Assessment of Quantum Threat To Bitcoin and Derived Cryptocurrencies

Stephen Holmes, Liqun Chen

University of Surrey

Abstract—All cryptocurrencies are not the same. Today, they share a common quantum vulnerability through use of non-quantum safe Elliptic Curve Digital Signature Algorithm (ECDSA) digital signatures yet they have very different risks of quantum attack. The risk of attack for a cryptocurrency depends on a number of identified factors such as the block interval time, the vulnerability to an attack that delays the time for an unprocessed transaction to be completed and the behaviour of a cryptocurrency user to increase the cost of a quantum computer attack. Shor's algorithm can be used to break ECDSA signatures with a quantum computer. This research addresses the two questions: When will a quantum computer be powerful enough to execute Shor's algorithm? How fast would a quantum computer need to be to break a specific cryptocurrency? In this paper we observe that by benchmarking the speed of circuits and the time for quantum addition on quantum computers we can determine when there is a potential threat to a specific cryptocurrency.

Keywords—digital signatures, quantum computing, cryptocurrency, bitcoin, altcoin, ECDSA

I. Introduction

THE bitcoin white paper was published in 2008 [1] and since that time, bitcoin and alternative currencies have grown in value and volume. Bitcoin and cryptocurrency systems provide a new means for multiple mutually distrusting and remote parties to transact through a protocol that enforces a level of trust between parties because of the immutability of the protocol and consensus mechanism. Bitcoin and altcoins derived from bitcoin architecture current value is over 100 billion USD.

Underpinning bitcoin and alternative cryptocurrencies are a series of cryptographic technologies used to secure bitcoin and alternative cryptocurrencies. With the advent of a future quantum computer many underpinning cryptographic protocols become susceptible to attack by the development of a sufficiently large quantum computer. More specifically, digital signature schemes, for example ECDSA [2], used in cryptocurrencies are not quantum resistant.

This research is focused on the assessing the potential mechanisms for attack that are possible by a quantum computer to break digital signatures used in bitcoin and derived cryptocurrencies (altcoins).

A public key of such a digital signature scheme is used as an address of an amount of cryptocurrency funds. The corresponding private key(s) indicates the ownership of the funds. This research highlights that if the public key has been disclosed or if it has been used for more than one transaction, allowing it to be recorded for later quantum attack, the mitigation is simply to move funds to a new address. Most currently supported wallet software can be configured to automate this approach making this almost invisible to users.

In order to keep the disclosure time of a valid address in a cryptocurrency blockchain as short as possible, a public key can be hidden by using a cryptographic hash function or a pair of two hash functions, e.g. in bitcoin two hash functions SHA256 and RIPEMD160 are used [3].

The quantum attack that is a common threat to all ECDSA-based cryptocurrencies can only be performed during the disclosure of a public key and signature to enable movement of cryptocurrency funds. A window of opportunity exists until the unprocessed transaction is included in a block.

ISO are currently developing a technical report for blockchain security - Blockchain and distributed ledger technologies Security risks, threats and vulnerabilities [4], A current proposed approach from Homoliak, Ivan & Venugopalan, Sarad & Hum, Qingze & Reijsbergen, Daniel & Schumi, Richard & Szalachowski, Pawel [5]. The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses presents a Security Reference Architecture (SRA) for blockchain which introduces a threat-risk model that is based on the template of ISO/IEC 15408 [6]. Using this stacked model of the security reference architecture, focus is on the replicated state machine layer and specifically transactions.

The wide adoption of ECDSA for blockchain signature schemes is understandable given the proven security against todays computers and the efficiency of the signature scheme (time to execute and storage space required for keys and signatures). This gives rise to the primary question of this research - How can we measure quantum computing advances regardless of the technology they are built from to assess the threat to bitcoin and derived cryptocurrencies? What quantum computer performance indicators can give advanced warning of the threat to the ECDSA signature scheme in a cryptocurrency implementation?

II. Contributions

THIS paper is organised as follows: An overview of related work on bitcoin and cryptocurrency vulnerability to quantum computer attack. Analysis and review of possible combined attacks to delay the time to process an unprocessed transaction. Review of estimating quantum computing resources to solve ECDLP using implementation of Shor's algorithm [7]. Assessment of the quantum computer capacity required to execute Shor's algorithm, taking into account circuit size and error rates on a quantum computer. A new proposed approach to assess unprocessed transaction time

window for each cryptocurrency and derive quantum computer speed of execution required to meet this time window. The impact of noise and error rates on quantum computers and an examination of quantum computers capabilities today.

The key discovery from this paper is that not all cryptocurrencies have the same timing attack vulnerabilities. A consequence of this is that for some cryptocurrencies migration from ECDSA to quantum safe signatures may be possible to delay until a quantum computer is approaching the capacity and performance required to pose a threat.

A further discovery is that cryptocurrency user behaviour can significantly impact the cost of a quantum adversary attack. The migration to one time use addresses is a practical protection that prevents a slow quantum computer attack on a disclosed public key. A further protection is to move to multi-signature addresses. Where n is the number of signatures required to unlock an address. In the case of Bitcoin, this enables up to n of 20 signatures to be required to unlock an address. Making the quantum adversary solve n ECDSA private keys from the public keys disclosed in the unprocessed transaction. Increasing the quantum computing requirement by n times within the same unprocessed transaction window. Paying a significantly higher fee than the current average transaction fee increases the cost to a quantum adversary in executing a denial of processing valid transactions attack. Reducing the amount of cryptocurrency in any one account by spreading currency to multiple accounts, reduces the risk and attractiveness of an attack.

This can stave off a quantum computer attack. However, this is at the expense of usability and reliance on user operational discipline. Enabling a cryptocurrency to decide when to implement countermeasures such as moving to accepting only multi-signature transactions or implementing new means to reduce the unprocessed transaction time.

The quantification of the unprocessed transaction time window by cryptocurrency and the quantification of the number and depth of gates required to run Shor's algorithm enables a benchmarking of quantum computer capabilities required to meet or exceed this window and hence a risk assessment of quantum capability to make a successful attack on a specific cryptocurrency. This simple measurement can give advanced warning that a cryptocurrency may be at risk regardless of quantum computing technology.

