submitted to the NIST call on post-quantum cryptography.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

Research at the Centre for Quantum Technologies is partially funded by the Singapore Ministry of Education and the National Research Foundation under grant R-710-000-012-135. This research was supported in part by the QuantERA ERA-NET Cofund project QuantAlgo.

Author Contributions

All authors contributed equally.

Notes and References

[1] Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System." *Bitcoin.org* (2009) (accessed 2 October 2018) http://www.bitcoin.org/pdf.

[2] Buterin, V. "Bitcoin Is Not Quantum-Safe, and How We Can Fix It When Needed." *Bitcoin Magazine* (2013) (accessed 2 October 2018) http://bitcoinmagazine.com/articles/bitcoin-is-not-quantum-safe-and-how-we-can-fix-1375242150/.

[3] Shor, P. W. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." *SIAM Review* **41.2** 303–332 (1999) https://doi.org/10.1137/S0036144598347011.

[4] Larimer, D. "Momentum–A Memory-Hard Proof-of-Work via Finding Birthday Collisions." (2014) (accessed 2 October 2018) http://www.hashcash.org/papers/momentum.pdf.

[5] Back, A. "Hashcash–A Denial of Service Counter-Measure." *Hashcash.org* (2002) (accessed 2 October 2018) http://www.hashcash.org/papers/hashcash.pdf.

[6] Specifically the root of a Merkle tree of hashes of the transactions.

[7] In early versions of the Bitcoin protocol the public key could be used as an address.

[8] Grover, L. K. "A Fast Quantum Mechanical Algorithm for Database Search." In *STOC '96 Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing.* New York: Association for Computing Machinery 212–219 (1996) https://doi.org/10.1145/237814.237866.

[9] Bellare, M., Rogaway, P. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols." In *CCS '93 Proceedings of the 1st ACM Conference on Computer and Communications Security.* New York: Association for Comuting Machinery 62–73 (1993) https://doi.org/10.1145/168588.168596.

[10] According to blockchain.info, on August 8, 2017, the hashing difficulty was $D = 860 \cdot 10^9$ and target was $t = 2^{184.4}$.

[11] Boyer, M., Brassard, G., Høyer, P., Tapp, A. "Tight Bounds on Quantum Searching." *Fortschritte der Physik* **46.4-5** 493–505 (1998) https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.

[12] Parent, A., Rötteler, M., Svore, K. M. "Reversible Circuit Compilation with Space Constraints." *CoRR (arXiv)* (2015) (accessed 2 October 2018) https://arxiv.org/abs/1510.00377.

[13] Suchara, M., Faruque, A., Lai, C.-Y., Paz, G., Chong, F., Kubiatowicz, J. D. "Estimating the Resources for Quantum Computation with the QuRE Toolbox." *EECS Department, University of California, Berkeley* (accessed 2 October 2018) http://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-119.html.

[14] Gingrich, R. M., Williams, C. P., Cerf, N. J. "Generalized Quantum Search with Parallelism." *Physical Review A* **61.5** 052313 (2000) https://link.aps.org/doi/10.1103/PhysRevA.61.052313.

[15] Kirchhoff, S., *et al.* "Optimized Cross-Resonance Gate for Coupled Transmon Systems." *arXiv* (2017) (accessed 2 October 2018) https://arxiv.org/abs/1701.01841.

[16] Using *e.g.* the Bitmain Antminer S9.

[17] Velner, Y., Teutsch, J., Luu, L. "Smart Contracts Make Bitcoin Mining Pools Vulnerable." *IACR Cryptology ePrint Archive* (2017) (accessed 2 October 2018) http://eprint.iacr.org/2017/230.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

[18] Roetteler, M., Naehrig, M., Svore, K., Lauter, K. "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms." *arXiv* (2017) (accessed 2 October 2018) https://arxiv.org/abs/1706.06752.

