Quantum Key Distribution Simulation

Implementing BB84 and E91 Protocols with Security Analysis

Project Presentation

Ghanshyam (B22PH009)

The Cryptography Shift

🔒The Problem

Classical encryption (like RSA) relies on complex mathematical problems. However, Shor's Algorithm running on a future quantum computer could break these keys in seconds.

⚛The Solution

Quantum Key Distribution (QKD) relies on the laws of physics. According to the Heisenberg Uncertainty Principle, measuring a quantum state irreversibly disturbs it, revealing any eavesdropper.

Project Objectives

Simulate BB84

Implement the "Prepare & Measure" protocol to demonstrate secure key exchange using polarization states.

Simulate E91

Implement the Entanglement-based protocol to verify security via Bell's Inequality (CHSH).

Analyze Security

Test thresholds against "Man-in-the-Middle" attacks (Eve) and realistic fiber optic noise.

Technology Stack

Python 3.11

Core programming language used for logic and orchestration.

Qiskit

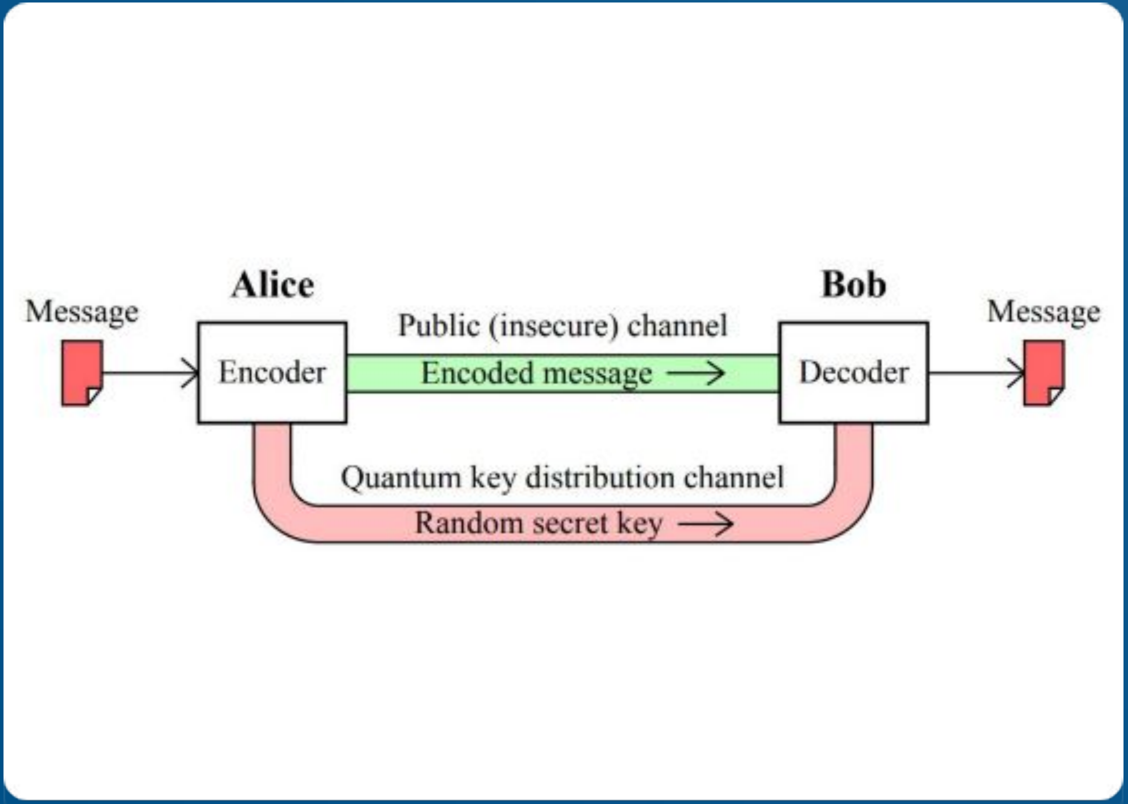
IBM's SDK for quantum circuit creation and AerSimulator execution.