III. Related work

THE threat posed by quantum computers to bitcoin and how to protect against them is described by Aggarwal D et al (2017) [8]. In the paper attacks on digital signatures are identified as a major threat vector, as elliptic curve discrete log problem is a hard problem for classical computers but using Shor's algorithm on a quantum computer is no longer hard for a sufficiently large quantum computer, given a public key for a signature, to calculate the private key. Aggarwal et al go on to classify three scenarios in bitcoin.

A. 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.

B. 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. Even with a quantum computer a double spending attack is unlikely once the transaction has many blocks following it.

C. 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.

The mitigation to protect against reusing an address is simple. However, in practice this will need to be achieved through either programatic wallet rules or disciplined changes in user behaviour. This makes every address a one-time use address (receiving and sending). Currently for bitcoin this stands at around 48.6% as of August 2020. Analysis of today's bitcoin wallet addresses shows that there are currently 2476 wallet addresses with bitcoin funds worth over 10 million US dollars. Yet, despite the advice to only use an address once (to prevent recording of public key by rogue miner, or intercepting across the network). Only 81 of these addresses only have one transaction. The remaining 2395 addresses have multiple transactions meaning that the public key can be recorded and broken at leisure by a quantum computer adversary to generate malign transactions.

Figure 1 illustrates that the practice of not reusing an address is still the default for most users. In the earliest version of bitcoin public key addresses were used directly - P2PK (Pay to Public Key). These early transactions are vulnerable to a quantum computer attack as the public key is exposed. In 2010 the P2PKH (Pay to Public Key Hash) was introduced and quickly became dominant. This prevents disclosure of public key address on the blockchain. The Hash function of the public key provides protection against a quantum computer deriving the public key. The algorithm used by bitcoin is to double hash the public key SHA256 followed by RIPEMD160. In April 2012 Pay to Script Hash P2SH was introduced in bitcoin, enabling scripts to execute that require multiple ECDSA signatures to unlock and move coins. P2SH uses

Figure 1: Bitcoin address re-use (percentage)[10]

public key hash algorithm as P2PKH and so protects against disclosure of the public key until a transaction is submitted. In 2015 bitcoin further extended this with Pay to Witness Public Key Hash (P2WPKH) which addressed the inherent transaction malleability issue with including hash of ECDSA signature in transaction. This combines with Pay to Script Hash. In common all these payment methods require a public key or multiple public keys to be presented to unlock and pay bitcoins. Apart from the original Pay to Public Key (P2PK) they all use double hashing to protect the public key until a transaction is submitted.

Unprocessed transactions are the most serious attack if the time taken for a quantum computer to compute the ECDSA private key can be done at close to the bitcoin block interval. This is because this attack can only be prevented by moving to a quantum resistant digital signature scheme.

In order to claim and unlock bitcoins the public key has to be disclosed to the nodes. Until the block is mined the disclosed key can allow a quantum adversary to calculate the private key and enter or change the transaction to sign a replacement or duplicate transaction.

IV. Combined hybrid attack scenario

THE first phase of an attack is to identify if an attack on an unprocessed transaction is worthwhile. Given the anticipated cost of conducting an attack, it is necessary to identify a worthwhile target. When a transaction is submitted to the network of miners it contains the 'from' address and the amount. High value addresses can be identified offline and if the 'from' address matches one of these high value addresses, or the amount is over a threshold then this is a candidate for attack.

Privacy cryptocurrencies such as Zcash provide shielded addresses that are not visible and transactions between shielded addresses do not reveal either address, the transaction amount or the contents of the encrypted memo field. This makes an attack on such a cryptocurrency probabalistically costly to a quantum equipped adversary. Without knowledge of the value held in an address, It is a gamble if the address under attack actually contains sufficient value of cryptocurrency to make this worthwhile. However, Zcash uses non-quantum safe Zero knowledge proofs today and structured reference strings that are public information which contain trapdoors that can be extracted by solving the ECDLP problem using Shor's algorithm enabling an attack to print money and remove anonymity from transactions. We do not examine this quantum threat in this paper and simply note this as a potential threat applicable specifically to the current Zcash implementation.

Unprocessed transactions are held in bitcoin in the mempool (memory pool) for each mining node. The mining node provides validation of transactions and protects against both malformed transactions and potential DDoS transaction attacks. This is achieved by performing the following checks before adding a transaction to the mempool:

- Verifying if an originating and outbound address is stated
- Verifying if the size is less than the maximum block size
- Verifying if the value and total amount are within correct ranges
- Checking for the 'n=-1' or 'hash=0' from Coinbase.
- Checking if duplicate transactions are already in the pool
- Checking the size and values of 'nLockTime'
- Checking unusual behaviour from 'scriptSig' or 'scriptPubkey'
- Keeping track of orphaned transactions
- Rejecting input sums that don't correctly match output sums
- Checking if fee meets minimum transaction fee

In the bitcoin network, each node maintains their own local mempool and because of timing and communication issues, mempools are not guaranteed to be identical. A miner who has solved the latest Proof of Work (PoW) challenge is the node that chooses which transactions are included in the next block from the local mempool. The miner chooses the transactions from the mempool by evaluating the maximum number of transactions that can be included in a block yielding the maximum transaction fee (reward).

Saad et al. [9] describe mempool DDoS attacks in PoW-based blockchain systems and suggest some possible mitigations. However, the mitigations proposed require the miners to change the way they pick unprocessed transactions and this to be adopted as a standard method. The focus on the research was on the transaction fee manipulation to trap legitimate users into paying higher fees or to flood the mempool with 'dust' transactions that will clog up a mempool. In this research this attack is extended to legitimate transactions that have 'normal' values that can flood the mempool and block the target transaction from being chosen by a miner for the next block. The mitigations suggested provide some level of protection by including age value for a transaction choice. However, as these 'sybil' transactions are legitimate transactions, this only provides limited protection.

For a quantum attack to be successful with a one-time use public key address, the adversary would need to derive the private key from the public key in the unprocessed transaction

before the transaction is added in a block. Because the private key enables any new transaction to be signed by the adversary, the original transaction simply needs to be deferred to enable a new forged transaction to be added to and executed from the mempool.