[19] Selinger, P. "Quantum Circuits of T-Depth One." *Physical Review A* **87.4** 042302 (2013) https://link.aps.org/doi/10.1103/PhysRevA.87.042302.

[20] Paetznick, A., Reichardt, B. W. "Universal Fault-Tolerant Quantum Computation with Only Transversal Gates and Error Correction." *Physical Review Letters* **111.9** 090505 (2013) https://link.aps.org/doi/10.1103/PhysRevLett.111.090505.

[21] Paz-Silva, G. A., Brennen, G. K., Twamley, J. "Fault Tolerance with Noisy and Slow Measurements and Preparation." *Physical Review Letters* **105.10** 100501 (2010) https://link.aps.org/doi/10.1103/PhysRevLett.105.100501.

[22] Ekmel Ercan, H., *et al.* "Measurement-Free Implementations of Small-Scale Surface Codes for Quantum Dot Qubits." *arXiv* (2017) (accessed 2 October 2018) https://arxiv.org/abs/1708.08683.

[23] Scherer, A., Valiron, B., Mau, S.-C., Alexander, S., van den Berg, E., Chapuran, T. E. "Concrete Resource Analysis of the Quantum Linear-System Algorithm Used to Compute the Electromagnetic Scattering Cross Section of a 2D Target." *Quantum Information Processing* **16.3** 60 (2017) https://doi.org/10.1007/s11128-016-1495-5.

[24] Harrow, A. W., Hassidim, A., Lloyd, S. "Quantum Algorithm for Linear Systems of Equations." *Physical Review Letters* **103.15** 150502 (2009) https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[25] Childs, A. M., Kothari, R., Somma, R. "Quantum Linear Systems Algorithm with Exponentially Improved Dependence on Precision." *arXiv* (2015) (accessed 2 October 2018) https://www.arxiv.org/abs/1511.02306.

[26] Scherer, A., personal communication.

[27] Kaliski, B. S., Jr. "A Quantum "Magic Box" for the Discrete Logarithm Problem." *IACR Cryptology ePrint Archive* (2017) (accessed 2 October 2018) https://eprint.iacr.org/2017/745.

[28] Tromp, J. "Cuckoo Cycle: A Memory Bound Graph-Theoretic proof-of-work." In M. Brenner, N. Christin, B. Johnson, K. Rohloff (Eds.), *Financial Cryptography and Data Security FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan Puerto Rico, January 30, 2015 Revised Selected Papers.* New York: Springer 49–62 (2015) https://doi.org/10.1007/978-3-662-48051-9_4.

[29] Biryukov, A., Khovratovich, D. "Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem." *Ledger* **2** 1–30 (2017) https://doi.org/10.5195/ledger.2017.48.

[30] Aaronson, S., Shi, Y. "Quantum Lower Bounds for the Collision and the Element Distinctness Problems." *Journal of the ACM* **51.4** 595–605 (2004) https://doi.org/10.1145/1008731.1008735.

[31] Ambainis, A. "Quantum Walk Algorithm for Element Distinctness." *SIAM Journal on Computing* **37.1** 210–239 (2007) https://doi.org/10.1137/S0097539705447311.

[32] Bennett, C., Bernstein, E., Brassard, G., Vazirani, U. "Strengths and Weaknesses of Quantum Computing." *SIAM Journal on Computing* **26.5** 1510–1523 (1997) https://doi.org/10.1137/S0097539796300933.

[33] Leighton, F. T., Micali, S. "Large Provably Fast and Secure Digital Signature Schemes Based on Secure Hash Functions." *Google Patents* (accessed 2 October 2018) https://patents.google.com/patent/US5432852A/en.

