WHITEPAPER / V0.3.3

quantus

quantum secure, encrypted money

PUBLISHED	VERSION	CLASSIFICATION
march 21, 2026	0.3.3	public

LEGAL DISCLAIMER

This whitepaper is provided for informational purposes only and does not constitute an offer to sell, a solicitation of an offer to buy, or a recommendation for any security, investment, or financial product. Readers should conduct their own due diligence and consult with qualified professionals before making any investment decisions. Quantus Network makes no representations or warranties regarding the accuracy or completeness of the information herein.

CONTENTS

01

introduction

the quantum threat

Traditional blockchains face an existential threat from cryptographically relevant quantum computers (CRQCs). The cryptographic foundations of blockchains rely on the hardness of the discrete logarithm problem (DLP), and quantum algorithms, notably Shor's, can solve the DLP exponentially faster than classical computers. This vulnerability could enable quantum- adversaries to derive private keys from public keys, which would allow them to forge transactions and decrypt sensitive financial data.

> without proactive quantum-resistant upgrades, the trillion-dollar crypto economy risks sudden devaluation from such attacks.

unique value proposition

Named after the Latin word for “how much”, Quantus Network enables scalable, quantum-secure, private money. Quantus is not a general purpose smart contract platform. Like a restaurant with a few highly perfected menu items, Quantus delivers:

— Post-quantum signatures for all transactions

— Post-Quantum signatures and encryption (ML-DSA and ML-KEM) to secure peer connections

- Post-Quantum zero-knowledge-proofs to scale
- High Security Accounts to deter theft and enable recovery from mistakes
- Human-Readable check-phrases for easy address verification

The decision to focus on scalable, quantum-secure, private money stems from the threat CRQCs present to the industry and Bitcoin’s inability to address these challenges.

02 the quantum threat to blockchain

quantum computing basics

Quantum computers leverage principles like superposition and entanglement to perform computations that are intractable for classical machines. Unlike classical bits, which are either 0 or 1, quantum bits (qubits) can exist in multiple states simultaneously, enabling exponential parallelism for certain problems. This capability poses existential risks to the cryptographic systems underpinning blockchain finance, as algorithms developed for quantum hardware undermine the security assumptions of most public-key cryptography.

Shor's algorithm, introduced in 1994 by Peter Shor, provides a polynomial-time method for factoring large integers and solving the discrete logarithm problem on a quantum computer. It exploits Quantum Fourier Transforms (QFT) to find the period of a function, allowing efficient reversal of the trapdoor functions that underlie schemes like RSA or elliptic curve cryptography (ECC). For blockchain finance, this means an attacker with a sufficiently powerful quantum computer (estimated at ~2,000 logical qubits [6] [7] [8] [9]) could derive private keys from public keys in polynomial time $O(n^3)$ — an extreme speed-up rendering vulnerable systems obsolete overnight. [1]

Grover’s algorithm, proposed by Lov Grover in 1996, offers a quadratic speedup for unstructured search problems, reducing search time from O(n) to O(√n) operations. While not as devastating as Shor’s for asymmetric cryptography, Grover’s impacts symmetric primitives like hash functions and AES encryption, effectively halving the security level (e.g., a 256-bit key behaves like 128 bits against quantum attacks). This attack is mitigated by simply doubling the security bits, rather than changing the cryptographic scheme. Additionally, Grover’s quadratic speedup is impractical due to its high qubit and gate requirements, requiring billions of sequential operations with limited parallelization, making it infeasible for real-world reversals even on future hardware. [2]

four threat categories

01 - forging digital signatures

Shor’s algorithm directly threatens ECC-based signatures used in most blockchains (e.g., Bitcoin’s secp256k1 curve), allowing adversaries to impersonate users and authorize fraudulent transactions. Such a capability would represent a critical failure of the most basic feature of a blockchain.

02 - forging false proofs in zero-knowledge systems

Many zero-knowledge proofs, such as those in zk-SNARKs for privacy-focused finance, rely on discrete logarithm hardness via elliptic-curve pairings for commitments. Shor’s could enable the creation of invalid proofs that appear valid, which could allow an attacker to mint new coins or falsify the state of Layer-2s (L2s).

03 - decrypting secret information

Quantum attacks could expose encrypted data protected by vulnerable public-key schemes in privacy protocols such as Zcash or Monero. It could also decrypt p2p communications in financial protocols, revealing sensitive wealth details and enabling targeted theft.