Tkinter & Matplotlib

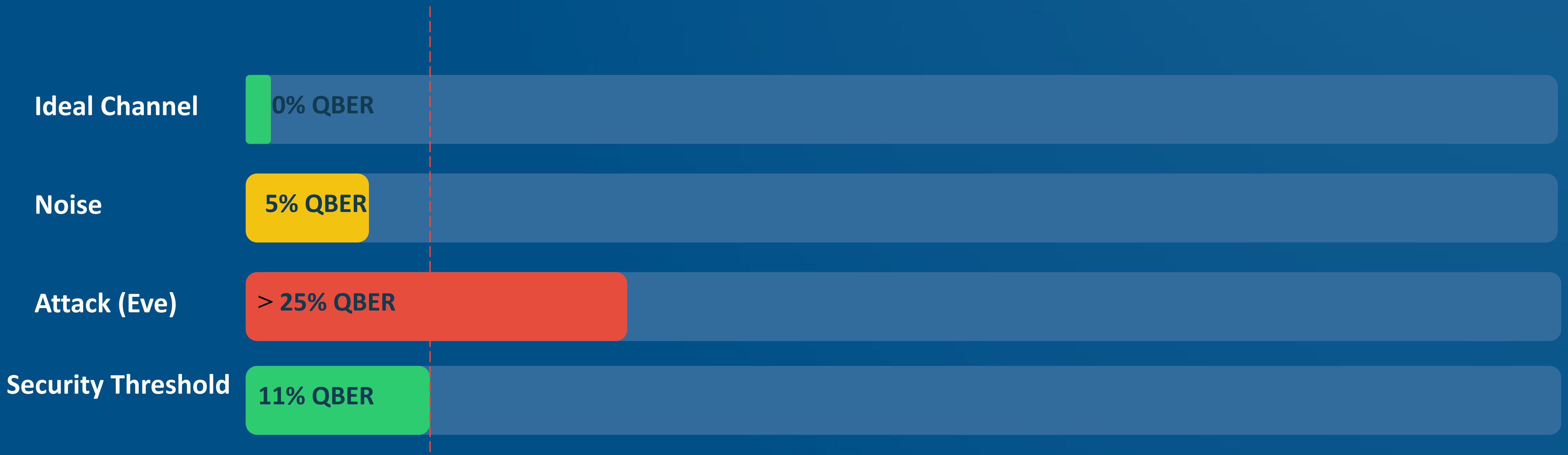
Built a custom GUI dashboard for real-time control and data visualization.

The BB84 Protocol

- **Preparation:** Alice sends photons polarized in random bases (Z or X).
- **Measurement:** Bob measures in random bases.
- **Sifting:** They compare bases (not bits). If bases match, the bit is kept.
- **Security:** If Eve intercepts, she introduces a ~25% error rate (QBER).



BB84 Simulation Results



Any error rate above 11% indicates the presence of an eavesdropper, triggering an abort protocol.