[34] Buchmann, J., Dahmen, E., Hülsing, A. "XMSS–A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions." In B. Yang (Ed.), *Post-Quantum Cryptography, 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011.* Berlin: Springer 117–129 (2011) https://doi.org/10.1007/978-3-642-25405-5_8.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

[35] Bernstein, D. J., *et al.* "SPHINCS: Practical Stateless Hash-Based Signatures." In E. Oswald, M. Fischlin (Eds.), *Advances in Cryptology–EUROCRYPT 2015, 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015.* Berlin: Springer 368–397 (2015) https://doi.org/10.1007/978-3-662-46800-5_15.

[36] Naor, D., Shenhav, A., Wool, A. "One-Time Signatures Revisited: Have They Become Practical?" *IACR Cryptology ePrint Archive* (2005) (accessed 2 October 2018) https://eprint.iacr.org/2005/442.

[37] Courtois, N., Finiasz, M., Sendrier, N. "How to Achieve a McEliece-Based Digital Signature Scheme." In C. Boyd (Ed.), *Advances in Cryptology—ASIACRYPT 2001.* Berlin: Springer 157–174 (2001) https://doi.org/10.1007/3-540-45682-1_10.

[38] Patarin, J., Courtois, N., Goubin, L. "Quartz, 128-Bit Long Digital Signatures." In D. Naccache (Ed.), *Topics in Cryptology–CT-RSA 2001, The Cryptographers' Track at RSA Conference 2001 San Francisco, CA, USA, April 8–12, 2001.* Berlin: Springer 282–297 (2001) https://doi.org/10.1007/3-540-45353-9_21.

[39] Ding, J., Schmidt, D. "Rainbow, A New Multivariable Polynomial Signature Scheme." In J. Ioannidis, A. Keromytis, M. Yung (Eds.), *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005.* Berlin: Springer 164–175 (2005) https://doi.org/10.1007/11496137_12.

[40] Gentry, C., Peikert, C., Vaikuntanathan, V. "Trapdoors for Hard Lattices and New Cryptographic Constructions." In *STOC '08 Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing.* New York: Association for Computing Machinery 197–206 (2008) https://doi.org/10.1145/1374376.1374407.

[41] Lyubashevsky, V. "Lattice Signatures Without Trapdoors." In D. Pointcheval, T. Johansson (Eds.), *Advances in Cryptology–EUROCRYPT 2012, 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012.* Berlin: Springer 738–755 (2012) https://doi.org/10.1007/978-3-642-29011-4_43.

[42] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V. "Lattice Signatures and Bimodal Gaussians." In R. Canetti, J. A. Garay (Eds.), *Advances in Cryptology–CRYPTO 2013, 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013.* Berlin: Springer 40–56 (2013) https://doi.org/10.1007/978-3-642-40041-4_3.

[43] Akleylek, S., Bindel, N., Buchmann, J., Krämer, J., Marson, G. "An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation." In D. Pointcheval, A. Nitaj, T. Rachidi (Eds.), *Progress in Cryptology–AFRICACRYPT 2016, 8th International Conference on Cryptology in Africa.* Berlin: Springer 44–60 (2016) https://doi.org/10.1007/978-3-319-31517-1_3.

[44] Ducas, L., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D. "CRYSTALS–Dilithium: Digital Signatures from Module Lattices." *IACR Cryptology ePrint Archive, 2017* (2017) (accessed 2 October 2018) https://eprint.iacr.org/2017/633.pdf.

[45] Melchor, C. A., Boyen, X., Deneuville, J.-C., Gaborit, P. "Sealing the Leak on Classical NTRU Signatures." In M. Mosca (Ed.), *Post-Quantum Cryptography, 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014.* Berlin: Springer 1–21 (2014) https://doi.org/10.1007/978-3-319-11659-4_1.

[46] Kirchner, P., Fouque, P. "Revisiting Lattice Attacks on Overstretched NTRU Parameters." In J.-S. Coron, J. B. Nielsen (Eds.), *Advances in Cryptology–EUROCRYPT 2017, 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 – May 4, 2017.* Berlin: Springer 3–26 (2017) https://doi.org/10.1007/978-3-319-56620-7_1.

