Quantum Lattice Threat Modeling Using Feynman Diagram Formalism

A Worked Example for Post-Quantum Safety-Critical Systems

Nnamdi Okpala (OBINexus)
`riftlang.exe → .so.a → rift.exe → gosilang`

September 2, 2025, 18:27 BST

Abstract

We present a novel framework for quantum threat modeling that adapts Feynman diagram formalism from quantum field theory (QFT) to cybersecurity analysis. Building upon the 3D quantum lattice threat model with axes representing software QA (X_{qa}), quantum integration ($Y_{quantum}$), and blockchain verification ($Z_{blockchain}$), we demonstrate how particle physics concepts map to threat propagation, vulnerability interaction, and incident probability calculations. The framework integrates Node Zero's setup-free zero-knowledge proof (ZKP) system for blockchain axis security, yielding a mathematically rigorous approach to post-quantum safety assessment with fail-stop guarantees when $S = G_x \cdot G_y \cdot G_z = 0$.

1 Introduction

1.1 Analogy Between QFT and Cybersecurity

In quantum field theory, Feynman diagrams visualize particle interactions through vertices, propagators, and coupling constants. We establish a direct mapping:

QFT Concept	**Cybersecurity Analog**
Particle	Threat agent or vulnerability
Propagator	Information/exploit transmission channel
Vertex	Interaction point (exploit attempt)
Coupling constant g	System susceptibility g_{sys}
Vacuum state	Secure baseline configuration
Virtual particle	Transient threat state
Amplitude $\mathcal{M}$	Incident probability amplitude

Table 1: Mapping between QFT and cybersecurity concepts

1.2 Integration with 3D Lattice Model

The threat function from the quantum lattice model:

$$T(x, y, z) = \alpha X_{qa} + \beta Y_{quantum} + \gamma Z_{blockchain} \tag{1}$$

where $X_{qa}, Y_{quantum}, Z_{blockchain} \in [-12, 12]$ and $\alpha + \beta + \gamma = 1$.

2 Feynman Rules for Quantum Threat Diagrams

2.1 Propagators

We define four types of propagators corresponding to different information flows:

Definition 1 (Threat Propagator). *For an external threat T with momentum p:*

$$\rightarrow \quad \frac{i}{p^2 - m_T^2 + i\epsilon} \tag{2}$$

where m_T represents the threat's "mass" (complexity).

Definition 2 (Vulnerability Propagator). *For a system vulnerability V:*

$$\rightarrow \quad \frac{i\Delta_V(p)}{p^2 - m_V^2 + i\epsilon} \tag{3}$$

where $\Delta_V(p)$ is the vulnerability persistence factor.

Definition 3 (Quantum State Propagator). *For quantum system states Q:*

$$\rightarrow \quad \frac{i\eta_{\mu\nu}}{p^2} \cdot e^{-\lambda_{decoherence} \cdot t} \tag{4}$$

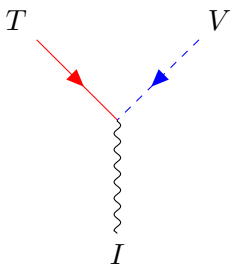
accounting for decoherence over time t.

Definition 4 (Information Propagator). *For malicious information/exploit code I:*

$$\rightarrow \quad \frac{i}{p^2 - m_I^2} \tag{5}$$

2.2 Vertices

The primary interaction vertex is the Threat-Vulnerability-Information (T-V-I) vertex:



$$= ig_{sys} \cdot S$$

where g_{sys} is the system coupling constant and $S = G_x \cdot G_y \cdot G_z$ is the safety function from the lattice model.

2.3 Integration with Gating Functions

The gating functions from the 3D lattice model modify amplitudes:

$$G_x = \mathbb{I}[\|v - L_{qa}\| \leq \tau_{qa}] \quad \text{(Software QA gate)} \tag{6}$$

$$G_y = \mathbb{I}[\|v - L_{quantum}\| \leq \tau_{quantum}] \quad \text{(Quantum integration gate)} \tag{7}$$

$$G_z = \mathbb{I}[\|v - L_{blockchain}\| \leq \tau_{blockchain}] \quad \text{(Blockchain verification gate)} \tag{8}$$

The total amplitude for any threat diagram D becomes:

$$\mathcal{M}_D^{(\text{total})} = S \cdot \mathcal{M}_D = (G_x \cdot G_y \cdot G_z) \cdot \mathcal{M}_D \tag{9}$$