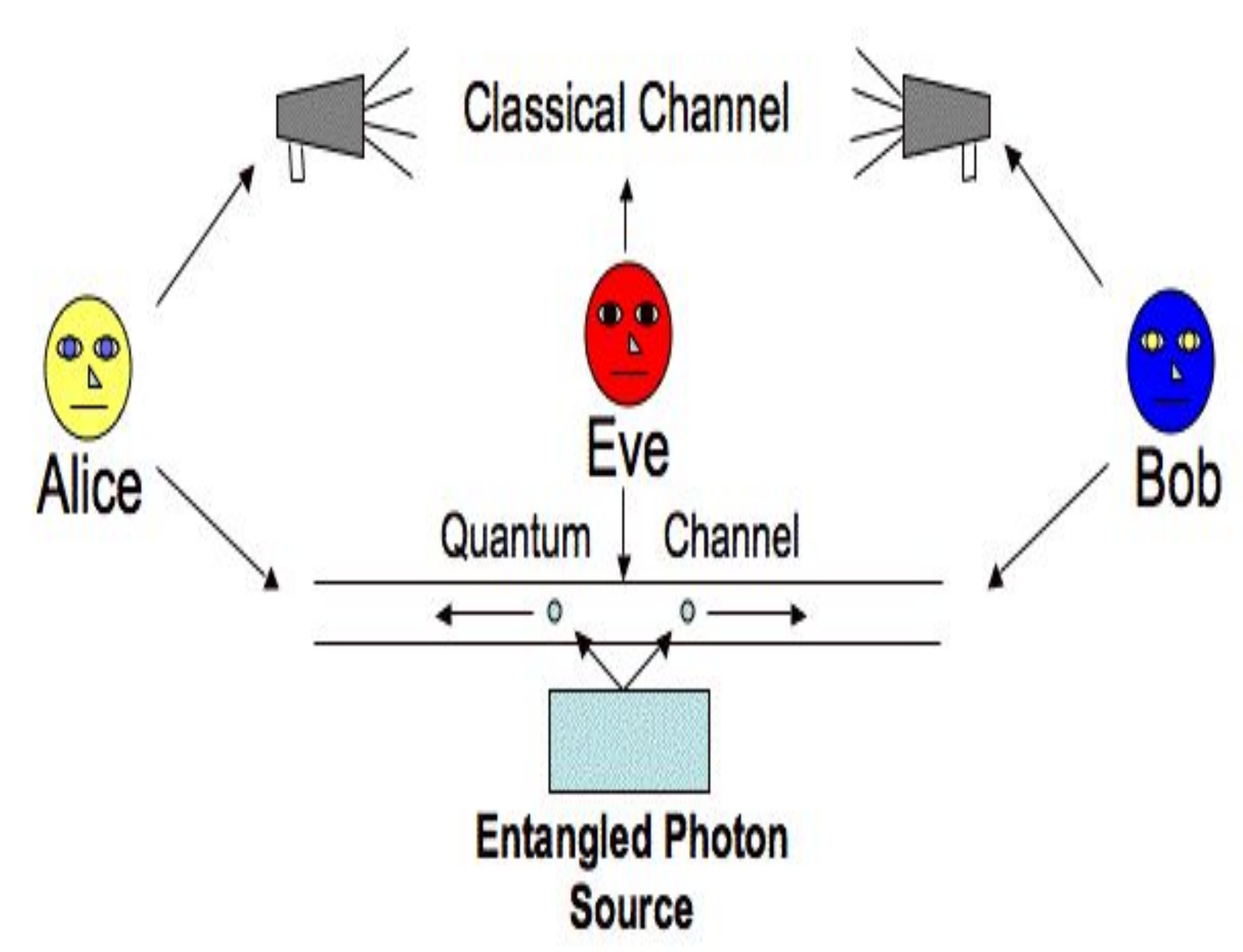
E91: Entanglement

Spooky Action

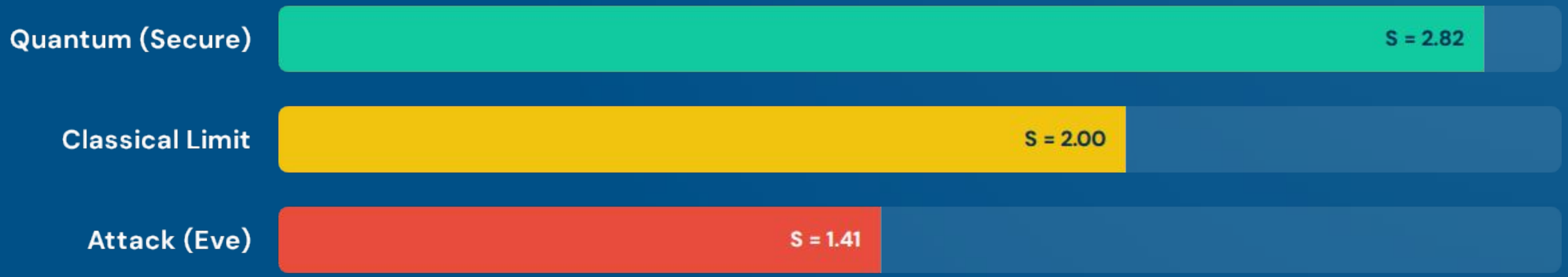
Alice and Bob do not send keys; they generate them from entangled pairs. Measuring one particle instantly determines the state of the other.

Bell's Theorem

We calculate the CHSH S-value. Classical physics is limited to 2.0. Quantum mechanics allows up to 2.82.