[47] Nguyen, P. Q., Regev, O. "Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures." In S. Vaudenay (Ed.), *Advances in Cryptology–EUROCRYPT 2006, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006.* Berlin: Springer 271–288 (2006) https://doi.org/10.1007/11761679_17.

[48] Ducas, L., Nguyen, P. Q. "Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures." In X. Wang, K. Sako (Eds.), *Advances in Cryptology–ASIACRYPT 2012, 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012.* Berlin: Springer 433–450 (2012) https://doi.org/10.1007/978-3-642-34961-4_27.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

[49] Pessl, P., Bruinderink, L., Yarom, Y. "To BLISS-B or Not to Be—Attacking strongSwan's Implementation of Post-Quantum Signatures." In *CCS '17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.* New York: Association from Computing Machinery 1843–1855 (2017) https://doi.org/10.1145/3133956.3134023.

[50] Howe, J., Poppelmann, T., O'Neill, M., O'Sullivan, E., Guneysu, T. "Practical Lattice-Based Digital Signature Schemes." *ACM Transactions on Embedded Computing Systems* **14.3** 41:1–41:24 (2015) https://doi.acm.org/10.1145/2724713.

[51] Fouque, P.-A., *et al.* "FALCON: Fast-Fourier Lattice-Based Compact Signatures over NTRU." (2018) (accessed 2 October 2018) https://falcon-sign.info/.

[52] Bindel, N., *et al.* "Submission to NIST's post-quantum project: Lattice-Based Digital Signature Scheme qTESLA." (2018) (accessed 2 October 2018) Full list of submissions and associated documentation can be found at https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[53] Petzoldt, A., Bulygin, S., Buchmann, J. "Selecting Parameters for the Rainbow Signature Scheme." In N. Sendrier (Ed.), *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010.* Berlin: Springer 218–240 (2010) https://doi.org/10.1007/978-3-642-12929-2_16.

[54] de Oliveira, A. K. D., López, J., Cabral, R. "High Performance of Hash-Based Signature Schemes." *International Journal of Advanced Computer Science and Applications* **8.3** 421–432 (2017) http://dx.doi.org/10.14569/IJACSA.2017.080358.

[55] Fowler, A. G., Mariantoni, M., Martinis, J. M., Cleland, A. N. "Surface Codes: Towards Practical Large-Scale Quantum Computation." *Phys. Rev. A* **86** 032324 (2012) https://www.doi.org/10.1103/PhysRevA.86.032324.

[56] Matthew, A., Di Matteo, O., Gheorghiu, V., Mosca, M., Parent, A., Schanck, J. "Estimating the Cost of Generic Quantum Pre-Image Attacks on SHA-2 and SHA-3." *IACR Cryptology ePrint Archive* (2016) (accessed 2 October 2018) https://eprint.iacr.org/2016/992.

[57] The factor of 20 for the number of Clifford gates per T gate is based is based on the construction of T gate depth one representations of the Toffoli gate in Selinger, P. "Quantum Circuits of T-Depth One."[19]

[58] Romero, G., Ballester, D., Wang, Y. M., Scarani, V., Solano, E. "Ultrafast Quantum Gates in Circuit QED." *Physical Review Letters* **108.12** 120501 (2012) https://www.doi.org/10.1103/PhysRevLett.108.120501.

[59] Córcoles, A. D., *et al.* "Process Verification of Two-Qubit Quantum Gates by Randomized Benchmarking." *Physical Review A* **87.3** 030301 (2013) https://www.doi.org/10.1103/PhysRevA.87.030301.

[60] Barends, R., *et al.* "Superconducting Quantum Circuits at the Surface Code Threshold for Fault Tolerance." *Nature* **508.7497** 500–503 (2014) https://doi.org/10.1038/nature13171.

[61] Chow, J. M., *et al.* "Implementing a Atrand of a Scalable Fault-Tolerant Quantum Computing Fabric." *Nature Communications* **5** https://doi.org/10.1038/ncomms5015.