04 - reversing hash functions

Grover's algorithm could accelerate preimage attacks on hashes like SHA-256, used for proof-of-work and address generation, but this is the least concerning threat. Many post-quantum cryptographic schemes incorporate hash-based constructions as hashes are considered secure-enough with a large enough digest.

scaling challenges in post-quantum cryptography

While post-quantum cryptography (PQC) offers essential protections against quantum threats, it introduces significant scaling hurdles due to the inherent design of these algorithms. Unlike elliptic curve schemes, which rely on compact mathematical structures, PQC primitives require larger parameters to maintain security against both classical and quantum adversaries. This results in substantially bigger public keys, private keys, and signatures, often by orders of magnitude. The following table illustrates typical sizes for ML-DSA at a 128-bit post-quantum security level compared to classical counterparts like 256-bit ECDSA: [10]

ALGORITHM	PUBLIC KEY	PRIVATE KEY	SIGNATURE
ML-DSA-87 (Dilithium)	2,592 bytes	4,896 bytes	4,627 bytes
ECDSA (256-bit)	32 bytes	32 bytes	65 bytes

Sizes at 128-bit post-quantum security level. Source: Open Quantum Safe Project [10]

As shown, ML-DSA signatures can be over 70 times larger than ECDSA equivalents, and public keys more than 80 times larger. Other PQC families exacerbate this: hash-based schemes like SPHINCS+ may produce signatures up to 41 KB, while even size-optimized lattice variants like FALCON still exceed classical sizes by a significant multiple.

In blockchain contexts, these inflated sizes compound into systemic scaling issues. Larger signatures bloat individual transactions, reducing transactions per second (TPS) as blocks fill faster and require more time for validation. This also strains peer-to-peer (P2P) communication, increasing bandwidth demands and propagation delays, which can heighten the risk of network forks or orphaned blocks in consensus mechanisms like proof-of-work. Storage requirements are also affected, leading to higher node operating costs and barriers for participation, especially for resource-constrained users or validators.

> **NOTE**
>
> These scaling challenges will have to be addressed by all blockchains in the future. Bitcoin, for example, will have much less than 1 TPS if the max block size is not increased.

03 the migration crisis

the coordination problem

Bitcoin's conservative culture resists protocol changes. Any PQC upgrade would require consensus on contentious issues such as migration timelines, potential coin seizure, and block size increases. Even if the community agreed, every individual user would need to migrate their coins to new quantum-secure addresses. Migration requires action from every crypto holder, many of whom have lost access to their wallets or remain unaware of the threat.

These issues exist for every public blockchain, but are uniquely challenging to Bitcoin due to its lack of clear leadership and philosophy of technical ossification.

the lost coin problem

An estimated $250 billion to $500 billion worth of Bitcoin is permanently inaccessible due to lost keys, deceased holders, or forgotten wallets. [3] These coins cannot be migrated and serve as a public bounty for creating a cryptographically relevant quantum computer (CRQC). Quantum attackers will derive private keys from exposed unmigrated public keys and likely dump billions of dollars of BTC onto the market.

> the only technical solution requires a hard deadline that freezes unmigrated coins — a political impossibility.

Without such a deadline, the outcome will be that unmigrated coins are stolen and sold, crashing the market and destroying confidence in the network.

the migration timeline problem

Post-quantum signatures are 20x–80x larger than current Bitcoin signatures. Without fundamental architectural changes, Bitcoin's throughput will collapse to a fraction of its already limited capacity.

Assuming Bitcoin solves the political and technical challenges, the migration itself would take months or years. Every holder must submit at least one transaction to move funds to a quantum-secure address. Many will send test transactions first. With bloated PQC signatures choking throughput, the network faces a backlog lasting months or years while quantum-vulnerable coins remain exposed.

> **QUANTUS'S ANSWER**
>
> These compounding challenges make retrofitting quantum security onto existing chains extraordinarily difficult. Quantus Network sidesteps this by building quantum security into the chain from day one.

04

quantus network architecture

foundation

Quantus Network is built on Substrate, a blockchain SDK developed by Parity Technologies, the team behind Ethereum and Polkadot. Substrate is highly modular, enabling easy replacement of components so we can focus on what makes Quantus unique.