E91 Security Verification

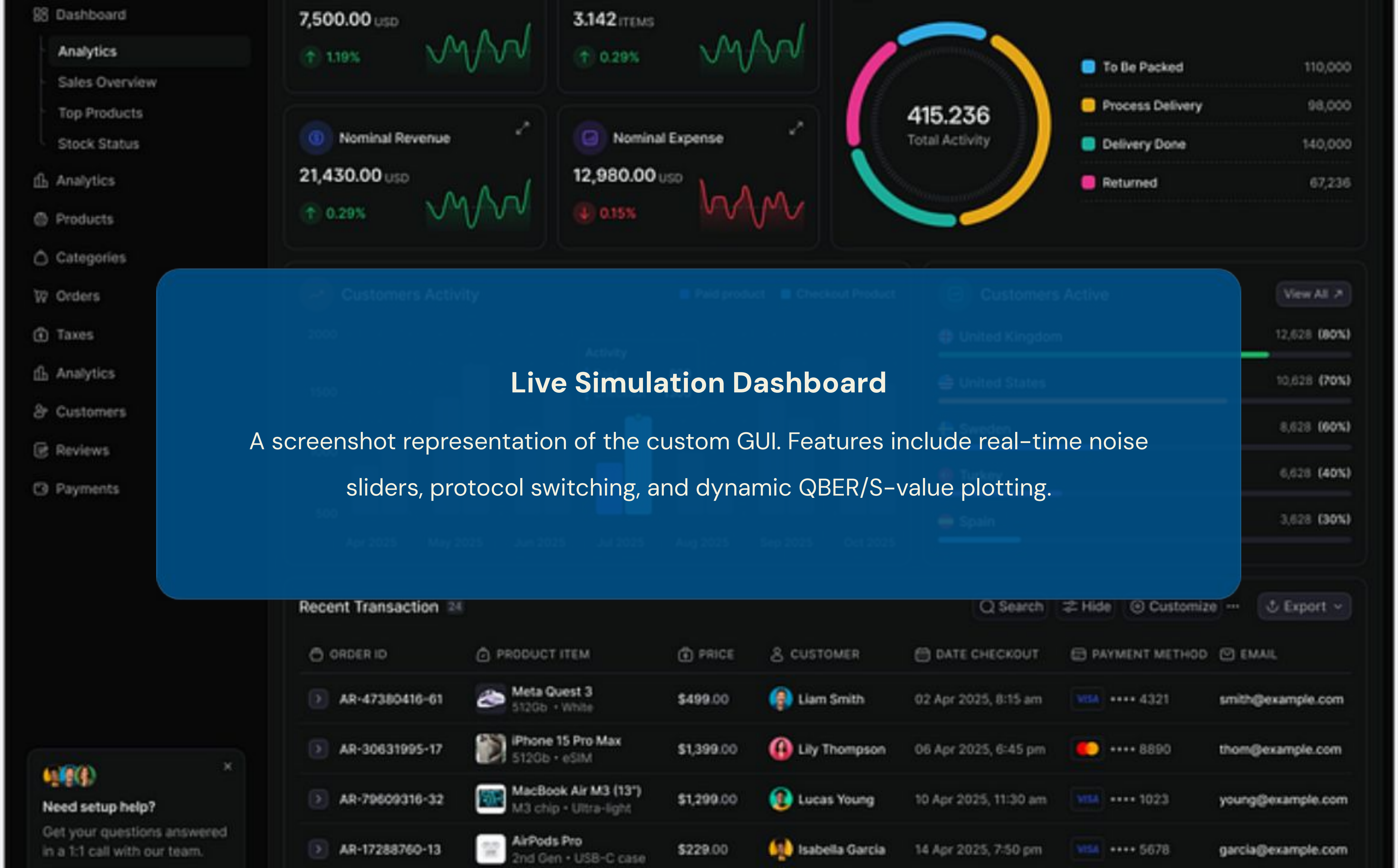


When Eve intercepts the entangled pairs, she collapses the wavefunction, dropping the S-value below 2.0, proving the channel is insecure.

Observed Experimental Outcomes

Protocol	Condition (Eve)	Measured Metric	Security Status
BB84	False (Ideal)	QBER: 0.00%	**SECURE**
BB84	True (Attack)	QBER: 26.00%	**INSECURE**
E91	False (Ideal)	S-Value: 2.842	**SECURE**
E91	True (Attack)	S-Value: 1.414	**INSECURE**

Actual data recorded from the simulation dashboard runs, confirming theoretical predictions.