[62] IBM. "IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation." (Accessed 2 October 2018) https://www-03.ibm.com/press/us/en/pressrelease/49661.wss.

[63] IBM. "IBM Doubles Compute Power for Quantum Systems, Developers Execute 300K+ Experiments on IBM Quantum Cloud." (accessed 2 October 2018) https://developer.ibm.com/dwblog/2017/quantum-computing-16-qubit-processor/.

[64] Reynolds, M. "Google on Track for Quantum Computer Breakthrough by End of 2017." *New Scientist* (accessed 2 October 2018) https://www.newscientist.com/article/2138373-google-on-track-for-quantum-computer-breakthrough-by-end-of-2017/.

[65] Herr, Q. P., Herr, A. Y., Oberg, O. T., Ioannidis, A. G. "Ultra-Low-Power Superconductor Logic." *Journal of Applied Physics* **109.10** 103903 (2011) http://www.doi.org/10.1063/1.3585849.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

[66] Córcoles, A., *et al.* "Demonstration of a Quantum Error Detection Code Using a Square Lattice of Four Superconducting Qubits." *Nature Communications* **6** 6979 (2015) https://www.doi.org/10.1038/ncomms7979.

[67] Sheldon, S., Magesan, E., Chow, J. M., Gambetta, J. M. "Procedure for Systematically Tuning Up Cross-Talk in the Cross-Resonance Gate." *Physical Review A* **93.6** 060302 (2016) https://www.doi.org/10.1103/PhysRevA.93.060302.

[68] Deng, X.-H., Barnes, E., Economou, S. E. "Robustness of Error-Suppressing Entangling Gates in Cavity-Coupled Transmon Qubits." *Physical Review B* **96.3** 035441 (2017) https://www.doi.org/10.1103/PhysRevB.96.035441.

Appendix A: Estimating Error Correction Resource Overheads for Quantum Attacks

Here we describe how the overhead factors for quantum error correction are calculated in order to obtain resource costs for quantum attacks on blockchains and digital signatures. The method follows the analysis given in Fowler (*et al.*)[55] and Matthew (*et al.*).[56] We first determine n_T and n_C, the number of T gates and Clifford gates respectively needed in the algorithm. The pseudo-code to compute the overhead is given in Table 3. For the blockchain attack on $n_L = 2402$ qubits, these values are

$$n_T = 297784 \times \pi 2^{14} \sqrt{10 \cdot D}, \quad n_C = 29.4 \times n_T.$$

For the Digital Signature attack on $n_L = 2334$ qubits,[57] the values are

$$n_T = 1.28 \times 10^{11}, \quad n_C = 20 \times n_T.$$

If we look some years into the future we can speculate as to plausible improvements in quantum computer technology. If we assume a quantum error correction code that supports transversal Clifford and non-Clifford gates so there is no distillation slow down and that it is done in a measurement free manner so that no classical error syndrome processing is necessary, then the number of cycles needed for one oracle call is determined solely by the circuit depth which is 2142094. This is based on an overall circuit depth calculated as follows. The oracle invokes two calls to the SHA256 hash function, and this is done twice, once to compute it and again to uncompute it. Each hash has a reversible circuit depth of 528768. Similarly, there are two multi-controlled phase gates used, one for inversion about the mean and one for the function call, each having a circuit depth 13511, for a total depth $4 \times 528768 + 2 \times 13511 = 2142094$ (these numbers are from Suchara (emphet al.) but could be further optimized[13]). Then accepting potential overhead in space and physical qubit number, but assuming no time penalty for error correction or non Clifford gate distillation, this implies an improved effective hashing rate of

$$h_{\mathrm{QC}} = 0.04 \times s\sqrt{D}.$$

which is substantially faster. For superconducting circuits, ultrafast geometric phase gates are possible at ~ 50 GHz, essentially limited by the microwave resonator frequency.[58] Using the above very optimistic assumptions, at difficulty $D = 10^{12}$ the effective hash rate would be $h_{\mathrm{QC}} = 2.0 \times 10^3$TH/s.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