This could be done by either replacing the transaction in the mempool of the winning miner or creating a new transaction with a higher transaction fee to ensure this gets processed ahead of the original transaction. In order to generate a valid transaction, a quantum adversary would need to derive a private key from the disclosed public key in the transaction originally provided. Bitcoin currently supports the Replace by Fee protocol to enable a“stuck" transaction to have additional fee added to make the transaction more commercially interesting for a miner to choose when they mine the next block.

An attacker when they receive a transaction can either check the amount of bitcoins in the from address, or check if the address corresponds with one with over a number of bitcoins in the account. If there are a sufficiently large amount of bitcoins then the public key corresponding to this transaction could be chosen as worthy of running a quantum calculation of the private key.

The adversary could then prevent this chosen transaction from being processed by flooding the network (and mempool) with transactions that have higher fees than the target transaction. In a denial of processing valid transactions attack.

This could be done using sybil accounts and the effect would be difficult to identify as the volume increase and transaction fee adjustment would be a spike that is not uncommon on the bitcoin network, The effect of this is a denial of processing for the original transaction and this could extend for multiple block interval epochs before detection.

The attack can be created offline with the set of sybil accounts and transactions created and ready to inject into the bitcoin network when required.

Given the maximum block size of 2MB for bitcoin, the maximum number of transactions can be calculated. Miners will choose transactions based upon their best return, so smaller and high fee paying transactions will be chosen first. Typical size per transaction is 250 bytes with minimum size of a transaction 63 bytes in bitcoin. So, if we assume that the transactions are crafted to be 240 bytes then this would make them more attractive than the average transactions. This would also require a maximum of

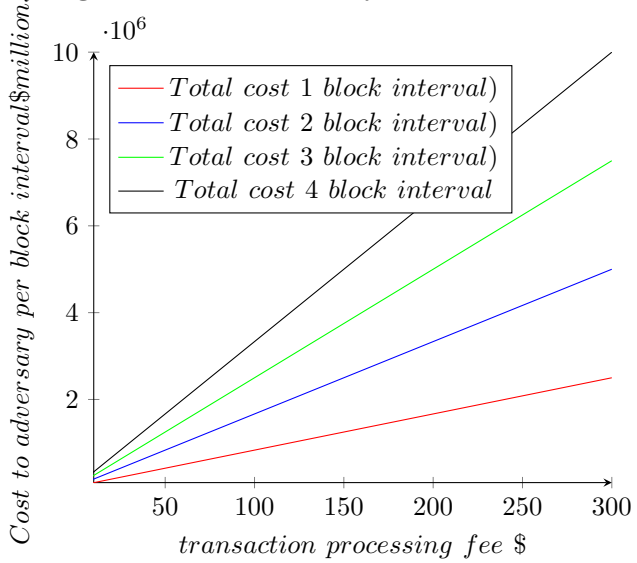
$$\text{Maximum txns per block} = \frac{size\ of\ mempool\ (bytes)}{size\ per\ transaction(bytes)}$$

$$\text{Maximum txns per block} = 2000000/240 \text{ bytes} = 8333 \text{ txns.}$$

An adversary could continue this attack without detection over multiple block interval windows with the cost increasing with every block interval deferred in order to extend the time available to break the private key(s) from the victim transaction. One way of making the cost higher for an adversary would be for the high value transaction to pay a significantly higher fee for the transaction. For example, if the transaction is placed with a $250 fee then this increases the quantum adversary cost by a factor of the increased transaction fee. The transaction fees would need to be higher for the sybil transactions to defer a block and in this case the fee would be Block attack transaction fee costs $= \$250 \cdot 8333 = \$2.083\ million\ USD$.

Figure 2 illustrates that through selecting a significantly higher transaction fee then the cost to an adversary per block interval is significant. If $250 transaction fee is paid for a multi million dollar transaction then the cost to execute this delayed processing attack is approximately $ 2m per block interval. Significantly discouraging a quantum adversary through increasing the cost of an attempted attack.

Figure 2: Cost to adversary for each block interval



This requires a change of behaviour by the user placing the high value transaction through acceptance of a higher fee for the transaction than required in today's non quantum computer environment. An alternative approach is to reduce the maximum amount held in any one address to reduce the attractiveness to a quantum attacker.

Given the limited storage for most bitcoin nodes (300mb for mempool typically) it is possible to over-run and ensure that the original transaction could be evicted from the mempool. This essentially gives a quantum attacker an infinite amount of time to perform the attack. However, this is highly visible on the network. Therefore it is considered that a quantum adversary would most likely take the option of delaying the transaction by the number of block intervals required to solve the private key calculation from the public key.

The block interval will vary depending upon the level of difficulty which impacts the time to mine a new block using proof of work as a consensus algorithm. For bitcoin the historical average block interval time is 10 minutes. However, this is an average and there are outliers that are faster or slower (26 minutes as an example). This is because the cryptographic proof of work algorithm is non-deterministic, finding a suitable nonce to create an acceptable hash is a random process that takes an average amount of time. Another factor that will impact the block interval rate is the pool of available hash

Cryptocurrency	Digital Signature	Parameters	Consensus Algorithm	Difficulty adjustment Mitigation	Block interval time target
Bitcoin (2009)	ECDSA	Secp256k1	Proof of Work	None	10 mins
Litecoin (2011)	ECDSA	Secp256k1	Proof of Work	None	2.5 mins
Namecoin (2012)	ECDSA	Secp256k1	Proof of Work	Deterministic salt commitment	10 mins
Dogecoin (2013)	ECDSA	Secp256k1	Proof of Work	None	1 min
Primecoin (2013)	ECDSA	Secp256k1	Proof of Work (chain of primes)	Difficulty adjusted every block	1 min
Auroracoin (2014)	ECDSA	Secp256k1	Proof of Work	Every 8 blocks	1 min
Dash (2014)	ECDSA	Secp256k1	Proof of Work	Dark Gravity Wave	2.5 mins
Vertcoin (2014)	ECDSA	Secp256k1	Proof of Work	Every Block	2.5 mins
Ethereum (2015)	ECDSA	Secp256k1	Proof of Work	None	15 seconds
Zcash (2016)	ECDSA EDDSA	Secp256k1 (transparent) Ed25519 (shielded)	Proof of Work	None	75 seconds
Bitcoin Cash (2016)	ECDSA	Secp256k1	Proof of Work	Emergency Difficulty Adjustment	10 mins

Table 1: Analysis of bitcoin derived Proof of Work cryptocurrencies

power provide by the bitcoin miners. This has historically grown over time. However, in the case of bitcoin, the hardness of the cryptographic challenge (bitcoin halving) and the ever reducing number of bitcoins left to mine may result in a reduction in mining hash rates as miners will increasingly only gain benefit from confirming transactions.