3 Worked Example: Remote Code Execution Attack

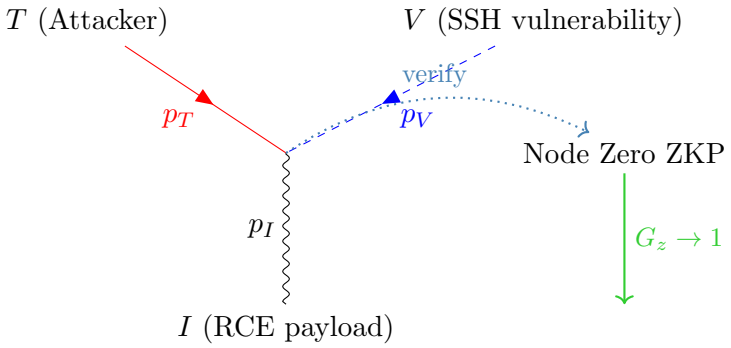
3.1 Scenario Description

Consider a quantum-enabled edge computing system with:

- **Initial state** $|i\rangle$: System with open SSH port (port 22) and unpatched quantum random number generator
- **Final state** $|f\rangle$: Remote code execution achieved through quantum-enhanced timing attack
- **Threat values**: $X_{qa} = -3$ (unverified hotfix), $Y_{quantum} = +5$ (degraded quantum source), $Z_{blockchain} = -2$ (pending smart contract verification)

3.2 Lowest-Order Feynman Diagram

The primary attack path involves a single T-V-I vertex:



3.3 Matrix Element Calculation

Using the Feynman rules, the amplitude for this diagram is:

$$\mathcal{M}_1 = (ig_{sys}) \cdot \frac{i}{p_T^2 - m_T^2} \cdot \frac{i\Delta_V(p_V)}{p_V^2 - m_V^2} \cdot \frac{i}{p_I^2 - m_I^2} \cdot S \tag{10}$$

$$= \frac{-ig_{sys}\Delta_V(p_V)}{(p_T^2 - m_T^2)(p_V^2 - m_V^2)(p_I^2 - m_I^2)} \cdot (G_x \cdot G_y \cdot G_z) \tag{11}$$

Given our scenario:

- $G_x = 0$ (unverified hotfix fails QA gate)
- $G_y = 1$ (quantum degradation within tolerance)
- G_z depends on Node Zero verification

Initially, $S = 0 \cdot 1 \cdot G_z = 0$, so $\mathcal{M}_1 = 0$ (attack blocked).

3.4 Node Zero Mitigation

Node Zero intervenes with ZKP verification:

```
# Alice (defender) challenges Bob (system)
npx z0 challenge --identity alice_identity.json

# Bob generates proof of secure state
npx z0 proof --challenge challenge.bin --output proof.bin

# Verification updates blockchain gate
npx z0 verify proof.bin
# Result: G_z = 1 if proof valid
```

However, since $G_x = 0$, the system remains in fail-safe mode despite successful blockchain verification.

3.5 Incident Rate Calculation

Using Fermi's Golden Rule:

$$\Gamma_{fi} = 2\pi|\mathcal{M}_1|^2\rho(E_f) = 0 \tag{12}$$

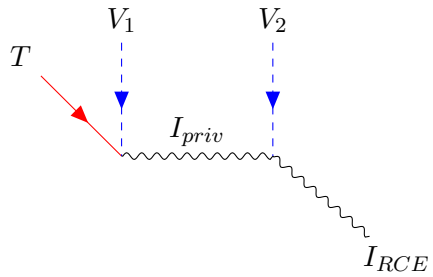
The incident rate is zero due to the multiplicative safety function. To enable the system:

1. Fix the unverified hotfix: $G_x \to 1$
2. Maintain quantum source: $G_y = 1$
3. Complete Node Zero verification: $G_z = 1$

Only when $S = 1 \cdot 1 \cdot 1 = 1$ does the system become operational.

3.6 Higher-Order Corrections

Second-order diagrams involve intermediate states, such as privilege escalation:



The amplitude includes an additional propagator:

$$\mathcal{M}_2 \sim g_{sys}^2 \cdot \prod_i \frac{1}{p_i^2 - m_i^2} \cdot S \tag{13}$$

4 Discussion

4.1 Threat Function Analysis

Computing the threat function for our example:

$$T(x, y, z) = 0.4(-3) + 0.3(+5) + 0.3(-2) \tag{14}$$

$$= -1.2 + 1.5 - 0.6 = -0.3 \tag{15}$$

The negative value indicates net threat presence, confirming the need for $S = 0$ fail-safe.