function CALCULATEFACTORYRESOURCES(p_g, n_T) ▷ iterates layers of error correction in factory
  $p_{\text{tol}} \leftarrow \frac{1}{n_T}$ ▷ (uncorrected) error tolerance
  $i \leftarrow 0$
  while $p_{\text{tol}} < 10p_g$ **do**
    $i \leftarrow i+1$ ▷ add layer
    $d_i \leftarrow \min\left\{d \in \mathbb{N} : 192d \cdot (100p_g)^{\frac{d+1}{2}} \geq \frac{p_{\text{tol}}}{2}\right\}$ ▷ code distance in this layer
    $p_{\text{tol}} \leftarrow (\frac{p_{\text{tol}}}{70})^{\frac{1}{3}}$ ▷ increased error tolerance
  end while
  layers $\leftarrow i$
  $\tau \leftarrow n_T \cdot 10\sum_{i=1}^{\text{layers}} d_i$ ▷ total clock cycles (only counts T gates)
  $Q_{\text{factory}} \leftarrow 50(d_{\text{layers}})^2 \cdot 15^{\text{layers}-1}$ ▷ total physical qubits for factory
  return $(\tau, Q_{\text{factory}})$
end function

function CALCULATECIRCUITRESOURCES(p_g, n_C, n_L)
  $d_C \leftarrow \min\left\{d \in \mathbb{N} : (80p_g)^{\frac{d+1}{2}} \geq \frac{1}{n_C}\right\}$ ▷ code distance for circuit (single layer)
  return $Q_{\text{circuit}} \leftarrow 3.125n_L d_C$ ▷ total physical qubits for circuit
end function

Table 3. Algorithms to compute space and time resources for quantum attacks. The inputs are p_g, the physical gate error rate; n_C, the total number of Clifford gates in the logical circuit; n_T, the total number of T gates in the logical circuit; and n_L, the number of logical qubits. The outputs are τ, the time cost in number of clock cycles; and $n_Q = Q_{\text{circuit}} + Q_{\text{factory}}$, the number of physical qubits used for the computation including state distillation.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

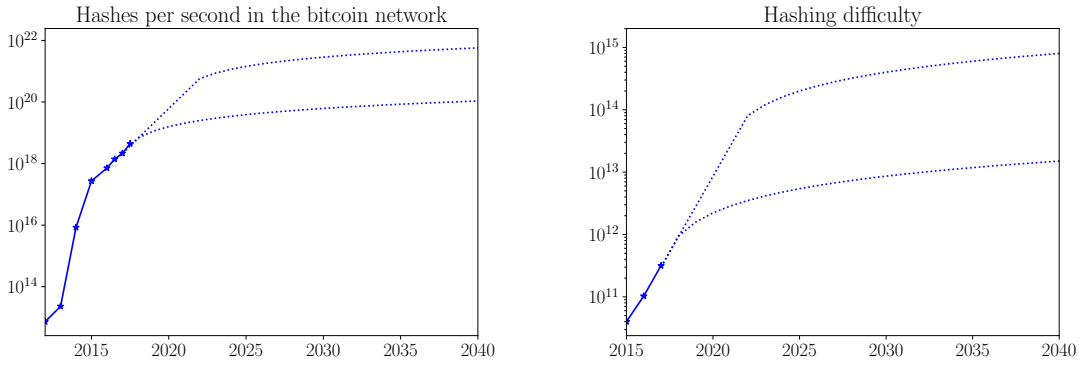


Fig. 5. Prediction of the hash rate of the Bitcoin network (in number of hashes per second) and the hashing difficulty as a function of time.