Quantus upgrades Substrate by:

- Adding support for post-quantum signature schemes
- Upgrading the p2p networking security to be post-quantum
- Adding user-controlled transaction reversibility
- Making the database zk-friendly by aligning all data types to field-element boundaries

post-quantum cryptographic primitives

Quantus Network employs NIST-standardized PQC to ensure the security of transactions and network communications against quantum threats. At the core of transaction integrity is **ML-DSA** (Module-Lattice-based Digital Signature Algorithm, formerly known as CRYSTALS-Dilithium), a lattice-based signature scheme selected for its balance of security, efficiency, and ease of implementation.

ML-DSA leverages the hardness of problems like Learning With Errors (LWE) and Short Integer Solution (SIS) over module lattices, providing robust resistance to both classical and quantum attacks, including those from Shor's algorithm. [4]

For transaction signatures, Quantus integrates **ML-DSA-87**, the parameter set offering the highest security level (NIST Security Level 5, equivalent to 256-bit classical and 128-bit quantum security) to protect against potential cryptanalytic breakthroughs in lattice problems. This choice prioritizes caution, as lattice cryptography is relatively new and less battle-tested than classical schemes. The larger parameters mitigate risks from potential advances in lattice cryptanalysis, which would still leave smaller key sizes as softer targets.

alternatives considered

ML-DSA was selected over alternatives like FN-DSA (Falcon) due to FN-DSA's greater implementation complexity (e.g., requiring floating-point operations, which are blockchain-unfriendly), lack of deterministic key generation in its specification, and its non-finalized status at the time of development.

Hash-based options like SLH-DSA were not chosen because of their even larger signature sizes (exceeding 17 KB). Crypto-agility (being able to swap in different signature schemes) is built into Substrate, so it is relatively easy to add these alternatives at a future date, should circumstances demand.

While ML-DSA-87 results in larger keys and signatures, these are manageable in Quantus's early-stage network, where storage is not yet a bottleneck, and optimizations like wormhole addresses via zero-knowledge proofs will address scaling.

For technical details about the implementation see QIP-0006.

libp2p - quantum-secure networking

Quantus Network secures peer-to-peer (P2P) node communications using a combination of ML-DSA for authentication and **ML-KEM** (Module-Lattice-based Key Encapsulation Mechanism, formerly CRYSTALS-Kyber) for encryption. This integration extends PQC to the libp2p networking stack, modifying core components for quantum resistance: using ML-DSA-87 signatures for peer identity and ML-KEM-768 for transport security (extending the Noise handshake with an additional KEM message for quantum-resistant shared secrets). [5]

The P2P layer is often neglected in quantum-security analysis. Authentication of peers is important, but the worst an attacker could do at the peer level is impersonate a node and send invalid messages, which could result in denial-of-service. This attack is already mitigated by the fact that nodes are generally untrusted in the blockchain model and nodes can easily switch their keys if the attack is detected. Likewise, decrypting P2P communications yields limited attacker benefits (e.g., tracking transaction paths, mitigated by proxies or Tor), and most data becomes public on- chain anyway.

Nevertheless, quantum-securing the P2P layer protects against eavesdropping, man-in-the-middle attacks, and quantum decryption, ensuring that node gossip, block propagation, and other network interactions remain confidential and tamper-proof for the foreseeable future.

For technical details about the implementation see QIP-0004.

scaling pqc - wormhole addresses

To address the scaling challenges inherent in post-quantum cryptography, Quantus Network introduces an innovative aggregated post-quantum signature scheme called **“Wormhole Addresses”**. This system leverages zero-knowledge proofs (ZKPs) generated via the Plonky2 proving system (basically STARKs) to move balance verification off-chain, allowing the chain to verify a single compact proof without processing individual signatures. Wormhole Addresses enable the verification of a large number of transactions with one proof, with the public inputs (e.g., nullifiers, storage root, exit addresses, and amounts) becoming the primary limiting factor. This reduces the amortized per-transaction storage demands to approximately 256 additional bytes per transaction, much smaller than any known PQC signature scheme.

The quantum security of the scheme derives from the use of the secure hash function **Poseidon2** for commitments via FRI (Fast Reed-Solomon Interactive Oracle Proofs), instead of the quantum-vulnerable elliptic-curve pairings commonly used in SNARKs.