The Bitcoin network has a moving network difficulty that is related to the total hash rate available. This is adjusted every 2016 blocks (approx. every 2 weeks historically) so that the average time block interval remains 10 minutes.

The miners in a bitcoin network are rewarded for each new block mined with a number of bitcoins. This reward initially was set at 50 bitcoins per block mined and after every 210,000 blocks mined the reward halves. Today the reward is 6.25 bitcoins per block mined. While the value of a bitcoin increases, this will offset the loss from a reduction in mining reward. This will in turn reflect the number of mining participants and the hash rate pool available.

Today there are many cryptocurrencies that are forks of the bitcoin core code. These cryptocurrencies all use Elliptic Curve digital signatures (ECDSA). These cryptocurrencies also use the same proof of work algorithm and so have the same vulnerability as classic bitcoin. A list of these cryptocurrencies are shown in Table 1.

Figure 3: Historic Daily Block interval time (minutes) [10]

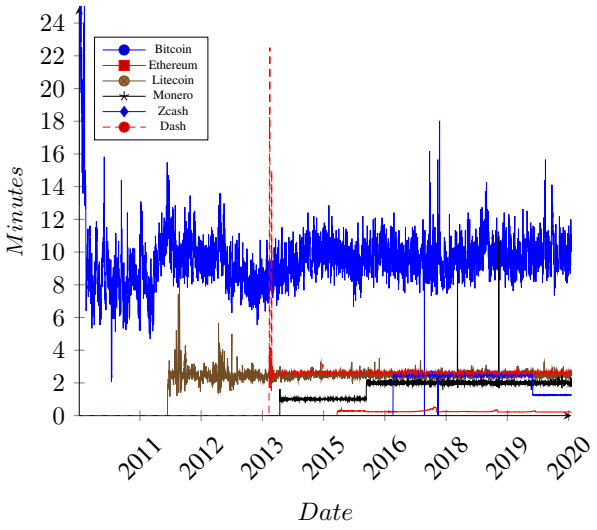


Figure 3 shows the average block time for Proof of Work based cryptocurrencies using ECDSA signatures. Bitcoin and bitcoin cash have on average 10 minutes between blocks. For other blockchain systems using a proof of work consensus algorithm. such as Ethereum, the average historical block interval is 14 seconds, Zcash 2.5 minutes, Litecoin and Dash 2.5 minutes. In order for an adversary to be successful in subverting a blockchain network that uses ECDSA signatures and a Proof of Work (PoW) consensus mechanism, the adversary would need to be able to calculate the ECDSA private key from the public key within this time window.

One issue with use of the average block interval time is what would happen if the pool of miners suddenly reduced?

Meshkov, Chepurnoy and Jensen [11] highlight the potential to manipulate the difficulty through miners withholding hash power at the end of a bitcoin epoch, by switching hash power to other proof of work cryptocurrencies in what they describe as a coin hopping attack. This attack is to give an advantage to an adversarial miner when the new bitcoin difficulty adjustment takes place in the next epoch.

In this paper, we propose inverting this attack vector. This attack can be adapted to add additional hash power, rather than reducing the hash power as proposed by Meshkov, Chepurnoy and Jensen [11] at the end of the epoch for bitcoin to ensure the difficulty is higher for the next epoch and then withdrawing the hash power at the start of the epoch to adversely impact the block interval time to give a quantum adversary the maximum time to calculate private keys from public keys.

In Bitcoin Cash, this is addressed with a new protocol the Emergency Difficulty Adjustment (EDA) to maintain the average block interval time in this scenario, Dash implements Dark Gravity Wave (DGW) that adjusts the difficulty level after every block.

This illustrates that even with the same fundamental architecture, the challenge for an adversary will be very different depending upon the time budget they have to calculate the private key from the transaction public key.

Given that these cryptocurrencies enable an adversary to set up a full participant node, which gives an adversary access to transactions that include public keys transmitted over the network which are not hashed, today an adversary does not need to evesdrop on a communication to collect this information. However, if the nodes are restricted, then an adversary would need to have the capability of evesdropping on the node to node traffic and the user to node transaction. Today, this is protected at a minimum with Transport Layer Security (TLS), which, is non-quantum safe implementation. This would add to the burden of an adversary but, given that TLS is session based, it is also a viable option to intercept transactions sent between nodes. It is anticipated that post-quantum TLS will be available in the near future to mitigate this attack.

V. Estimating Quantum Computing Resources to solve ECDLP

THE fundamental quantum algorithm that speeds up the ability to solve the Elliptic Curve Discrete Logarithm Problem is the Shor Algorithm [7]. Shor's algorithm enables solving ECDLP problems in polynomial time.

Proos and Zalka [12] converted Shor's algorithm into a practical implementation for a quantum computer using quantum circuits and qubits. The net result of this was a formula to calculate both the number of qubits required for a quantum computer to execute Shor's algorithm and the number of quantum gates (time) to execute this. In the ECDSA algorithm n is the size of the private signing key.

Proos and Zalka initial high level estimation for the number of qubits required to execute Shor's algorithm on a quantum computer is: (roughly) $6n$ where n is the number of bits in the ECDLP problem. Hence an initial estimate for the number of qubits required to execute and solve the ECDSA algorithm with a 256-bit private key is 6 x 256 = 1536 qubits.