Live Simulation Dashboard

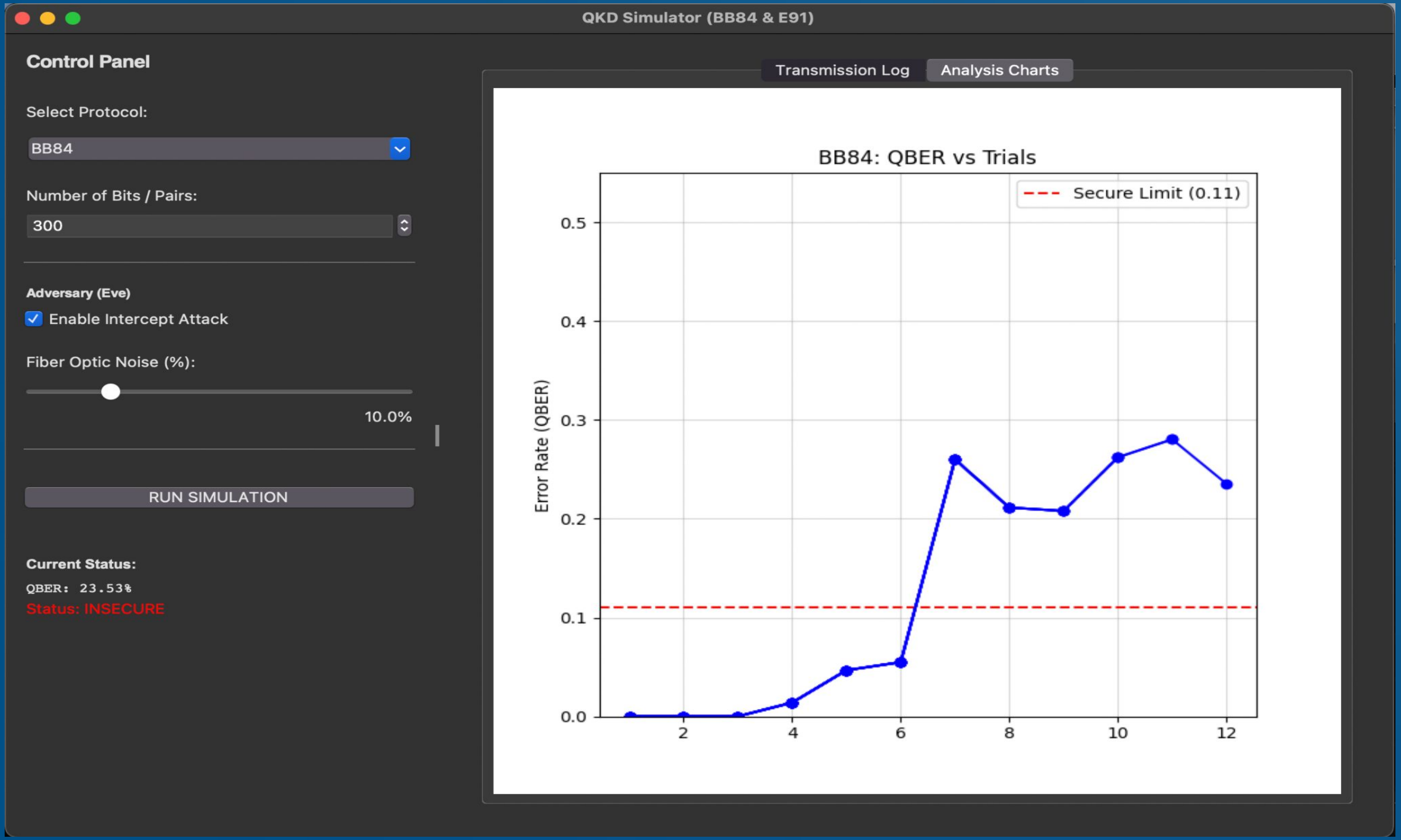
A screenshot representation of the custom GUI. Features include real-time noise sliders, protocol switching, and dynamic QBER/S-value plotting.

Live Analysis : BB84 Simulation

QBER vs Trials

The graph demonstrates the real-time calculation of Error Rate.

- **Fluctuation:** Initial trials show variance due to sample size.
- **Attack Detection:** The curve clearly rises above the red 11% threshold line when Eve is active.



