

QCrypton v4.0: Unified Quantum-Safe Security Platform with Noise Reachability Analysis and Enhanced QKD Eavesdrop Detection

Gunjan Jain

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

Abstract—We present QCrypton v4.0, extending the unified quantum-safe security platform with two major capabilities. First, a Quantum Noise Reachability Engine that classifies physical qubit noise into three source categories (channel, gate, environment) with 12 sub-types, evaluates each against six quantum error correction code families at multiple code distances (720 evaluations per scan), determines which errors propagate past correction to become logical errors, and recommends the optimal QEC strategy with 24 source-specific compensating controls. Second, an enhanced BB84 eavesdrop detection module that replaces the standard fixed 11% error rate threshold with statistical noise decomposition, demonstrating detection of an eavesdropper at 9.8σ significance when standard BB84 reported “secure.” The platform now comprises 12 threat scanners, a post-quantum cryptographic toolkit implementing NIST FIPS 203/204/205, a fault-tolerant quantum computing attack cost estimation engine covering 28 algorithms, quantum noise reachability analysis, a multi-language cryptographic code scanner with CBOM generation, and automated remediation—all accessible via CLI, SDK, and REST API from a single codebase with zero third-party cryptographic dependencies. We evaluate the v4.0 additions, demonstrating noise reachability scans in under 2ms and enhanced eavesdrop detection in under 0.1 ms with no increase in external dependencies.

Index Terms—quantum-safe security, quantum error correction, noise reachability, BB84, post-quantum cryptography, unified security platform

I. Introduction

The convergence of quantum computing advances and AI-driven security threats demands comprehensive security platforms that address both domains simultaneously. QCrypton was introduced as a unified platform combining AI/LLM threat detection with quantum cryptographic security [1], and has been progressively extended with FTQC attack cost estimation [2], MCP tool poisoning detection [3], a zero-dependency post-quantum cryptographic toolkit [4], cryptographic bill of materials generation [5], QKD-protected secret reporting [6], and stateless cloud-native deployment [7].

As quantum computing advances toward fault-tolerant operation, a new category of analysis becomes essential: quantum error correction noise analysis. Organizations deploying or planning to deploy quantum hardware need to understand not just “what is my error rate?” but “which errors actually matter?”—i.e., which physical errors propagate past QEC codes to corrupt computation. Simul-

taneously, as QKD networks are deployed in real-world noisy channels, the standard BB84 eavesdrop detection threshold proves inadequate for distinguishing legitimate noise from eavesdropper signals.

QCrypton v4.0 addresses both needs with two new capabilities, maintaining the platform’s design principles of zero third-party cryptographic dependencies, three-mode deployment (CLI/SDK/API), and consistent scanner-pattern output.

A. Version History

TABLE I
QCrypton Version History

Version	Key Additions	Date
v1.0	11 scanners, quantum crypto, FTQC engine	Apr 18, 2026
v2.0	Multi-tenant RBAC, code scanner, CBOM, HSM	Apr 19, 2026
v3.0	Domain provisioning, permissions, K8s deploy	Apr 21, 2026
v4.0	Noise reachability, enhanced BB84 detection	Apr 23, 2026

II. Platform Architecture

QCrypton v4.0 maintains the seven-layer stateless architecture introduced in v3.0 [7], with the noise reachability engine added at Layer 4 alongside the existing threat scanners:

Listing 1. QCrypton v4.0 seven-layer architecture

Layer 7: Kubernetes / Docker deployment
Layer 6: Web Dashboard + REST API (17 routes)
Layer 5: Authentication & Multi-Tenancy
Layer 4: Threat Scanners (12) + Quantum Engines
Layer 3: Quantum Crypto Engine
Layer 2: PostgreSQL Persistence
Layer 1: Node.js built-in crypto (zero deps)

A. Module Inventory

III. New Capability 1: Noise Reachability Engine

The Quantum Noise Reachability Engine introduces a novel analysis pipeline for quantum error correction, comprising four layers:

TABLE II
QCrypton v4.0 Module Inventory

Component	Count
Threat scanners	12 (11 original + 1 new)
Quantum modules	7 (quantum-crypto, quantum-cost, noise-reachability, qkd, lsh, hpke, hsm)
API route modules	17
CLI commands	12
Sequelize models	11
Service modules	9
Total source lines	~13,000+