Additionally the authentication secrets are hidden behind Poseidon2. Since secure hash functions are only quadratically weakened by Grover’s algorithm, not broken, hash preimage proofs can serve as lightweight post-quantum signatures in ZK contexts, similar to hash-based schemes like SPHINCS+.

client / prover flow

Users generate a provably unspendable address by double-hashing a salt concatenated with a secret:

```
H(H(salt|secret))
```

This construction prevents false positives (e.g., mistaking a single-hash public key for an unspendable address) because in Substrate (and generally) blockchain addresses are the single hash of a public key, which is derived from the private key via some algebraic operation, not via a secure hash. The security of the construction therefore reduces to finding the preimage-of-a-preimage of a secure hash. Tokens sent to this address are effectively burned. They cannot be spent because no private key exists for the address that received them. These coins can therefore be re-minted without inflating supply.

For each transfer, a TransferProof storage object is created, containing details like a unique global transfer count. The user's wallet generates a Merkle-Patricia-Trie (MPT) storage proof from a recent block header's storage root to the leaf for this TransferProof. A nullifier is computed to prevent double-spends:

```
H(H(salt | secret | global_transfer_count))
```

aggregator flow

Any party (client, miner, or third-party) can aggregate multiple proofs using Plonky2's recursion, forming a tree of proofs where each parent proof is a verification of the child proofs, with the child proofs' public inputs aggregated:

- Nullifiers pass unchanged
- Exit addresses are deduplicated
- Block hashes are proven to be linked and then all but the most recent is dropped
- Amounts for duplicate exit addresses are summed

chain / verifier flow

The network verifies the aggregated proof by checking: block hash is on chain and recent, nullifier uniqueness (to prevent double-spends), and proof validity. The ZK circuit enforces storage proof correctness, nullifier computation accuracy, address unspendability, balance match between inputs and outputs, and block header linkage.

why plonky2

- Already audited
- Post-quantum
- No trusted setup
- Efficient proving/verification
- Seamless proof aggregation
- Rust-native implementation
- Compatible with Substrate’s no-std environment

> **PERFORMANCE**
>
> Recursive proofs complete in 170 milliseconds with compact sizes (100 KB per aggregated proof). In an optimal case with 5 MB blocks and all transactions going to the same output, Wormhole Addresses could pack ~153,000 transactions into a single block (4.9 MB / 32 bytes per nullifier) — a 223x improvement over ~685 raw ML-DSA transactions (5 MB / 7.3 KB each).

security notes

Potential risks include inflation bugs from faulty circuit/verification implementations, although this would be economically detectable if re-minted coins exceed balances of zero-send addresses. Users can optionally prove an address is a wormhole by publishing the first hash without revealing the secret. Verification transactions are unsigned, so denial-of-service via failed transactions needs to be mitigated non-financially. Token supply calculations are maintained, as re-mints appear as new coins but maintain maximum supply guarantees via burns.

For more technical details about the implementation see QIP-0005.

consensus mechanism

Quantus Network uses a Proof-of-Work (PoW) consensus algorithm that preserves the desirable properties of Bitcoin's consensus algorithm while improving compatibility with ZK-proof systems by switching out SHA-256 with **Poseidon2**.

Importantly, this change is not being made for quantum security. Cryptographic hash functions like SHA-256 are weakened but not destroyed by quantum algorithms, notably Grover's. Some post-quantum signature schemes use secure hashes as a building block for this reason.

Poseidon2 is a refinement of the Poseidon hash function. Creating SNARKs or STARKs for computations involving traditional hash functions like SHA-256 often requires nearly 100x the number of gates compared to using Poseidon, which relies entirely on algebraic functions over field elements, instead of bit-level operations.

We use the **Goldilocks field** for both Poseidon2 and Plonky2. The Goldilocks field's order fits in an unsigned 64-bit integer, which increases efficiency without compromising soundness.

05 wealth preservation

There are many risks in managing cryptocurrency keys. Most of them are avoidable.

reversible transactions

Quantus Network offers user-configurable reversible transactions. Senders set a time window during which they can cancel outgoing transfers. This deters theft and corrects errors without sacrificing finality. The system uses a modified Substrate “scheduler pallet” with timestamps. Wallets display countdowns for both sender (with a cancel button) and recipient.

Reversible transactions enable novel security protocols while maintaining decentralization through on-chain enforcement.

For more technical details see QIP-0009.

check-phrases

Quantus Network introduces “check-phrases,” a cryptographically-secure human-readable checksum for blockchain addresses. The address is hashed to generate a short sequence of memorable words from the BIP-39 mnemonic list. Check-phrases protect against typos, tampering, and address poisoning attacks. A 50,000 iteration key derivation function makes rainbow table attacks expensive. For large transactions, users should still verify every character of the address.

