Post-Quantum Cryptography for Finances

1. Phase 1: Research & Planning

Timeline: Weeks 1–2

Objectives

1. **Literature & Standards Review**
    - Familiarize yourself with **NIST PQC finalist** algorithms (Kyber, Dilithium, SPHINCS+, etc.) as well as the latest research on quantum-resistant cryptosystems.
    - Analyze regulatory and industry requirements (e.g., **PCI DSS**, banking security standards), taking into account the transition to PQC.
    - **KPI:**
        - A review of **10+ relevant** articles/reports/standards, plus an internal summary document with key findings.
2. **Project Scope & Architecture Definition**
    - Determine the goals: **where** exactly to implement PQC in the financial product (TLS/SSL for transactions, database protection, digital signatures for documents, etc.).
    - Select initial tools/libraries (e.g., liboqs, HSM vendor libraries, other open-source solutions).
    - **KPI:**
        - A finalized **Research Plan** with clear sub-goals and deadlines.
        - An initial choice of **2–3** algorithms to be studied in detail (e.g., Kyber for encryption, Dilithium for signatures).
3. **Team Setup & Environment**
    - Configure the basic development and testing environment: CI/CD, repositories with PQC libraries.

- Assign roles (Crypto Engineer, Security Analyst, DevOps, Project Manager).
- **KPI:**
  - A **Git repository** created with initial dependencies and an integrated CI process.
  - Official or written **“kick-off”** meetings with all project participants.

> Parallel Start:
>
> As early as the second week, certain specialists may begin **prototyping** simple modules with the chosen PQC algorithms.

---

2. Phase 2: Proof of Concept (PoC)

Timeline: Weeks 3–4 (partially overlapping with Phase 1)

Objectives

1. **Initial Prototype with PQC**
   - Implement a **minimal working module** (PoC) using one selected algorithm (e.g., Kyber) for **encrypting** financial transactions, or Dilithium for **digital signatures**.
   - Compare with **classical** RSA/ECC in terms of key generation time, key size, and transaction latency.
   - **KPI:**
     - A working PoC script in GitHub.
     - An initial **Performance Report** (time measurements, key sizes, encryption/signing speed).
2. **Preliminary Security & Compatibility Checks**
   - Verify basic compatibility with **TLS libraries** and existing company protocols.
   - Assess whether changes are needed in **key management** (KMS/HSM, PKI certificates).

- **KPI:**
  - A concise **Compatibility Matrix** indicating which components support/do not support PQC.
  - **2–3** ideas for further optimization (e.g., hybrid encryption schemes).

---

3. Phase 3: Scaling & Integration

Timeline: Weeks 5–8

Objectives

1. **Data & Transaction Flow Preparation**
   - Identify the **actual financial data** or transaction flows (or similar test data) for integration testing.
   - Set up an **ETL pipeline** to handle a large volume of transactions (≥10^5 records) or to simulate a live stream.
   - **KPI:**
     - A validated and **cleaned** transaction/data set.
     - A functioning **test environment** capable of running high-load tests.
2. **Algorithm Optimization & Integration**
   - Implement optimizations: reduce circuit depth or **key sizes** (within security limits), configure caching, and enable multi-threading.
   - Integrate the PoC module into a **pilot version** of the financial platform (e.g., an updated API where encryption is performed by a PQC library).
   - **KPI:**
     - **20%** execution time compared to the initial PoC (optimization goal).
     - A complete **end-to-end scheme**: data → PQC encryption/signing → storage/transfer → decryption/verification.
3. **Hybrid Approach (Classical + PQC)**
   - If needed, implement **hybrid algorithms**: use classical methods and PQC in parallel (for **backward compatibility**).
   - **KPI:**

- Seamless operation of both algorithms (RSA/ECC + PQC) with minimal impact on transaction speed.
- Documentation with **recommendations** for various hybrid scenarios.