TABLE III
Noise Source Taxonomy

Source	Sub-Types	Default %
Channel	Depolarizing, Dephasing, Amplitude Damping, Erasure	45, 35, 15, 5
Gate	Coherent, Crosstalk, Leakage, Measurement	40, 25, 15, (separate)
Environment	Thermal, Cosmic Ray, EMI, Calibration Drift	30, 10, 35, 25

A. Layer 1: Noise Classification

The `classifyNoise()` function decomposes aggregate error rates into a structured taxonomy of three source categories with four sub-types each:

The classifier also detects:

- Noise bias: The ratio $\eta = p_{\text{dephasing}}/p_{\text{bit-flip}}$ identifies Z-biased ($\eta > 10$), unbiased ($0.33 < \eta < 3$), or X-biased ($\eta < 0.1$) noise, enabling tailored code selection.
- Correlated noise: Cosmic ray rates above 10^{-5} trigger a flag indicating that the independent-error assumption may be violated.

B. Layer 2: Reachability Analysis

The `analyzeReachability()` function evaluates each of the 12 noise sub-types against six QEC code families at 10 code distances, producing 720 individual evaluations per scan.

For each evaluation, the logical error rate is computed using the code-specific formula (e.g., for the surface code: $p_L = 0.1 \cdot (p/0.01)^{(d+1)/2}$). A noise type is classified as “reachable” if its best achievable logical error rate exceeds the target (default 10^{-10}).

The analysis produces a reachability ratio—the fraction of noise types that propagate past QEC—analogous to the fraction of vulnerabilities that are truly exploitable in a network.

Five-stage error propagation paths trace each error from its physical source through syndrome extraction and QEC decoding to the logical qubit, with a suppression factor $S = \log_{10}(p_{\text{physical}}/p_{\text{logical}})$ quantifying correction effectiveness.

C. Layer 3: Correction Planning

The `recommendCorrection()` function selects the optimal QEC code based on the noise profile:

- Strongly Z-biased noise ($\eta > 10$) → Repetition code (minimal overhead)
- Correlated burst noise → Color code (handles spatial correlations)
- Very low error rate ($< 3 \times 10^{-4}$) → Steane $[[7, 1, 3]]$ (7:1 overhead)
- General noise below threshold → Surface code (highest threshold)

The engine computes the minimum code distance, physical qubit overhead, and integrates with QCrypton’s existing FTQC cost estimation engine [2] for complete resource analysis including magic state distillation.

Additionally, 24 source-specific compensating controls are recommended beyond code distance increase, including dynamical decoupling for dephasing, randomized compiling for coherent errors, leakage reduction units, and cosmic ray detection protocols.

D. Layer 4: Scan Interface

The `scanNoiseReachability()` function provides the full pipeline as a single call, producing output consistent with QCrypton’s scanner pattern:

Listing 2. Noise reachability scan output structure

```
{
  "scanId": "uuid",
  "scanner": "noise-reachability",
  "verdict": "all_correctable",
  "riskScore": 0,
  "findings": [],
  "noiseProfile": { ... },
  "reachabilityAnalysis": {
    "reachabilityRatio": "0.0%"
  },
  "correctionPlan": {
    "primaryCode": "Surface Code",
    "distance": 17
  }
}
```

E. API Endpoints

Six new REST API endpoints are added:

TABLE IV
Noise Reachability API Endpoints

Endpoint	Function
POST /api/noise/scan	Full pipeline scan
POST /api/noise/classify	Noise classification only
POST /api/noise/reachability	Reachability analysis only
POST /api/noise/recommend	QEC recommendation only
POST /api/noise/eavesdrop-detect	Enhanced BB84 detection
GET /api/noise/codes	List QEC codes + taxonomy

IV. New Capability 2: Enhanced BB84 Eavesdrop Detection

A. Problem

Standard BB84 detects eavesdropping when the observed quantum bit error rate (QBER) exceeds 11% [8]. This threshold is theoretically optimal for worst-case analysis but practically limited: it does not account for the known baseline noise of the quantum channel.