For more technical detail please see QIP-0008.

high-security accounts

Any account can be upgraded to a “high-security account” with mandatory reversal periods on all outgoing transfers. A designated **guardian** (hardware wallet, multisig, or trusted third party) can cancel suspicious transactions during the reversal period, sending funds to the guardian instead of the sender or receiver. This opt-in feature is permanent once activated, preventing thieves from disabling it.

Guardians can be chained: a high-security account’s guardian can itself be a high-security account with its own guardian. This creates composable hierarchies where each guardian has superior permissions to the account it protects. The design gives users time to detect and respond to unauthorized activity without compromising finality for legitimate transfers.

For more technical details see QIP-0011.

key recovery

Many crypto-fortunes have gone to the grave with their owners. Quantus Network offers a simple way to specify a recovery address that can pull your funds at any time, subject to a fixed delay. During this time, the owner can deny recovery if they have access to the key. This feature enables survivorship: users have an on-chain will without the need for courts or estates.

hd-lattice

Hierarchical Deterministic (HD) wallets are the industry standard for blockchains, allowing users to back up one seed phrase for all keys, improving security and convenience over manual backups per action. Adapting this to lattice schemes like Dilithium involves two challenges:

- HMAC-SHA512 outputs can't directly form lattice private keys, which are polynomials sampled from a ring with certain properties.
- Non-hardened key derivation relies on elliptic curve addition, absent in lattices (public keys aren't closed under any algebraic operation).

Quantus Network addresses the first issue by using the output of the HMAC as entropy to deterministically construct the private key, not as the private key itself. The second issue is less critical and remains an open research question whether lattice cryptography can be adapted to address it.

For more technical details see QIP-0002.

06

tokenomics and governance

Quantus Network exists in a changing environment, and we cannot assume that we will get everything right on the first try. For this reason, we choose a simple starting point and allow the governance system to make changes as new information is acquired. This design makes the blockchain a living entity that can adapt to its environment at will. In particular, the Substrate governance process allows deep changes to the chain with minimal coordination among the various node-runners.

block rewards

Quantus Network employs a straightforward tokenomics model imitating that of Bitcoin. There is a maximum supply of **21,000,000 coins** and a simple heuristic determines the reward each block:

```
block_reward = (max_supply - current_supply) / co
```

This heuristic forms a smooth exponentially decaying curve as the block_reward contributes to the current_supply which reduces the block_reward computed at the next block. Any burns from fees or otherwise reduce current_supply and essentially become part of the budget for block rewards. The constant is chosen so that, in the absence of any burns, 99% of the coins will be emitted in about 30 years.

investor allocation

Quantus Network was built with the help of investors who took great risk in funding it. Private investors are subject to a 4 year vesting schedule, like the team. Public sale investors will be fully liquid on day one. The funds raised in the public sale will be matched with tokens and used for liquidity (DEX, CEX, and market makers). These investor allocations as well as the liquidity will be the only “pre-mine”. All other tokens will have to be mined into existence.

In the event that less than the maximum 10% is sold during the public sale, there will be a corresponding reduction in the liquidity tokens and the remainder will be emitted to miners via block rewards.

company allocation

To compensate the team for taking the risk to build new technology with no promise of success, a portion of the block reward is sent to the company for approximately four years. This gives a de facto vesting schedule of about **15% of the total supply** to the company.

After that point, the company’s portion of block rewards can be turned off, adjusted or redirected according to a token holder vote.

transaction fees

TRANSACTION TYPE	FEE STRUCTURE	DESTINATION
Standard	Fixed fee	Miners
Reversed (high-security)	1% volume-based	Burned
ZK aggregated	0.1% volume-based	50% miner / 50% burned

forkless upgrades

Quantus Network supports “forkless” upgrades through Substrate’s runtime upgrades, allowing the blockchain’s core logic (the “runtime”) to evolve without hard forks that could disrupt the network or split the community. This is achieved via on-chain governance referenda, where approved proposals trigger a runtime swap — essentially replacing the existing WASM code blob with a new one in a single block, ensuring continuity of state and operations. This upgrade path minimizes downtime and risks, empowering the community to iteratively refine the protocol as real-world usage reveals potential improvements.

As the community gains confidence in the system over time, the power to change the runtime will be significantly reduced to limit the attack surface, should a malicious actor obtain control of the upgrade process.

governance system

Quantus Network inherits its governance framework from Polkadot’s OpenGov system via Substrate. Token holders participate via **conviction voting**, where they agree to lock their assets for varying periods to amplify their vote’s weight. This amplification can range from 1x (no lock) to 6x (maximum lockup). This design encourages long-term alignment by tying influence to commitment.

