How to hard-fork to save most users' funds in a quantum emergency

Execution Layer Research

vbuterin 1 Mar 2024

Suppose that it is announced tomorrow that quantum computers are available, and bad actors already have access to them and are able to use them to steal users' funds. Preventing such a scenario is the goal of **quantum-resistant cryptography** (eg. Winternitz signatures, STARKs), and once account abstraction is in place, any user can switch to using a quantum-resistant signature scheme on their own schedule. But what if we don't have that much time, and a sudden quantum transition happens long before that?

I argue that actually, **we are *already* well-positioned to make a pretty simple recovery fork to deal with such a situation**. The blockchain would have to hard fork and users would have to download new wallet software, but few users would lose their funds.

The main challenge with quantum computers is as follows. An Ethereum address is defined as `keccak(priv_to_pub(k))[12:]`, where `k` is the private key, and `priv_to_pub` is an elliptic curve multiplication to convert the privkey into a pubkey. With quantum computers, elliptic curve multiplications become invertible (because it's a discrete-log problem), but hashes are still safe. If a user has not made any transactions with their account, then only the address is publicly visible and they are already safe. But if a user has made even one transaction, then the signature of that transaction reveals the public key, which in a post-quantum world allows revealing the private key. And so most users would be vulnerable.

But we can do much better. The key realization is that in practice, **most users' private keys are themselves the result of a bunch of hash calculations**. Many keys are generated using BIP-32 55, which generates each address through a series of hashes starting from a master seed phrase. Many non-BIP-32 methods of key generation work similarly: eg. if a user has a brainwallet, it's generally a series of hashes (or medium-hard KDF) applied to some passphrase.

This implies the natural structure of an EIP to hard-fork the chain to recover from a quantum emergency:

1. Revert all blocks after the first block where it's clear that large-scale theft is happening
2. Traditional EOA-based transactions are disabled
3. A new transaction type is added to allow transactions from smart contract wallets (eg. part of RIP-7560 86), if this is not available already
4. A new transaction type or opcode is added by which you can provide a STARK proof which proves knowledge of (i) a private preimage `x`, (ii) a hash function ID `1 <= i < k` from a list of `k` approved hash functions, and (iii) a public address `A`, such that `keccak(priv_to_pub(hashes[i](x)))[12:] = A`. The STARK also accepts as a public input the hash of a new piece of validation code for that account. If the proof passes, your account's code is switched over to the new validation code, and you will be able to use it as a smart contract wallet from that point forward.

For gas efficiency reasons (after all, STARKs are big), we can allow the STARK to be a batch proof, proving N STARKs of the above type (it has to be a STARK-of-STARKs rather than a direct proof of multiple claims, because each user's `x` needs to be kept private from the aggregator).

The infrastructure to implement a hard fork like this could in principle start to be built tomorrow, making the Ethereum ecosystem maximally ready in case a quantum emergency does actually come to pass.

So you wanna Post-Quantum Ethereum transaction signature 6

Tasklist for post-quantum ETH 5

Post quantum TXs in The Verge 1

22.5k views 104 likes 11 links 20 users 12 min read

irnb Mar 2024

I created this to help visualize the proof statement.

18

myaksetig Mar 2024

This is a great suggestion! A while ago Chaum, myself, and Mario Larangeira created the notion of quantum-secure fallbacks for wallets. First we started with actually hiding a post-quantum key behind the ecdsa key (easier to rollover, but not completely compatible with traditional wallets).

We now have a paper under review where we propose pretty much what you suggest here (which has already been discussed in a different thread) as well as the integration of that preimage into the ecdsa signature nonce to create a fail-stop signature scheme

4

pldd Mar 2024

The more keys have to be replaced, the bigger the future upgrade.

We made quantum resistant smart contract wallets at anchorwallet.ca using Lamport signature.

A new version built on ERC 4337 is coming.

2

DogeProtocol Mar 2024

If quantum computers are already in the hands of a bad actor and are able to crack Ethereum wallets (like using Shor’s algorithm) fast enough, it would be too late, since determining a bad-actor from the actual owner of the account wouldn't be possible.

Do not use stateful post-quantum algorithms; rather use NIST standardized ones in hybrid mode (combiner) with a classical algorithm, like Dilithium + ed25519. Ethereum have to take a significant hit on block sizes though, due to the currently standardized post-quantum dsa scheme's large signature and public key sizes.

1 Reply 2 ♡

---

nvmmonkey Mar 2024

If quantum computers are in bad hand, we need a ML monitor system in the node tree to detect large transactions of unsafe/abnormal human transfer first to trigger Stark fail-safe pre-built emergence fork. Dynamic on chain fuzzing/firewall intercept protocol could be a big leap for prior the enforced fork.