B. Method

The `enhancedEavesdropDetection()` function decomposes the observed error rate into expected noise contributions and tests the residual:

- 1) Decompose expected noise: $p_{\text{exp}} = p_{\text{channel}} + p_{\text{dark}} + p_{\text{align}} + p_{\text{jitter}}$
- 2) Compute anomalous rate: $e_{\text{anom}} = \max(0, e_{\text{obs}} - p_{\text{exp}})$
- 3) Compute significance: $z = e_{\text{anom}}/\sigma$
- 4) Verdict: $z > 5\sigma = \text{detected}$, $z > 3\sigma = \text{suspected}$, $z > 2\sigma = \text{elevated}$, else = noise only

C. Demonstrated Result

TABLE V
Enhanced BB84 Detection: Key Result

Metric	Standard	Enhanced
Observed QBER	6.88%	6.88%
Threshold	11% (fixed)	Statistical
Anomalous rate	—	4.88%
Significance	—	9.8σ
Verdict	SECURE	DETECTED
Correct?	No	Yes

With Eve intercepting 30% of qubits and channel noise of 2%, the standard threshold misses the eavesdropper entirely. The enhanced method detects with overwhelming confidence.

V. Integration Design

The v4.0 additions integrate with existing QCrypton modules through three connection points:

1. `noise-reachability.js` \leftrightarrow `quantum-cost.js`: The noise reachability engine calls `estimatePhysicalResources()` from the FTQC cost engine to compute total physical qubit overhead for recommended QEC codes. This reuses the existing surface code model with magic state distillation [2].
2. `enhancedEavesdropDetection` \leftrightarrow `qkd.js`: The enhanced detection function takes the output of `distributeKey()` (BB84 key distribution result) as input, adding noise-aware analysis on top of the existing QKD implementation [6].
3. Scanner pattern consistency: The noise reachability scanner follows the same output pattern as all 11 existing scanners: `{findings, riskScore, verdict}` with severity weights (critical=30, high=20, medium=10, low=5) and verdict thresholds (clean/low_risk/suspicious/critical).

VI. Complete Feature Matrix

VII. Evaluation

A. New Capability Performance

B. Dependency Analysis

The v4.0 additions introduce zero new external dependencies. The noise reachability engine uses only `uuid` (already a dependency) and the existing `quantum-cost.js` module.

C. Test Coverage

VIII. Related Work

IBM Quantum Safe Explorer provides quantum risk assessment for cryptographic algorithms but does not include AI/LLM threat detection, QEC noise analysis, or code scanning.

Cryptosense Analyzer performs cryptographic code analysis but lacks quantum attack cost estimation, noise reachability analysis, and threat scanning.

Stim [10] enables fast QEC circuit simulation but is a simulation tool, not an analysis/recommendation engine—it does not classify noise, recommend codes, or integrate with security scanning.

TQEC compiles fault-tolerant circuits but does not perform noise analysis or code selection.

No existing platform combines AI/LLM threat detection, post-quantum cryptography, FTQC cost estimation, QEC noise reachability analysis, enhanced QKD eavesdrop detection, cryptographic code scanning, and automated remediation in a unified system.

IX. Conclusion

QCrypton v4.0 extends the unified quantum-safe security platform with two novel capabilities that address emerging needs in quantum computing: noise reachability analysis for QEC and enhanced eavesdrop detection for QKD. The noise reachability engine provides a complete pipeline from noise classification through reachability analysis to correction recommendation, while the enhanced BB84 detection overcomes the limitations of fixed-threshold eavesdrop detection in noisy real-world channels.

With 12 threat scanners, 7 quantum modules, 17 API routes, and ~13,000+ lines of code—all using zero third-party cryptographic dependencies—QCrypton v4.0 represents the most comprehensive unified quantum-safe security platform available, spanning from AI threat detection to quantum error correction analysis.

References

- [1] G. Jain, “QCrypton: A unified quantum-safe security and AI threat detection platform,” TechRxiv Preprint, 2026.
- [2] G. Jain, “Fault-tolerant quantum computer attack cost estimation engine,” TechRxiv Preprint, 2026.
- [3] G. Jain, “MCP tool poisoning detection with rug pull indicators,” TechRxiv Preprint, 2026.
- [4] G. Jain, “Zero-dependency post-quantum cryptographic toolkit,” TechRxiv Preprint, 2026.