The number of circuits measured in terms of quantum additions is also calculated for the algorithm and a formula derived for Time to execute DLP algorithm. This formula is: $T = 360 \cdot kn^3$ Where k is the time taken for each quantum addition

Thus the estimate for time to execute Shor's DLP algorithm is dependent upon the number of bits n and the time taken for each 1-qubit quantum addition k. thus the number of quantum gates in an addition network per length of the registers involved. This number is about 9 quantum gates, 3 of which are Toffoli gates.

Thus for 256 ECDSA (n=256) and with a time of k determined by the clock cycle time megahertz cycle multiplied by the number of gates per 1-qubit quantum addition (9). Jordan [13] makes the case that energy considerations alone are not sufficient to obtain an upper bound on computational speed. This enables quantum computers to have clock speeds in excess of today's classical computers. A high performance computer today will have a clock speed of between 4 to 5 GHZ.

Assuming a 5 Ghz clock cycle time. The number of cycles per second for 5Ghz is $5 \cdot 10^9$ for each addition it requires 9 quantum gates and 9 cycle times. Therefore the time per addition k will be $k = \frac{9}{(5 \cdot 10^9)}$ seconds. $T = \frac{360 \cdot 9 \cdot 256^3}{5 \cdot 10^9}$ thus $T = 10.87 seconds$.

To solve the 256 bit ECDSA private key from the public key within less than 10 minutes to successfully attack bitcoin, the minimum quantum addition compute time can be calculated: Where ct is the compute time for an addition which is made up of the 9 cycles and quantum computer clock cycle time in Hertz

$$600 = 360 \cdot ct \cdot 256^3$$

$$ct = \frac{600}{360 \cdot 256^3}$$

This compute time for an addition is made up of the number of cycles required per addition (9) and the speed of each cycle in a quantum computer cy in hertz.

$$ct = \frac{9}{cy}$$

Figure 4: Quantum computer clock speed (Hertz) to meet unprocessed transaction time

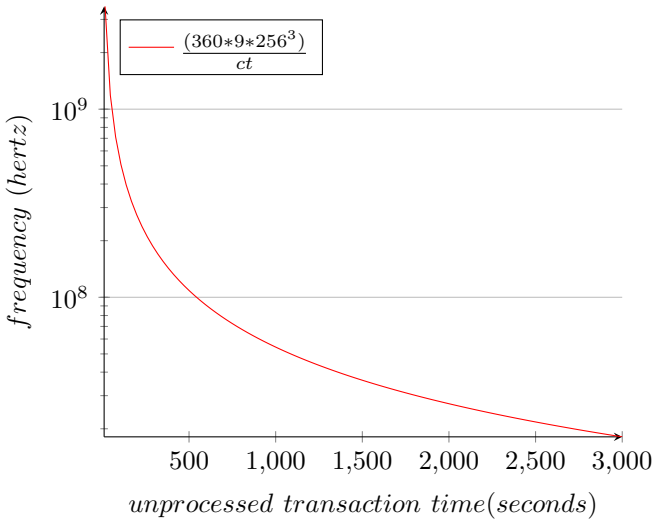


Figure 4 illustrates the quantum addition time required to solve the ECDSA problem using Shor's algorithm before the block interval budget is reached. This gives a benchmark of the computing speed required for an addition on a quantum computer in order to calculate the private key from the public key within the timeframe.

Roetteler, Naehrig., Svore, Lauter [14] refine this implementation and estimation to calculate a more exact figure for the number of qubits and express the calculation time as a function of Toffoli gates.

The calculation of qubits is defined as

$$q = 9n + 2[log_2(n)] + 10$$

This gives a higher number of qubits required to practically implement Shor's algorithm.

$$q = 9.256 + 2[log_2(256)] + 10$$

$$q = 2330$$

The number of quantum bits required is 2330 versus Proos and Zalka estimation of 1536 qubits.

Roetteler et al. [14] provide an estimate for the scaling of the number of Toffoli gates in Shor's ECDLP algorithm in their results and for 256 bits report 1.26×10^{11} Toffoli gates.

However, in order to account for some parallel activities that can take place in the algorithm, the Toffoli depth measure was developed. This reduces the total number of Toffoli circuits required to 1.16×10^{11} and a reported simulation time of 3848 seconds.

Häner et al. [15] in their paper build upon the work by Proos and Zalka and Roetteler et al to provide a more optimised approach for implementing Shor's algorithm. This set of optimisations focuses not just on reducing the number of qubits but also the number of gates and circuit depths. A set of trade-offs is presented and a revised set of resource estimates are provided. This reduces the number of qubits required for ECDSA P256 curve from 2338 to 2124 for qubit optimised, 2619 for gate optimised and 2871 for circuit depth optimised.

Optimising for low qubits, the circuit for low width (qubits) to solve ECDLP on n-bit elliptic curve is approximately $8n + 10 \cdot [\log n] - 1$ logical qubits using roughly $436n^3 - 1.05 \cdot (2^{26})$ T-gates at a T-depth of $120n^3 - 1.67 \cdot 2^{22}$. The total number of gates is $2900n^3 - 1.08 \cdot 2^{31}$ with depth $509n^3 - 1.84 \cdot 2^{27}$.

Optimising for low T-gates, the (qubits) to solve ECDLP on n-bit elliptic curve is approximately $10n + 7.4 \cdot [\log n] + 1.3$ logical qubits using roughly $1115n^3 / \log n - 1.08 \cdot (2^{24})$ T-gates at a T-depth of $389n^3 / \log n - 1.70 \cdot 2^{22}$.

Optimising for low depth the (qubits) to solve ECDLP on n-bit elliptic curve is approximately $11n + 3.9[\log n] + 16.5$ qubits with the number of T-gate $2523n^3 + 1.10 \cdot 2^{20}$ T-gates at a T-depth of $285n^2 - 1.54 \cdot 2^{17}$.

Häner et al. [15] show the number of T-gates as the dominant gates

A core assumption is that the number of T-gates although important, is not as important as the T-gate depth. This is because the number of T-gates does not significantly impact the execution time as they can in most cases execute in parallel. However, the depth of the T-gates is a measure of non-parallel operations, so assessing the depth of the T-gates gives an approximation for the speed of execution. A further assumption is that non T-gates can execute in parallel.