3 ♡

---

tanteikg Mar 2024

My startup, pQCee dot com, has been working on a quantum-safe ECDSA based on the pre-image proof which is similar in design to @vbuterin post. We called it Signature Pre-image proof (SPP) and is patent-pending. It was presented at decompute 2023 (a side event from Token2049)

We started with making ECDSA signatures in PDF documents quantum-safe and our next steps include making ECDSA in Ethereum and Bitcoin quantum-safe based on the same principle. Watch out for our EIP to be available soon.

1 Reply 3 ♡

ShaiW Mar 2024

Hi @vbuterin, thank you for the post.
About a year ago (almost to the day, actually!), my Ph.D advisor Or Sattath and I considered the same problem, and proposed a protocol employing similar ideas – in particular using the BIP-32 derivation process for post-quantum authentication.
A preprint is available on IACR /2023/362, and our work was also presented in the PQCSM2 workshop, slides available on their website (your policy does not allow me to post links).
We flesh out ideas very similar to yours and discuss how they can be composed. Our work also includes a careful analysis of the collision resistance of BIP-32 derivation paths, as well as a security analysis of the resulting signature scheme.
The greatest difference between our approaches, I think, is that we used Picnic signatures rather than STARKs. The advantage of our approach is that gas can be paid from the spent pre-quantum account itself, but of course, the disadvantage is that the signatures cannot be batched.
We propose another approach to deal with signature sizes: a protocol where the signature must only be posted to the blockchain in case of fraud attempts.
We also describe a "quantum canary" mechanism for detecting quantum adversaries (inspired by Justin Drake's cryptographic canaries) and provide some analysis of its game theory.
You might find that our work expands and complements the ideas presented in your post.

5 ♡

MaverickChow Mar 2024

Is it possible to just do an EIP upgrade whereby users have a choice to generate a totally different address/private key that is quantum resistant anytime he wants and do the transition on his own?

In this situation, we would have 2 types of addresses/keys in use; the traditional non-quantum resistant one and the newer quantum resistant one. After the quantum attack, any user who has already transitioned to using the newer keys (before the attack) need not do anything, while users who are still using the traditional one can then transition to the newer keys.

This way, I think, will create the least disruption to the network.

1

---

pldd Mar 2024

We have a Cryptokitty stored in a Lamport wallet since last year, see nft/0x06012c8cf97bead5deae237070f9587f8e7a266d/1850 on etherscan, stored at address 0xE1C67B7Db6eA02125D4a2d4Ec91D382dbF98E3d9
It's impossible to move this NFT without a Lamport signature verified by the network.

So it's already possible with the smart contract wallets on anchorwallet.ca

We are testing a version built on ERC 4337 on Sepolia currently.

1

---

corepool Mar 2024

5 years ago, My team has organized one small summit to talk about this topic. The conclusion is cannot use hard fork to help the real owner take back the coins after the fork. Because there is no way to separate the owner and Q-hacker.
Hope we find the right way now.

---

myaksetig DogeProtocol Mar 2024

Unlike what you think, it is actually possible. That 'impossibility' was already proven incorrect (by us). You can show specific one-way hashing encoding in the way you create the secret key, which the malicious actor can't…

Also, the point of this construction is not to specify what will be the post-quantum signature scheme. But how to rollover a regular ecdsa wallet to *any type* of post-quantum wallet securely

---

domothy Mar 2024

While I understand this is in the hypothetical scenario of a "zero-day" quantum attack, I really don't like the idea of having to rollback a huge number of blocks, especially given that (i) it might not be so obvious where the large-scale theft began and (ii) the hard fork has to be implemented and coordinated, taking more time where new blocks are produced that will ultimately have to be rolled back (effectively a massive liveness failure)

more on (i), picture a single large exchange wallet being drained by a quantum computer. Everyone would naturally assume it was a security failure of some kind on the exchange's end. Or if a smart wallet relying on discrete log assumption gets drained, a smart contract bug/exploit would be the first thing that comes to mind. Or the quantum-enabled attacker avoids high profile targets altogether and slowly steals funds from various large EOAs, and we never even know a quantum attack took place

I would rather see this EIP proactively implemented *today* with the addition that the mechanism stays inactive until a kill switch is linked to previous ideas around “Quantum Canary” is triggered – i.e. a large enough amount of ETH sitting in a contract, secured only by a discrete log problem weaker than the ECDSA used by EOAs, but still infeasible to crack through a classical computer. If/when the ETH gets claimed, then no need for a messy emergency hard fork, since the mechanism outlined in the original post would kick in in the very next block

4 ♡

MaverickChow Mar 2024

How does the solution work with EOA-based keys that were generated pre-BIP, i.e. those generated without 12/24-word seed/passphrase?

