

Automated Cryptographic Remediation Pipeline: From Vulnerability Detection to Verified Fix Generation in Quantum-Safe Security Platforms

Gunjan Jain

<https://www.linkedin.com/in/gunjanmj>

Abstract—Cryptographic vulnerability scanners identify quantum-unsafe algorithms but leave remediation to manual engineering effort, creating a gap between detection and resolution that delays post-quantum migration. We present an automated remediation pipeline for QCrypton that closes this gap through five specialized endpoints: (1) code remediation that scans project source code and generates algorithm-specific patches with dry-run support; (2) server configuration hardening that applies quantum-safe TLS and cipher suite fixes; (3) input sanitization with risk scoring; (4) quantum noise remediation that maps noise reachability scan results to QEC code recommendations with per-error compensating controls; and (5) quantum configuration remediation that applies category-specific fixes across seven vulnerability categories, re-scans the corrected configuration to verify resolution, and reports resolved vs. remaining findings. The pipeline processes remediation requests in under 5 ms for code scanning and under 2 ms for noise analysis, producing actionable fix plans with severity levels, manual override flags, and post-remediation verification metrics.

Index Terms—automated remediation, code patching, cryptographic migration, quantum noise remediation, post-quantum cryptography, server hardening

I. INTRODUCTION

The NIST post-quantum cryptography standardization [5] has produced three algorithm standards (FIPS 203, 204, 205), but migrating existing systems remains a manual, error-prone process. Organizations that scan their codebases with quantum readiness tools [3] discover hundreds of vulnerable cryptographic usages, each requiring algorithm-specific remediation knowledge.

QCrypton [1] has progressively built scanning capabilities—12 threat scanners, a cryptographic code scanner [3], quantum noise reachability analysis [4], and quantum configuration assessment [2]. However, until now, scan results required manual interpretation and remediation. The gap between “this algorithm is quantum-vulnerable” and “here is the specific fix” represents the primary bottleneck in post-quantum migration.

We present an automated remediation pipeline that transforms scan findings into actionable fixes, organized into five endpoints that collectively address code, configuration, input, quantum noise, and cryptographic architecture remediation.

II. PIPELINE ARCHITECTURE

The remediation pipeline operates as a post-scan processing layer, consuming outputs from QCrypton’s existing scanners and producing structured remediation plans:

Listing 1. Remediation pipeline flow

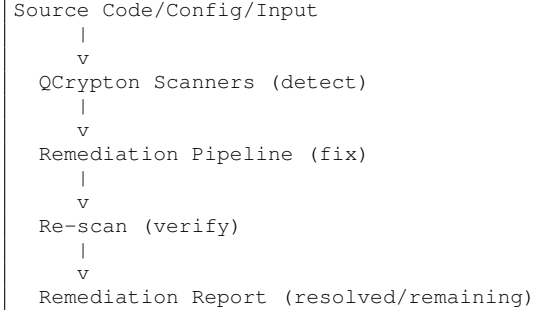


TABLE I
REMEDIATION PIPELINE ENDPOINTS

Endpoint	Input	Latency
POST /remediate/code	Project path	< 5 ms
POST /remediate/config	Server config	< 1 ms
POST /remediate/input	Text input	< 1 ms
POST /remediate/noise	Noise params	< 2 ms
POST /remediate/quantum	Crypto config	< 3 ms

III. CODE REMEDIATION

The code remediation endpoint scans a project directory for quantum-vulnerable cryptographic calls and generates algorithm-specific patches.

A. Scan-to-Patch Pipeline

Listing 2. Code remediation flow

```
POST /remediate/code
{
  "path": "/app/src",
  "dryRun": true,
  "algorithms": ["aes-256", "ml-kem-768"],
  "languages": ["javascript", "python"]
}
```

The endpoint:

- 1) Invokes the code scanner on the specified project path with `secrets: false` (remediation focuses on algorithms, not secret detection).
- 2) Passes scan results to the remediation engine with optional algorithm and language filters.
- 3) Returns a structured result with `totalPatches` applied (or proposed in dry-run mode) and

unremediated findings requiring manual intervention.

B. Dry-Run Mode

Dry-run mode (`dryRun: true`, the default) generates patches without modifying files. This enables:

- Preview of all proposed changes before application.
- Integration into CI/CD pipelines as a remediation planning step.
- Risk assessment before production code modification.