Appendix B: Modeling the Development of Bitcoin Network Difficulty

The total number of hashes per second in the Bitcoin network are taken from blockchain.info. The data points in Figure 5a are the hash rates for the first of January (2012–2015) and first of January and July (2016–2017). The two dotted curves correspond to optimistic and less optimistic assumptions for the extrapolations. The optimistic extrapolation assumes that the present growth continues exponentially for five years and then saturates into a linear growth as the market gets saturated with fully optimized ASIC Bitcoin miners. The less optimistic assumption assume linear growth at the present rate.

From the extrapolation of the Bitcoin network hashrate we can determine the difficulty as a function of time. The expected number of hashes required to find a block in 10 minutes (600 seconds) is given by $\text{rate}(t) \cdot 600$, where $\text{rate}(t)$ is the total hash rate displayed in Figure 5a. Thus the Bitcoin hashing difficulty is calculated as $D(t) = \text{rate}(t) \cdot 600 \cdot 2^{-32}$ for the two scenarios discussed above. In Figure 5b we compare this with values from blockchain.info for the first of January of 2015–2017.

Appendix C: Modeling the Development of Quantum Computers

There are several aspects of the development of quantum technologies that we must model. Since only few data points are available at this early stage of the development there is necessarily a lot of uncertainty in our estimates. We therefore give two different estimates, one that is optimistic about the pace of the development and another one that is considerably more pessimistic. Nonetheless, these predictions should be considered as a very rough estimate and might need to be adapted in the future.

First, we need to make an assumption on the number of qubits available at any point of time. As we focus only on solid state superconducting implementations there are only a few data points available. We assume that the number of available qubits will grow exponentially in time in the near future. The optimistic assumption is that the number will double every 10 months whereas the less optimistic assumption assumes the number doubles every 20 months. These two