Maslov [16] reported a Toffoli circuit benchmarked implementation time of 1285μs. Advances in quantum computing are rapidly evolving and the performance is rapidly increasing.

VI. Combined approach

OUR proposed combined approach synthesis both the identified attack vectors with the individual cryptocurrency implementation. Enabling the proposal of a set of performance indicators that a quantum computer would need to exceed to alter an unprocessed transaction - either replacing the transaction or adding additional transactions through the derivation of the private key for the digital signature and submitting new transactions.

For bitcoin and derived alternative currencies, the time window before an Unprocessed transaction is fixed within a block is dependent upon a number of factors.

1) What is the system target block interval target?
2) What approaches could be used to influence this (dynamic versus lagging).
3) Can a denial of transaction processing attack be performed to increase the number of block intervals required before the transaction is fixed within a block?

Applying the algorithm from Häner et al. [15] and the implicit number of T-gates required to calculate the ECDSA private key from a given public key, we can calculate the time benchmark per T-gate in order to achieve this within the time budget.

However, by conducting a denial of processing valid transactions attack, this could delay the time to process the transaction by a number of blocks. If this was 2 blocks then we would have a time budget of 30 minutes and if this was combined with a difficulty adjustment attack the block interval could be extended further on the final block interval time, for example in bitcoin the removal of 10 percent hash rate could increase the time to mine a block from 10 minutes to 11 minutes. Giving a total of 31 minutes to calculate the private key for the ECDSA signature.

Cryptocurrency	Difficulty adjustment Mitigation	Block interval time	Assessed block interval manipulation timing risk	Assessed denial of unprocessed transaction risk	Assessed unprocessed transaction time
Bitcoin (2009)	None	10 mins	High	High	31 Mins
Litecoin (2011)	None	2.5 mins	High	Medium	10 Mins
Namecoin (2012)	Deterministic salt commitment	10 mins	High	High	31 Mins
Dogecoin (2013)	None	1 min	Medium	Low	5 Mins
Primecoin (2013)	Difficulty adjusted every block	1 min	Low	Low	3 Min
Auroracoin (2014)	Every 8 blocks	1 min	Low	Low	3 Min
Dash (2014)	Dark Gravity Wave	2.5 mins	Low	Low	7.5 Min
Vertcoin (2014)	Every Block	2.5 mins	Low	Medium	7.5 Min
Ethereum (2015)	None	15 seconds	Low	Low	45 Sec
Zcash (2016)	None	75 s	Low	Low	225 Sec
Bitcoin Cash (2016)	Emergency Difficulty Adjustment	10 mins	Medium	High	31 Mins

Table 2: Analysis of cryptocurrency unprocessed transaction time under hybrid attack

In order to calculate the performance required for a quantum computer to be successfully applied to successfully alter an unprocessed transaction we need to identify the following:

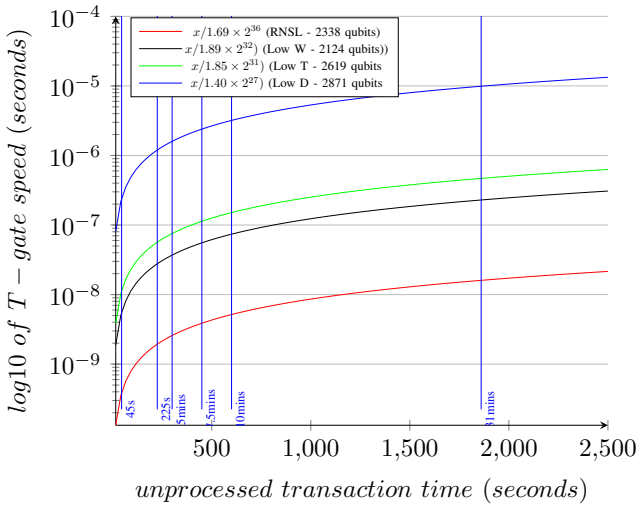
1) The number of qubits required to execute Shor's algorithm.
2) The total time available to conduct the attack (combination of block interval time and number of blocks a transaction can be deferred). Minus the time taken to generate and sign an alternative transaction and distribute this.

This can be applied to any bitcoin and derived cryptocurrency to evaluate both the time window to attack an unprocessed transaction through both a denial of processing valid transactions attack and a block interval extension timing attack that enables a calculation of the quantum computer resources required for a successful attack.

Table 2 illustrates this approach. If a cryptocurrency does not dynamically adjust block interval time, then an assessment of possible block interval stretching was added. In the case assessed as 10 percent as the hash power under control of the adversary will determine the hash reduction possible. In evaluating the possible use of a denial of processing valid transactions attack, an assumption was taken that this would be visible after 2 block intervals and so an assumption was that an adversary could perform this attack for two epochs of the block interval cycle and then this would be both noticeable and the cost of flooding the mempool with valid transactions would significantly increase as the new transaction fee base line increases, making the attack cumulatively more expensive for an adversary.

Figure 5 shows the total number of T-gates required to calculate ECDLP circuit as calculated by Häner et al. [15] and the time available to solve this. This gives a y axis value of the minimum time per Toffoli gate required to meet this window and an effective speed a quantum computer would need to exceed in order to achieve this. Based upon the cryptocurrency and the combined hybrid attacks described in this paper with the assesses unprocessed transaction time, a view of the speed of a Toffoli gate can be formed that will enable tracking of this performance metric over time to assess vulnerability of a specific cryptocurrency.

Figure 5: gate (depth) speed required to meet unprocessed transaction time (Seconds) for ECDLP P256



Through use of multi-signatures, requiring n of m signatures and moving in bitcoins case from a P2PKH to P2SH (Pay to Script Hash) this increases the number of private keys needed to be attacked to find private keys within the unprocessed transaction window.