---

4. Phase 4: Testing & Optimization

Timeline: Weeks 9–10

Objectives

1. **Extensive Performance & Stress Testing**
   - Conduct large-scale load tests with a **high number of transactions**, simulating peak operations (e.g., 10–50 transactions/sec).
   - **KPI:**
     - Execution of **5–7** key test scenarios (stress tests, network failures, latency).
     - A **report** on average transaction time, percentage of successful operations, and critical latency points.
2. **Security & Compliance Audit**
   - Check compliance with internal security policies and **PCI DSS** or other financial standards.
   - Perform **Penetration Testing** or **Security Scans** on the integrated PQC module.
   - **KPI:**
     - An **Audit Report** listing discovered vulnerabilities and a plan to address them.
     - **0 critical** (or promptly resolved) security issues.
3. **Comparative Analysis (PQC vs. Classical)**
   - Compare the obtained PQC results with classical RSA/ECC (speed, key size, resistance to attacks).
   - **KPI:**

- A **Comparison Report** with tables/graphs illustrating advantages and drawbacks.
- **3–5 recommendations** on using PQC in various financial scenarios.

---

5. Phase 5: Final Implementation & Publications

Timeline: Weeks 11–12

Objectives

1. **Final MVP / Demonstration**
   - Consolidate all components into a **demo prototype** (CLI or a simple web demonstrator).
   - Showcase the module's operation to management and key stakeholders.
   - **KPI:**
     - An **MVP** complete with detailed instructions (README / brief manual).
     - A demo session (internal presentation with Q&A).
2. **Documentation & Possible Publications**
   - Prepare a **conceptual overview** of the solution, covering algorithms, efficiency metrics, and steps for financial integration.
   - If possible, produce a **scientific publication** (arXiv, IEEE, ACM) or an internal technical paper.
   - **KPI:**
     - **1–2** documents (academic paper, technical report) submitted or ready for submission.
     - 1 official **presentation** (internal meetup or industry workshop).
3. **Final Roadmap & Next Steps**
   - Create a **Final Report** with all findings, challenges, and recommendations for production-scale implementation.
   - Outline future research directions (side-channel resistance, hybrid schemes, key repositories).

- **KPI:**
  - A **10–15-page** report detailing the results.
  - **3–5** concrete ideas for subsequent R&D initiatives.

---

6. Timeline Summary

Phase	Weeks	Focus	KPIs
1. Research & Planning	1–2	Standards overview, goal setting, environment preparation	~10+ sources, 1 “Research Plan” doc, Git repo & CI created
2. Proof of Concept (PoC)	3–4	Minimal PQC prototype (encryption/signing), basic comparisons	PoC code, initial metrics, short Performance Report
3. Scaling & Integration	5–8	Scaling to large data/transactions, optimization, hybrid schemes	Dataset ≥10^5, -20% execution time vs. PoC, integrated pipeline
4. Testing & Optimization	9–10	Stress tests, security audit, comparison with classical methods	≥5 test scenarios, Audit Report, Comparison Report (PQC vs. RSA/ECC)
5. Final Implementation & Pubs	11–12	MVP demonstration, documentation, publications, final report	MVP + guide, 1–2 publications/reports, 10–15 p. final summary document

---

7. Key Success Factors

1. **Parallel Task Execution**
   - **Crypto Team** explores and integrates PQC libraries while **Infrastructure/DevOps** provides test environments, and **Security/Compliance** oversees adherence to standards.
2. **Frequent Checkpoints**
   - Short **weekly** or **bi-weekly** sessions with a “demo day”: the team discusses progress, identifies risks, and makes quick adjustments.

3. **Robust Documentation & Version Control**
    - A structured **Git repository** and automated tests (CI/CD) to track all changes and ensure consistent repeatability of results.
4. **Realistic Expectations & Priorities**
    - PQC is still in an active standardization phase, so focus on **hybrid solutions** and tested libraries rather than purely “experimental” algorithms.
5. **Publication & Visibility**
    - Emphasize **internal or external presentations** and **papers** to showcase project success, attract potential partners, and help the R&D department secure additional funding.

Conclusion

1. **Conduct research** and identify the optimal PQC algorithms for specific financial use cases.
2. **Develop and validate a PoC**, followed by scaling to large transaction volumes and data sets.
3. **Optimize and test** the solution under real loads while meeting security standards.
4. **Deliver a final MVP**, complete with documentation and publications, setting the stage for production deployment and enhancing the team’s reputation.