Proposals are categorized into multiple voting tracks called “origins”. Each origin has tailored parameters like approval thresholds (e.g., supermajority for high-impact changes), minimum deposits to deter spam, preparation/enactment periods, and decision timelines to prevent gridlock. This multi-track design allows parallel processing of diverse referenda, from routine treasury spends to critical runtime upgrades.

The **Technical Collective** is a curated group of technical experts serving as a specialized body to propose, review, or whitelist urgent technical matters, expediting them through a dedicated track while maintaining community oversight.

Quantus adopts this system without modifications but starts with a minimalistic setup to avoid complexity in its early stages. Initially, only the Technical Collective track is active, which will be used for binding, high-privilege decisions like protocol upgrades or parameter tweaks.

Later, Quantus can add a non-binding community vote track for gauging sentiment on non-enforceable topics, such as feature suggestions or ecosystem polls. This system will become binding when the company turns the network over to the DAO. This phased approach allows the network to evolve organically via future governance votes without burdening users with unnecessary complexity at the beginning.

07

roadmap

The current roadmap through 2026, subject to change.

heisenberg inception December 2024	Funding Secured, Substrate Chosen.
resonance alpha July 2025	Public Testnet, Dilithium Signatures, Reversible Transactions.
schrödinger beta October 2025	Features Complete, Ready for Audit.
dirac beta November 2025	PoW changed to Poseidon2, Audits Addressed.
planck beta January 2026	High Security Accounts, Multisigs, Hardware Wallet, ZK integration.
bell mainnet Q2 2026	Mainnet Launch.
fermi upgrade Q4 2026	ZK proof aggregation infrastructure.

08 risks

Building Quantus Network comes with inherent risks.

implementation issues

Flaws in software logic can cause serious failures in even the best designed systems.

nist algorithm selection issues

Potential flaws or backdoors in selected post-quantum standards (e.g., ML-DSA, ML-KEM) that could emerge post-standardization. In the worst case, such flaws would allow an attacker to forge signatures by deriving a private key from the public, representing a catastrophic failure mode of the chain. If such flaws were made public, Quantus Network could be upgraded to a new algorithm, but if such flaws are exploited sparingly they may never be discovered.

quantum computing timelines

Quantum breakthroughs might arrive much later than anticipated, delaying the need for PQC; conversely, secretive development (e.g. by governments) could lead to sudden threats if the blockchain community fails to update swiftly.

other considerations

General adoption barriers, regulatory uncertainties in finance/blockchain, and the inherent volatility of crypto ecosystems.

09 references & further reading

[1] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5), 1484–1509. https://doi.org/10.1137/S0097539795293172

[2] Grover, L.K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the Twenty-Eight Annual ACM Symposium on Theory of Computing*, 212-219. https://doi.org/10.1145/237814.237866

[3] Chainalysis. (2024). *The Chainalysis 2024 Crypto Crime Report*. https://www.chainalysis.com/blog/2024-crypto-crime-report-introduction/

[4] National Institute of Standards and Technology. (2024). *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML- DSA)*. U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf

[5] National Institute of Standards and Technology. (2024). *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)*. U.S. Department of

Commerce. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf

[6] Häner, T., Jaques, S., Naehrig, M., Roetteler, M., & Soeken, M. (2020). Improved quantum circuits for elliptic curve discrete logarithms. *arXiv:2002.12480*. https://arxiv.org/abs/2002.12480

[7] idney, C., & Ekerå, M. (2021). *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits*. *arXiv:1905.09749*. https://arxiv.org/abs/1905.09749

[8] Aggarwal, D., et al. (2021). Assessment of Quantum Threat To Bitcoin and Derived Cryptocurrencies. *ePrint IACR*. https://eprint.iacr.org/2021/967.pdf

[9] Roetteler, M., Naehrig, M., Svore, K. M., & Lauter, K. (2017). Quantum resource estimates for computing elliptic curve discrete logarithms. *arXiv:1706.06752*. https://arxiv.org/abs/1706.06752

[10] Open Quantum Safe Project. (n.d.). ML-DSA | Open Quantum Safe. Retrieved January 29, 2026, from https://openquantumsafe.org/liboqs/algorithms/sig/ml-dsa.html