For example, moving to a one time use multi-signature address with 2 out of 3 signatures required would require two public keys to be broken in the same time window. Currently bitcoin support up to 20 of 20 multisig. These increase the computing power required by a quantum adversary. However, they can be performed in parallel so although a 2 of 3 multi-signature address requires breaking 2 public keys and double the quantum computing power, it will not limit the time in terms of Toffoli gate speed but the available quantum computing power and costs associated with an attack. Multi-signature support is included in the bitcoin core base code and thus supported by bitcoin and derived cryptocurrencies. The level of integration will vary by cryptocurrency. As an example the Zcash cryptocurrency currently only supports multi-signatures through the bitcoin API, meaning that the privacy protecting features of Zcash are not possible to use in conjunction with multi-signatures.

Vitalik Buterin [17] in his article 'Bitcoin Is Not Quantum-Safe, and How We Can Fix It When Needed.' Illustrates this approach. We can upgrade the blockchain ahead of a quantum threat by changing the signature scheme. In this article Buterin proposes a Lamport signature scheme. However, this has the limitation of one time signatures and management of these to remain quantum safe. NIST post quantum signature schemes provide perhaps the best alternative signature schemes today.

VII. The Impact of Noise and Error Rates

A Major challenge with error-prone qubits in a quantum computer is that errors can potentially upset an entire calculation, yet it is not possible to deal with them in the same way as we do in classical computers. In a classical computer you can just make several copies of each bit for backup, and take the consensus value to be the correct one; the chance of a majority all switching in error is tiny. But in quantum computers this is not possible: The entire quantum computation relies on not knowing which state the qubit is in until the calculation is completed, and a fundamental principle of quantum mechanics says that you can't make a copy of an unknown quantum state without changing it.

Circuit	bit modulus	Total Gates	Total Depth d	Qubits w	Error rate ϵ
Low W	256	$1.45 \cdot 2^{35}$	$1.89 \cdot 2^{32}$	2124	$\epsilon \ll \frac{1}{1.89 \cdot 2^{32} \cdot 2124}$ $\epsilon \ll 5.79994429 \cdot 10^{-14}$
Low T	256	$1.80 \cdot 2^{34}$	$1.85 \cdot 2^{31}$	2619	$\epsilon \ll \frac{1}{1.85 \cdot 2^{31} \cdot 2619}$ $\epsilon \ll 9.61087453 \cdot 10^{-14}$
Low D	256	$1.40 \cdot 2^{34}$	$1.40 \cdot 2^{27}$	2871	$\epsilon \ll \frac{1}{1.4 \cdot 2^{27} \cdot 2871}$ $\epsilon \ll 1.85365492 \cdot 10^{-12}$

Table 3: ECDLP Shor's algorithm - Error rate required

Consequently, reducing the error rates in qubits is a major research area. Cross et al. [18] make the argument that performance should be measured not in terms of crude qubit counts but using a quantity they call "quantum volume," which takes into account such things as error rate. derived from such implementation design choices such as coherence, calibration errors, crosstalk, spectator errors, gate fidelity, measurement fidelity, initialization fidelity, qubit connectivity and gate set error rates. This enables one number to represent the power of a particular quantum computer implementation and benchmarking against other quantum computers.

The assessment of Shor's algorithm implementation so far has assumed ideal hardware with no error rates. In order to take account practical error rates, we need to understand what the effect of error rate has upon the algorithm circuit size Leyman et al. [19] describe the following simple formula results in a rule-of-thumb that is very helpful to assess the limits of executing a quantum algorithm on a given quantum computer:

$$d \cdot w \ll \frac{1}{\epsilon}$$

In this formula ϵ is the error rate of the quantum computer. Informally, the error rate subsumes decoherence times of qubits, precision and error frequency of gates etc; a formal and detailed discussion is given by Wilsch et al. [20] . For our purpose, the detailed definition of the error rate is not needed, an informal understanding suffice. As implied by the formula, the depth d or the width w have to be "small". For example, if an algorithm requires 50 qubits (w=50) and it should run on a quantum computer with an error rate of about $10^{-3} (\epsilon \approx 10^{-3})$, then d must be (significantly) less than $20 (d \ll 20)$, i.e. the algorithm has to complete after at most 20 sequential steps - otherwise the result would be much to imprecise.

Table 3 shows results applying this to Shor's algorithm circuit and the range of width and depth as calculated by Häner et al. [15].

VIII. The Current State of Quantum Computing

QUANTUM computing is still in its early stages. There are many practical issues that need to be overcome to make a practical and scalable quantum computer. However, progress is accelerating and many commercial companies are developing technology roadmaps that increase the quantum

computing capacity in an exponential manner. Perhaps the biggest issue today is the error rate and time that a quantum state can be maintained - decoherence time. This is partly because of the technology used - Noisy Intermediate Scale Quantum devices (NISQ). However, new quantum error correcting schemes are currently an area of active research.

In a recently published article - The biggest flipping challenge in quantum computing [21] Chad Rigetti (Founder of Riggetti Computing) is quoted as stating

> "It is really the difference between a $100 million, 10,000-qubit quantum computer being a random noise generator or the most powerful computer in the world".

Table 4 illustrates current industry approaches and characteristics for underpinning technologies for quantum computers.

Clasification	Examples	Qubit Lifetime	Gate fidelity	Gate operation time	Scalability
Superconducting	IBM, Google, Rigetti, Alibaba, Intel, Quantum Circuits	50-100μs	99.4%	10-50ns	Highly Scalable
Ion Trap	IonQ, Alpine Quantum technologies, Honeywell	>1000s	99.9%	3-50μs	TBC
Photonics	PsiQuantum, Xanadu	150μs	98%	1ns	Highly Scalable
Neutral Atoms	Atom computing, PASQAL, QuEra	>1000s	95%	TBC	TBC
Silicon	Intel, Silicon Quantum Computing	1-10s	99%	1-10ns	Expect High Scalability
Topological qubits	Microsoft	Very High	Very High	Unknown	Unknown

Table 4: Classification and characteristics of quantum computers

IX. Conclusion and outlook

THE question of how vulnerable bitcoin or a derived altcoin cryptocurrency is to a quantum computer attack is dependent upon a quantum computer capable of executing Shor's algorithm and executing this within a time budget that is dependent upon the architecture of the cryptocurrency and the time taken to process an unprocessed transaction.

The common issue that all cryptocurrencies have in common is the need to disclose the public key and signature to execute the unlocking script to prove ownership and move funds.