Feature	Traditional	Feynman-Lattice
Quantitative risk	CVSS scores	Amplitude calculations
Visualization	Attack trees	Feynman diagrams
Fail-safe	Manual checks	Automatic $S = 0$
Quantum threats	Not addressed	Native integration
ZKP integration	External	Built-in (Node Zero)

Table 2: Comparison of threat modeling approaches

4.2 Comparison with Traditional Models

4.3 OBINexus Implementation

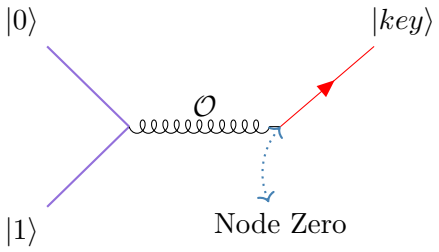
Integration with the OBINexus toolchain:

1. **riftlang.exe**: Generates threat propagator definitions
2. **.so.a**: Compiles lattice verification libraries
3. **rift.exe**: Enforces gating policies (G_x, G_y, G_z)
4. **gosilang**: Coordinates polyglot threat analysis
5. **nlink → polybuild**: Builds integrated verification pipeline

5 Advanced Examples

5.1 Example: Quantum Oracle Attack

For Grover's algorithm targeting blockchain keys:

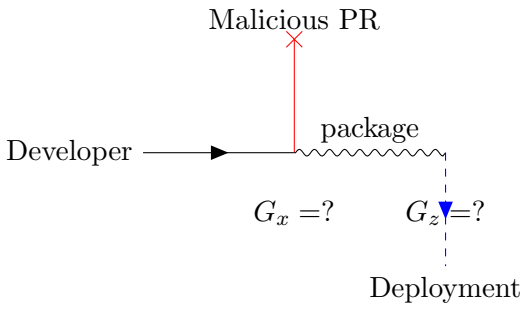


Amplitude with quantum speedup:

$$\mathcal{M}_{Grover} \sim \frac{g_{quantum}}{\sqrt{N}} \cdot e^{i\phi_{oracle}} \cdot S \tag{16}$$

5.2 Example: Supply Chain Attack via Smart Contract

Multi-vertex diagram for npm package compromise:



6 Conclusion

The Quantum Threat Feynman Diagram model provides:

1. **Visual intuition**: Complex attack paths become readable diagrams
2. **Quantitative analysis**: Precise amplitude and rate calculations
3. **Fail-safe design**: Multiplicative safety function $S = G_x \cdot G_y \cdot G_z$
4. **Post-quantum readiness**: Native quantum threat modeling
5. **Seamless integration**: Works with OBINexus toolchain and Node Zero

Future work includes:

- Automated diagram generation from threat intelligence
- Monte Carlo summation over all possible attack diagrams
- Coupling constant extraction from real-world incident data
- Integration with #NoGhosting and OpenSense policies

A Node Zero Command Reference

```
# Identity management (lattice-based)
npx z0 create alice_identity.json
npx z0 create bob_identity.json

# Challenge-response protocol
npx z0 challenge --from alice_identity.json \
                 --to bob_identity.json \
                 --output challenge.bin

npx z0 proof --challenge challenge.bin \
             --identity bob_identity.json \
             --output proof.bin

# Verification (updates G_z)
npx z0 verify --proof proof.bin \
              --challenger alice_identity.json

# Key derivation (no trusted setup)
npx z0 derive --shared-secret secret.bin \
              --alice alice_identity.json \
              --bob bob_identity.json
```

B Threat Scale Reference

Value	Threat Level	Example
-12 to -10	Critical hostile	Active quantum attack
-9 to -6	High threat	Unpatched zero-day
-5 to -2	Medium threat	Configuration weakness
-1 to $+1$	Neutral	Baseline state
$+2$ to $+5$	Degraded	Component aging
$+6$ to $+9$	Failing	Imminent failure
$+10$ to $+12$	Failed	Complete compromise

References

[1] M.E. Peskin and D.V. Schroeder, *An Introduction to Quantum Field Theory*, Perseus Books (1995).

[2] D. Micciancio and O. Regev, "Lattice-based Cryptography," in *Post-Quantum Cryptography*, Springer (2009).

[3] OBINexus, "Node Zero: Setup-Free Zero-Knowledge Proofs," Technical Report (2025).

[4] N. Okpala, "The RIFT Architecture: Quantum Determinism Through Governed Computation," OBINexus (2025).