1 ♡

Mirror Mar 2024

As far as I remember, the Ethereum Foundation has a dedicated team researching this (I remain optimistic about this), and I do not think that the progress of quantum computing will pose a crisis to Ethereum in the short term. I have not researched the impact of quantum computing on Ethereum, but I have studied about Bitcoin, so perhaps I can help clarify:

Currently, the most advanced superconducting processors have only hundreds of quantum bits (qubits), and the ultra-low temperature working environment limits the processor size and prevents physical manipulation. Room temperature superconductivity (not yet achieved) will address the existing hardware expansion issues in quantum computing.

Bitcoin’s public key is vulnerable and easily attacked within a time window of approximately 10-60 minutes after a transaction is initiated. Breaking encryption within an hour requires approximately 317 million physical qubits.

Citing “The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime” 13 it states:

"We quantify the number of physical qubits needed to break encryption as a function of code loop time and basic physical error rate. Using surface codes, with a code loop time of 1 microsecond, reaction time of 10 microseconds, and physical gate error rate of 10^-3, breaking encryption within an hour requires approximately 317 million physical qubits.

If the basic physical error rate is a more optimistic value of 10^-4, then breaking encryption within an hour would require 33 million physical qubits. This significant demand for physical qubits implies that the Bitcoin network will not be threatened by quantum computing attacks for many years to come (potentially over a decade)."

Do not let “Vitalik says” become an accomplice to fraud, quantum computing cannot harm Bitcoin, let alone harm the actively innovative Ethereum.

2 ♡

---

pa7x1 Mar 2024

I’m glad to see a plan is being formed. I have a few questions.

The proposal above deals with an emergency hard fork to undo an attack.

Q1: How would the transition to a post-quantum Ethereum be changed (if at all) in case we were to perform the change preemptively? That is, if instead of having to rush a change to fix we have time to plan an ideal post-quantum solution.

Q2: How would we morph the hotfix solution proposed above into this final solution (if at all)?

The idea of the above two questions is that it's good to have a plan to hot fix. But hot fixes usually take shortcuts and trade-offs that may not necessarily be ideal. How would the ideal solution look like and how we could migrate the hotfix to it?

Q3: The above solution deals with the use of elliptic curves for public key derivation. But Ethereum uses elliptic curves also for BLS signature aggregation and the KZG commitments. How are these impacted and what plan is there to move them to quantum resistant algorithms if needed?

2 ♡

---

doctor-gonzo Mar 2024

Glad to see people of such caliber thinking / planning for this – I did some basic research on this topic 5 years ago and think there are a few difficulties which are not completely addresses by this proposal:

- **It might not be obvious when the quantum attack started** (if used in a targeted way, or *if the goal is to discredit the security of Bitcoin / Ethereum rather than to steal the funds of any individual accounts*), similar to concerns mentioned by @domothy above
- **The social and coordination aspects of such an emergency hard-fork / pause on transactions might be thornier than they appear**, and may destroy a large part of the financial value stored in BTC and ETH even if the technical aspects of relevant hard-forks are sound. With such a highly technical matter, most users will not have any idea of the cryptographic details and trusted individuals / teams (such as Vitalik) will be crucial for advocating which path to take, and there are sure to be suspicious parties and contentious decisions.

I have previously worked with the Quantum Resistant Ledger 5 project, which has had a post-quantum XMSS-secured mainnet live for 5+ years. They are a member of the Post-Quantum Cryptography Alliance 8 and have funded grants for research by Geometry Labs to push the frontier of post-quantum signature aggregation techniques 4 and turn their results into an open-source implementation to be evaluated by NIST.

QRL is working to incorporate smart-contracts / EVM compatibility in their next update, and **if any talented cryptographers and/or blockchain developers want to contribute to a post-quantum life-raft, collaboration with QRL and Geometry Labs might be among the most immediate / direct / open-source ways.**

I love Ethereum and am not trying to shill QRL too much, but I am concerned about a sudden quantum advance destroying the credibility of blockchain networks (and/or the financial value stored in them). It doesn't seem like the Bitcoin ecosystem takes the quantum risk seriously at this point, and with BTC being introduced into the mainstream financial system it strikes me as somewhat of a time bomb.

It seems possible that public knowledge of the SOTA on quantum computers will be lacking right up until there is some plausible "y2q" event (such as Satoshi's BTC and other pay-to-public-key mining reward coins being moved), due to the national security implications of such technology and the incentive for relevant governments to push for secrecy around advances (even though it seems the SOTA is in the private sector).

2 ♡

pldd Mar 2024

Here we go @vbuterin : ERC-4337 + Lamport on Ethereum Mainnet as a browser plugin wallet. We are doing a few more tests and we will start distributing more widely.

2

13 days later

pldd Mar 2024

The plugin wallet Anchor Vault is now available in the Chrome Web store. The wallet is built on ERC 4337 and implements quantum resistance by using Lamport signatures.

This way it’s possible to protect Ethereum assets now @vbuterin .

12 days later

matthiasgeihs Apr 2024

How would all of this be affected if EC-based Verkle tries are included in the picture? @vbuterin

1 Reply

1 / 39