# Post Quantum Registry (PQR):
## A Preemptive Ownership Validation Mechanism for Bitcoin Under Quantum Threats

**Abstract:** As quantum computing progresses, established cryptographic assumptions underlying Bitcoin's security—particularly ECDSA-based signatures—face potential compromise. We present the Post Quantum Registry (PQR), a protocol extension enabling current Bitcoin address owners to proactively bind their holdings to post-quantum (PQ) cryptographic identities, preserving demonstrable control in the event of a future quantum break. By embedding a hash of a PQ public key into the Bitcoin blockchain via a provably controlled transaction, PQR establishes a time-stamped, on-chain record. This record can serve as authoritative evidence of pre-quantum ownership, facilitating recovery in a potential hard-fork or reorg following a cryptographic crisis.

## 1. Introduction

Bitcoin's security model critically relies on the hardness of the Elliptic Curve Digital Signature Algorithm (ECDSA). A sufficiently advanced quantum computer could solve the discrete logarithm problem underlying ECDSA within a feasible timeframe, enabling attackers to derive private keys from exposed public keys. Such an event ("quantum break") would allow malicious reappropriation of assets and destabilize the network's trust model.

While post-quantum cryptography schemes such as Dilithium have emerged as candidates resistant to known quantum algorithms, retroactively asserting ownership of Bitcoin holdings after a quantum break is non-trivial. Users who fail to transition to PQ addresses before the break risk losing verifiable linkage to their prior balances. The Post Quantum Registry aims to mitigate this risk by creating a forward-compatible binding between current ECDSA-controlled addresses and PQ identities prior to any known quantum compromise.

## 2. Background

• **Quantum Vulnerability of ECDSA:**

Shor's algorithm can factor integers and solve discrete logarithms in polynomial time, threatening ECDSA-based cryptocurrencies.

• **Post-Quantum Signatures (Dilithium):**

Dilithium, a lattice-based signature scheme, is considered secure against classical and quantum attacks. Despite larger key sizes, it enables quantum-resistant signatures.

• **On-Chain Identity Binding via OP_RETURN:**

Bitcoin allows embedding of arbitrary data in the blockchain using OP_RETURN outputs. Storing the hash of a PQ public key is efficient and tamper-evident.

## 3. System Overview

### 3.1 Prerequisites

• A Bitcoin address currently controlled by a user's ECDSA keypair.
• A Dilithium keypair (sk_D, pk_D), generated securely offline.
• A hash function (SHA-256) to compress pk_D into a fixed-length fingerprint.

### 3.2 Registration Procedure

1. **Key Generation:** Users generate a Dilithium keypair (sk_D, pk_D).
2. **Fingerprinting:** Compute H = SHA256(pk_D).
3. **On-Chain Commitment:**

Users create a Bitcoin transaction, funded by their ECDSA-based address, with an OP_RETURN output containing a prefix (PQR_) followed by H.

4. **Finality and Indexing:**

After confirmation, the Bitcoin blockchain and indexing services record the binding between the user's Bitcoin address and the Dilithium-based identity.

### 3.3 Post-Quantum Recovery Scenario

If a quantum adversary compromises ECDSA, the original private keys are no longer secure. However, the pre-established link (address → PQ hash) still exists. Users can present their (pk_D, sk_D) that matches the on-chain hash H, proving legitimate ownership before the quantum event. In a community-led fork or recovery plan, these PQ proofs enable rightful asset reclamation.

## 4. Security Considerations

• **Cryptographic Soundness:**
PQR's efficacy relies on the chosen PQ scheme and the hash function retaining security.
• **Social Consensus:**
PQR requires communal agreement to honor registered PQ identities in a post-quantum recovery scenario.
• **Adoption and Key Management:**
Users must proactively register and securely store their Dilithium keys. Without wide participation, PQR's utility as a universal safety net diminishes.

## 5. Limitations and Future Work

• **Preemptive Nature:** Only addresses pre-registered before a quantum break gain this security advantage.
• **Scalability:** While individual hashes are small, large-scale PQR adoption adds data to the blockchain.
• **Algorithmic Robustness:** As post-quantum standards evolve, new PQ schemes or stronger hash functions may be required.

## 6. Conclusion

PQR offers a forward-looking mechanism to preserve provable ownership of Bitcoin addresses in anticipation of future quantum attacks. By linking current ECDSA-based identities to PQ keys on-chain, PQR sets the stage for a more robust post-quantum recovery process. Its success will depend on the ongoing confidence in PQ algorithms, widespread adoption, and social consensus to enact recovery measures if a quantum crisis arises.

## 7. References


• [1] Bernstein, D. J., et al. "Post-quantum cryptography." Springer, 2009.
• [2] Lyubashevsky, V., et al. "Lattice Signatures without Trapdoors." EUROCRYPT (2012).
• [3] NIST Post-Quantum Cryptography Standardization. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography
• [4] Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System." (2008).

# Step-by-Step User Instructions

## 1. Generate a Dilithium Keypair:

• Use a secure PQ wallet or key generation tool endorsed by trusted developers.
• Click “Generate PQ Keypair” and save both the public key (pk_D) and private key (sk_D) in a safe, offline location (e.g., a hardware device or encrypted USB drive).

## 2. Create a Fingerprint of Your Dilithium Public Key:

• In your PQ wallet tool or where elsewhere available, select “Create Hash of PQ Public Key.”
• The tool will compute H = SHA256(pk_D).
• Copy this hash H (it should be a short string of letters and numbers).

## 3. Prepare a Bitcoin Transaction From Your Regular Address:

• Open your regular Bitcoin wallet where you control an ECDSA-based address.
• Start a new transaction.
• Add an OP_RETURN output. Your wallet’s interface should allow adding a “Data” or “Message” field.
• In that field, type PQR_ and then paste the hash H.
Example:
**PQR_*HHHHHHHHHHH...HHHH***
(Replace HHHH...HHHH with your actual hash.)

## 4. Fund and Send the Transaction:

• Confirm that you are spending from the exact Bitcoin address you want to secure under PQR. This proves you own that address at this moment.
• Set an appropriate transaction fee and click “Send.”
• Wait for the transaction to confirm on the Bitcoin network. Confirmation may take some time depending on network conditions.

## 5. Verify Your Registration:

• After the transaction is confirmed, go to a blockchain explorer or a PQR-indexing service that supports the standard.
• Search for your Bitcoin address. You should see a record showing that it is linked to your PQR hash H.

## 6. **Securely Store Your Dilithium Keys:**

• Keep your (pk_D, sk_D) pair safe. If a quantum break happens in the future, you will need sk_D to prove your rightful claim to the Bitcoin address.
• Regularly back up and protect these keys. Consider offline storage and multiple backups in different secure locations.

## 7. **In Case of a Quantum Break:**

• If a recognized quantum event threatens ECDSA-based ownership, the Bitcoin community may initiate a recovery process.
• Present your (pk_D, sk_D) to verify your hash H matches the on-chain record. This will help you reclaim your Bitcoin.

By following these steps, even a newcomer can set up their Bitcoin address with a post-quantum backup. While there is no guarantee of the community's future recovery actions, having this record significantly improves the likelihood of maintaining rightful ownership in a post-quantum landscape.