TABLE VI
QCCrypton v4.0 Complete Feature Matrix

Category	Capabilities
Threat Scanners (12)	Prompt injection (22+ patterns), tool poisoning (8 checks incl. rug pull), data exfiltration (30+ secret types), server config audit, quantum vulnerability (45+ algorithms), brute force, credential stuffing, dictionary attack, phishing, keylogger/malware, card fraud, noise reachability (NEW)
Post-Quantum Crypto	AES-256-GCM + HKDF-SHA3-256, ChaCha20-Poly1305, SHA3-256/512, SHAKE256, LSH-256 (KS X 3262), HMAC-LSH-256, HMAC-SHA3-256, HPKE (RFC 9180: Base/PSK/Auth/AuthPSK), BB84 QKD simulation
HSM/KMS	ML-KEM-512/768/1024 (FIPS 203), ML-DSA-44/65/87 (FIPS 204), SLH-DSA-SHA2-128s/192s/256s (FIPS 205); Backends: Entrust nShield, AWS KMS, Azure KV, GCP KMS, Thales Luna, Software HSM
FTQC Cost Estimation	28 algorithms (RSA, ECC, AES, ChaCha20, SHA, post-quantum); surface code + magic state distillation model; feasibility timeline (2030/2035/2040); batch + comparison modes
Noise Reachability (NEW)	3 source categories \times 4 sub-types = 12 noise types; 6 QEC codes; 720 evaluations per scan; reachability ratio; error propagation paths; suppression factors; 24 compensating controls; noise bias + correlation detection; FTQC resource integration
Enhanced BB84 (NEW)	Statistical noise decomposition; per-channel adaptive baseline; 4-level detection verdict; demonstrated 9.8σ detection where standard threshold misses
Code Scanner + CBOM	6 languages (JS/TS, Python, Go, Java, Rust, C/C++); 88+ crypto patterns; 30+ secret patterns; binary analysis; CycloneDX CBOM; SPDX; CI/CD quality gate
Remediation	Auto-patches across 6 languages (MD5 \rightarrow SHA3, DES \rightarrow AES-256-GCM, RC4 \rightarrow ChaCha20); config remediation; input sanitization; middleware mode
Auth & Multi-Tenancy	JWT + OAuth (Google/GitHub/Azure/Okta) + API keys (SHA3-256) + MFA (TOTP + WebAuthn/FIDO2); domain-verified provisioning; 5 RBAC roles; per-user permission overrides
Deployment	CLI (12 commands) + SDK (Node.js, in-process) + REST API (17 routes); Kubernetes-native (HPA 2-10 replicas, PDB, health probes); Docker multi-stage; fully stateless

TABLE VII
v4.0 Performance Metrics

Operation	Time	Notes
classifyNoise()	< 0.1 ms	Pure function, $O(1)$
analyzeReachability()	< 1 ms	720 evaluations
recommendCorrection()	< 0.5 ms	With FTQC estimate
scanNoiseReachability()	< 2 ms	Full pipeline
enhancedEavesdropDetection()	< 0.1 ms	Pure function, $O(1)$

TABLE VIII
Dependency Comparison Across Versions

Metric	v3.0	v4.0	Change
Third-party crypto packages	0	0	None
Runtime dependencies	8	8	None
Security-critical npm packages	0	0	None
New lib/ modules	—	1	noise-reachability.js
New route modules	—	1	noiseReachability.js
New CLI commands	—	1	noise

TABLE IX
Test Coverage (Cumulative)

Test Suite	Count
LSH-256 KAT vectors (KCMVP)	22
HPKE RFC 9180 roundtrip tests	12
Fuzz tests (random/malformed inputs)	19,000+
Noise reachability scenarios (v4.0)	3 validated
Enhanced BB84 scenarios (v4.0)	12 validated

032324, 2012.

- [10] C. Gidney, “Stim: a fast stabilizer circuit simulator,” Quantum, vol. 5, p. 497, 2021.

- [5] G. Jain, “Cryptographic bill of materials for quantum readiness,” TechRxiv Preprint, 2026.
[6] G. Jain, “QKD-protected secret reporting,” TechRxiv Preprint, 2026.
[7] G. Jain, “Stateless cloud-native architecture for quantum-safe security platforms,” TechRxiv Preprint, 2026.
[8] P. W. Shor and J. Preskill, “Simple proof of security of the BB84 quantum key distribution protocol,” Physical Review Letters, vol. 85, no. 2, pp. 441–444, 2000.
[9] A. G. Fowler et al., “Surface codes: Towards practical large-scale quantum computation,” Physical Review A, vol. 86, no. 3, p.