In this paper the consideration has been restricted to Proof of Work consensus algorithms for the reason that they have the largest block interval time versus other consensus mechanisms.

The underpinning digital signature scheme (Elliptic curve) is vulnerable to a quantum computer attack. However, the risk profile will depend not only on the availability of a quantum computer with the requisite capacity (qubits) but also the time budget before a unprocessed transaction is placed in a block which will determine the required speed of execution of the quantum computer for an attack to be successful.

The key discovery from this paper is that not all cryptocurrencies have the same timing attack vulnerabilities, or attractiveness by a quantum attacker. There are several mitigating approaches that a user can adopt, the most critical is to never re-use an address. A user, through use of multi-signatures, can increase the number of qubits required by a quantum attacker and therefore increase the cost of an attack. When a user is submitting a transaction from an address with a large amount of cryptocurrency, by paying a high transaction fee this increases the cost of a denial of processing valid transactions attack. Users can reduce attractiveness to a quantum attacker by holding a maximum value of cryptocurrency in each address that would make a quantum adversaries attack unprofitable. This maximum amount may reduce over time as quantum computers reduce in cost.

A consequence of this is that for some cryptocurrencies migration from ECDSA to quantum safe signatures may be possible to delay until a quantum computer is approaching the performance required to pose a threat. However, this requires a significant change in user behaviour and a cryptocurrency becomes increasingly less useable as each additional protection is added.

Migration to an alternative quantum safe digital signature scheme will require a 'hard fork', a non-compatible new blockchain created.

A more plausible scenario would be that a cryptocurrency is migrated from one cryptocurrency to a new cryptocurrency that is designed to be quantum resistant. This could be achieved relatively easily through an exchange process as a cryptocurrency is becoming more at risk to the ever increasing performance of quantum computers.

This research provides an approach to assess the relative quantum threat posed to a cryptocurrency. This enables the tracking of risk against advances in quantum computers over time, regardless of the underpinning technologies used in a quantum computer.

Future anticipated research; investigation of the impact of replacing ECDSA signature scheme with a National Institute of Standards and Technology (NIST) candidate quantum safe signature scheme. Investigate attacks on alternative consensus mechanism such as proof of stake (PoS) to increase unprocessed transaction times.

References

[1] Nakamoto, S. Bitcoin: a peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf. Accessed 7/10/2020 (2008)

[2] National Institute of Standards and Technology (NIST). Digital Signature Standard (DSS). FIPS pub 186-4. (2013)

[3] Bitcoin.org, https://developer.bitcoin.org/devguide/transactions.html Accessed 10/10/2020 (2020)

[4] International Standards Organisation (ISO). Technical Committee 307. Blockchain and distributed ledger technologies Security risks, threats and vulnerabilities (2020)

[5] Homoliak, Ivan & Venugopalan, Sarad & Hum, Qingze & Reijsbergen, Daniel & Schumi, Richard & Szalachowski, Pawel The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses arXiv:1910.09775 (2019)

[6] International Standards Organisation (ISO). ISO/IEC 15408-1:2009 reviewed 2015 Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model https://www.iso.org/standard/50341.html (2015)

[7] Shor, P. W Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer SIAM Review 41.2303?332 https://doi.org/10.1137/S0036144598347011 (1999)

[8] Aggarwal, D. & Brennen, G & Lee, T & Santha M & Tomamichel, M The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses https://doi.org/10.5195/ledger.2018.127 (2017)

[9] M. Saad, L. Njilla, C. Kamhoua, J. Kim, D. Nyang and A. Mohaisen, Mempool optimization for Defending Against DDoS Attacks in PoW-based Blockchain Systems, IEEE International Conference on

Blockchain and Cryptocurrency (ICBC), Seoul, Korea (South), 2019, pp. 285-292, doi: 10.1109/BLOC.2019.8751476. (2019)
[10] Bitinfocharts https://bitinfocharts.com Accessed 25/10/2020 (2020)
[11] Meshkov, Dmitry et al. Revisiting Difficulty Control for Blockchain Systems. IACR Cryptol. ePrint Arch. 2017 (2017): 731. (2017)
[12] Proos, John & Zalka, Christof. Quantum Information & Computation. QIC 3 (No. 4) (2003) pp.317-344 arXiv:quant-ph/0301141 (2003)
[13] Jordan, Stephen Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves Physical Review A. 95. 10.1103/PhysRevA.95.03230 DOI:10.1103/PhysRevA.95.032305 (2003)
[14] Roetteler M., Naehrig M., Svore K.M., Lauter K Fast quantum computation at arbitrarily low energy Lecture Notes in Computer Science, vol 10625. Springer, Cham (2017)
[15] Häner, Thomas & Jaques, Samuel & Naehrig, Michael & Roetteler, Martin & Soeken, Mathias. Improved Quantum Circuits for Elliptic Curve Discrete Logarithms. arXiv:2001.09580 [quant-ph] (2020)
[16] Dmitri Maslov, Basic circuit compilation techniques for an ion-trap quantum machine New J. Phys. 19 023035 https://doi.org/10.1088/1367-2630/aa5e47 (2017)
[17] Buterin,V. Bitcoin Is Not Quantum-Safe, and How We Can Fix It When Needed. http://bitcoinmagazine.com/articles/ bitcoin-is-not-quantum-safe-and-how-we-can-fix-1375242150/ Accessed 12/11/2020 (2013)
[18] Cross,A, et al. Validating quantum computers using randomized model circuits. Physical Review A doi = 10.1103/PhysRevA.100.032328 (2019)
[19] Leymann, Frank and Barzen, Johanna The bitter truth about gate-based quantum algorithms in the NISQ era Quantum Science and Technology doi=10.1088/2058-9565/abae7d (2020)
[20] Willsch, Dennis & Nocon, Madita & Jin, Fengping & Raedt, Hans & Michielsen, Kristel. Gate error analysis in simulations of quantum computers with transmon qubits. Physical Review A. 96 doi = 10.1103/PhysRevA.96.062302 (2017)
[21] Cho, Adrian The biggest flipping challenge in quantum computing AAAS Science doi: 10.1126/science.abd7332 (2020)