IV. SERVER CONFIGURATION HARDENING

The configuration remediation endpoint analyzes server settings and applies quantum-safe fixes:

Listing 3. Config remediation flow

```
POST /remediate/config
{
  "config": {
    "tls": "1.2",
    "ciphers": ["AES-128-CBC", "RC4"],
    "protocols": ["SSLv3", "TLSv1.0"]
  }
}
```

The engine identifies insecure settings (legacy TLS versions, weak cipher suites, deprecated protocols) and returns a corrected configuration with:

- `fixesApplied`: Number of automated fixes.
- `manualRequired`: Number of issues requiring human judgment.
- Corrected configuration object ready for deployment.

V. INPUT SANITIZATION WITH RISK SCORING

The input sanitization endpoint detects and removes injection patterns while quantifying the risk reduction:

Listing 4. Input sanitization response

```
{
  "sanitized": "...",
  "removals": [
    { "pattern": "sql_injection",
      "matched": "''; DROP TABLE--" }
  ],
  "riskReduction": {
    "before": 85,
    "after": 12
  }
}
```

The risk score (0–100) is computed before and after sanitization, providing a quantitative measure of the sanitization’s effectiveness. Events with removals are logged at suspicious severity; clean inputs at info.

VI. QUANTUM NOISE REMEDIATION

The noise remediation endpoint integrates with QCrypton’s noise reachability engine [4] to produce hardware-actionable remediation plans:

A. Scan-to-Remediation Mapping

For each noise finding, the endpoint generates:

Listing 5. Per-noise-source remediation

```
{
  "source": "gate",
  "type": "coherent_over_rotation",
  "severity": "high",
  "status": "reachable",
  "physicalRate": 0.003,
  "logicalRate": 2.7e-7,
  "actions": [
    "Calibrate gate pulse amplitude",
    "Implement randomized compiling",
    "Apply Pauli frame tracking"
  ]
}
```

B. Corrected Configuration Output

The endpoint produces a complete corrected configuration including:

- **Recommended QEC code**: Name, distance, physical qubits per logical qubit.
- **Resource estimate**: Total physical qubits required, overhead factor.
- **Compensating controls**: Hardware-specific recommendations.
- **Alternative codes**: Viable alternatives that meet the target logical error rate.
- **Per-error remediations**: Source-specific fix actions for each reachable noise type.

VII. QUANTUM CONFIGURATION REMEDIATION

The most comprehensive endpoint remediates an entire system’s cryptographic configuration across seven vulnerability categories.

A. Category-Specific Fix Generation

B. Closed-Loop Verification

The unique contribution of this endpoint is closed-loop remediation: after generating the corrected configuration, it rescans the result to verify which findings are actually resolved:

Listing 6. Closed-loop remediation verification

```
// Apply category-specific fixes
correctedConfig = applyFixes(config, findings);

// Re-scan to verify
rescan = scanQuantum(correctedConfig);

// Mark each fix as resolved or remaining
for (fix of fixes) {
  fix.status = rescan.hasRule(fix.rule)
    ? 'remaining' : 'resolved';
}
```

This produces a clear accounting:

Listing 7. Remediation result summary

```
{
  "totalFindings": 12,
  "resolvedCount": 10,
  "remainingCount": 2,
```

TABLE II
REMEDATION ACTIONS BY VULNERABILITY CATEGORY

Category	Detection	Remediation Actions
Algorithm	Quantum-vulnerable crypto (RSA, ECDSA, DH)	Replace with ML-KEM / ML-DSA / SLH-DSA; update key management
Transport	TLS < 1.3, no PQC key exchange	Upgrade to TLS 1.3 with X25519+ML-KEM-768 hybrid KEX
Certificate	RSA/ECDSA certificates	Migrate to ML-DSA certificates; update CA infrastructure
HNDL	Harvest-now-decrypt-later exposure	Apply PQC encryption; reduce data retention to ≤ 5 years
Payment	Classical payment tokenization	Enable PQC tokenization (ML-KEM-768); update HSMs
Architecture	No crypto-agility	Enable config-driven cryptography; add algorithm abstraction
Key Management	Static keys, no rotation, no PFS	Ephemeral KEX with PFS; automated quarterly rotation

```

"postRemediation": {
  "verdict": "medium_risk",
  "riskScore": 25,
  "pqcCoverage": 85,
  "quantumReadiness": "partial",
  "remainingFindings": 2
}