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

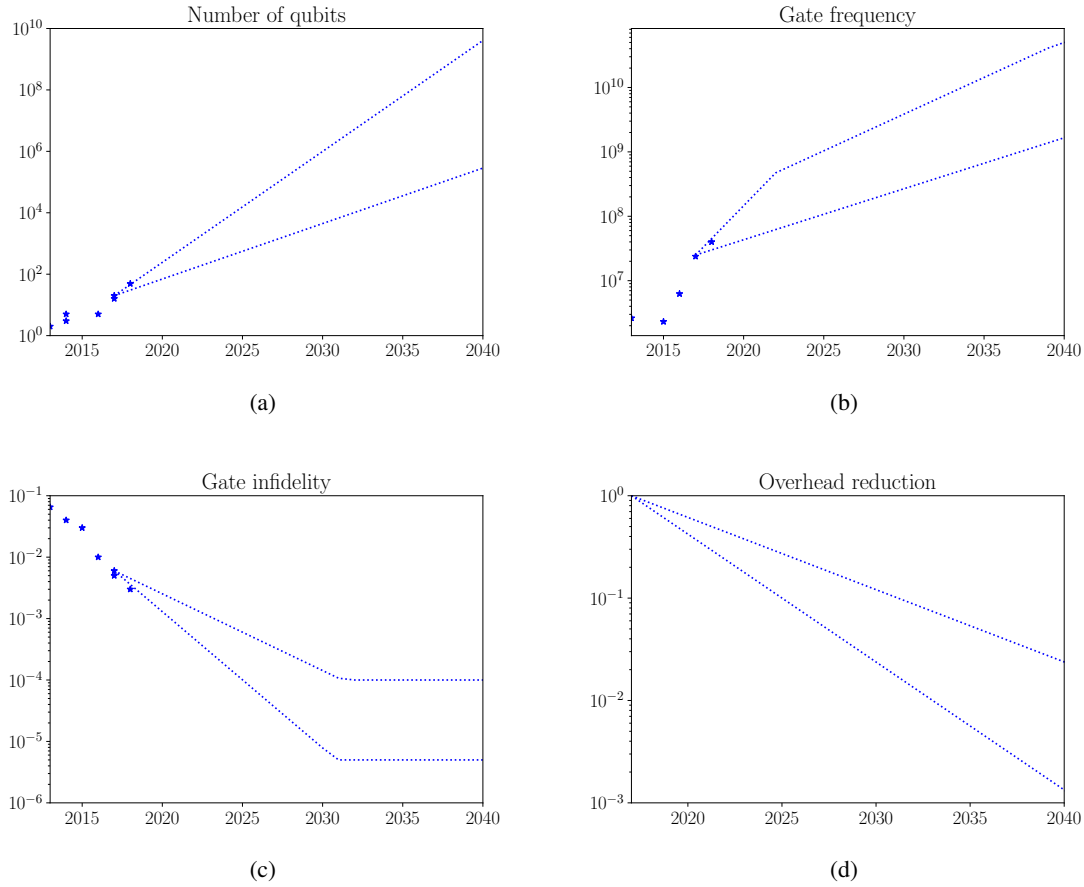


Fig. 6. Prediction of the number of qubits, the quantum gate frequency (in gate operations per second) and the quantum gate infidelity as a function of time. The fourth plot models a reduction of the overhead due to theoretical advances.

extrapolations are plotted in Figure 6a. The data points are taken from the following table:

number of qubits	year	reference
2	2013	Córcoles, A. D. (*et al.*) [59]
5	2014	Barends (*et al.*) [60]
3	2014	Chow (*et al.*) [61]
5	2016	IBM (2016) [62]
16	2017	IBM (2017) [63]
20	2017	Reynolds [64]
49	2018	Reynolds [64]

We predict that the quantum gate frequency grows exponentially for the next years. This assumes that the classical control circuits will be sufficiently fast to control quantum gates at this frequencies. After a couple of years the growth slows down considerably because faster classical control circuits are necessary to further accelerate the quantum gates. We cap the quantum gate

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127

frequency at 50 GHz (for the optimistic case) or 5 GHz (for the less optimistic case), respectively, mostly because we expect that classical control circuits will not be able to control the quantum gates at higher frequencies. (See, *e.g.*, Herr (*et al.*) for progress in this direction.[65]) This is shown in Figure 6b. The data points are taken from the following table:

gate time	year	reference
420ns	2013	Córcoles, A. D. (*et al.*) [59]
433ns	2015	Córcoles, A. D. (*et al.*) [66]
160ns	2016	Sheldon (*et al.*) [67]
42ns	2017	Deng (*et al.*) [68]
25ns	2018	Google, projected for end of 2017

The predicted development of the gate infidelity is shown in Figure 6c. We assume that the gate infidelity will continue to drop exponentially but that this development will stall at an infidelity of $5 \cdot 10^{-6}$ (optimistic case) or $5 \cdot 10^{-5}$ (less optimistic case). For the optimistic case we expect that the gate infidelity will continue to follow DeVincenzo's law which predicts a reduction of the infidelity by a factor of 2 per year. The data points are taken from the following table:

gate fidelity	year	reference
0.9347	2013	Córcoles, A. D. (*et al.*) [59]
0.96	2014	Chow (*et al.*) [61]
0.97	2015	Córcoles, A. D. (*et al.*) [66]
0.99	2016	Sheldon (*et al.*) [67]
0.995	2017	Reynolds [64]
0.997	2018	Reynolds [64]

Finally, we assume that the number of qubits and time steps required by any algorithm will be reduced over time for two reasons. First, the gate fidelity will increase over time and thus allow for more efficient fault-tolerant schemes to be used. Second, theoretical advances will allow to decrease the number of qubits and gates required to implement the algorithm and fault-tolerant schemes. We expect that this factor will be $\text{overhead}(t) = \beta^{t-2017}$ where $\beta \in \{0.75, 0.85\}$ for optimistic and less optimistic assumptions, respectively.

SSN 2379-5980 (online)
DOI 10.5195/LEDGER.2018.127