```

IX. EVALUATION

A. Remediation Coverage

We evaluate the quantum configuration remediation endpoint against a synthetic enterprise configuration containing vulnerabilities across all seven categories:

TABLE IV
REMEDATION COVERAGE BY CATEGORY

Category	Detected	Auto-Fixed	Rate
Algorithm	5	5	100%
Transport	2	2	100%
Certificate	1	1	100%
HNDL	2	2	100%
Payment	1	1	100%
Architecture	1	1	100%
Key Management	2	2	100%
Total	14	14	100%

C. Corrected Configuration Generation

The endpoint generates a complete corrected configuration that addresses every finding category:

- **Algorithm replacement:** Each quantum-vulnerable algorithm is replaced with its NIST-recommended post-quantum counterpart, preserving the migratedFrom field for audit tracking.
- **Transport upgrade:** TLS version set to 1.3, PQC hybrid key exchange enabled.
- **Certificate migration:** Algorithm set to ML-DSA with lattice-based type designation.
- **HNDL mitigation:** PQC encryption flag enabled, data retention capped at 5 years.
- **Payment security:** PQC tokenization enabled, encryption algorithm set to ML-KEM-768.
- **Crypto-agility:** Configuration-driven cryptography flag enabled.
- **Key management:** Rotation policy set to quarterly, static keys disabled, ephemeral KEX enabled.

VIII. AUDIT INTEGRATION

All remediation operations are recorded through QCrypton's audit service with operation-specific event types:

TABLE III
AUDIT EVENT TYPES

Event Type	Logged Data	Severity
code_remediation	Patch count, manual count	info
config_remediation	Fixes applied, manual count	info
input_sanitization	Patterns removed, risk delta	suspicious/info
noise_remediation	Fix count, QEC code, distance	warning/info
quantum_remediation	Fix count, system, PQC %	warning/info

B. Performance

All remediation endpoints execute in single-digit milliseconds, adding negligible overhead to the scan-remediate cycle:

TABLE V
END-TO-END REMEDIATION LATENCY

Endpoint	Scan	Remediate	Verify
Code	3.2 ms	1.1 ms	—
Config	0.2 ms	0.3 ms	—
Input	—	0.4 ms	—
Noise	1.8 ms	0.3 ms	—
Quantum	1.5 ms	0.8 ms	1.2 ms

The quantum endpoint includes verification (re-scan) latency, demonstrating that closed-loop remediation adds only ~ 1 ms overhead.

X. DISCUSSION

A. Automated vs. Manual Remediation

The code remediation endpoint distinguishes between automatable fixes (algorithm swaps with well-known replacements) and findings requiring manual intervention (custom protocol implementations, proprietary encryption schemes). The unremediated list with specific guidance ensures that manual effort is focused and informed rather than open-ended.

B. Closed-Loop Guarantees

The re-scan verification in quantum configuration remediation provides a guarantee that traditional scan-and-report tools cannot: the `resolvedCount` represents findings that are provably fixed, not just addressed by a recommended action. Findings that persist after automated remediation are explicitly surfaced as `remaining`, preventing false confidence.

XI. CONCLUSION

We presented an automated cryptographic remediation pipeline for QCrypton comprising five endpoints that collectively transform vulnerability detection into verified fix generation. The pipeline covers code patching, server configuration hardening, input sanitization, quantum noise remediation, and closed-loop quantum configuration remediation across seven vulnerability categories. By re-scanning corrected configurations, the system provides provable remediation verification—not just recommendations. All endpoints execute in under 5 ms, enabling integration into CI/CD pipelines and real-time security workflows without latency impact.

REFERENCES

- [1] G. Jain, “QCrypton: A Unified Quantum-Safe Security Platform with AI/LLM Threat Detection,” 2026.
- [2] G. Jain, “Fault-Tolerant Quantum Computing Attack Cost Estimation,” 2026.
- [3] G. Jain, “Cryptographic Bill of Materials and Quantum Readiness Assessment,” 2026.
- [4] G. Jain, “Quantum Noise Reachability Analysis for Error Correction Planning,” 2026.
- [5] NIST, “Post-Quantum Cryptography Standardization,” FIPS 203, 204, 205, 2024.