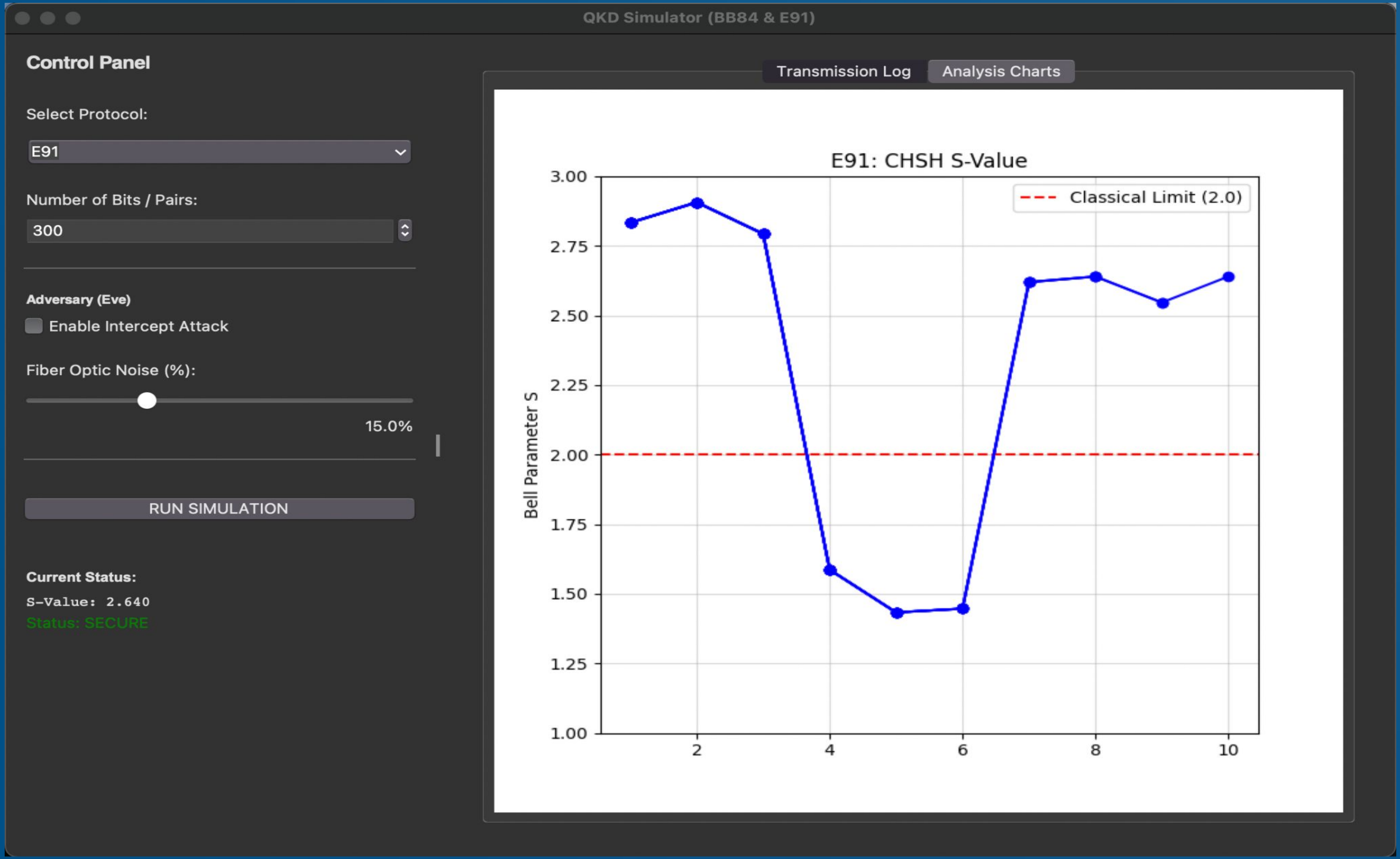

```
Running BB84 | Size: 300 | Eve: True | Noise: 0.10
------------------------------------------------------------
Sifted Key Length: 153
Error Rate (QBER): 0.2353
Alice Key: 11100001010010101110100100101010000001010...
```

Live Analysis : E91 Simulation

CHSH S-Value

Verification of Quantum Entanglement.

- **Secure:** Values approach 2.82 (Bell Violation).
- **Insecure:** Values drop below 2.0 when entanglement is destroyed by interception.




```
Running E91 | Size: 300 | Eve: False | Noise: 0.15
------------------------------------------------------------
Total Bell Pairs: 1200
Calculated S: 2.6400
Result: VIOLATION (Quantum Entanglement)
```

Technical Challenges

The Memory Bottleneck

Simulating 50+ qubits simultaneously creates a state vector of 250 complex numbers, causing a CircuitTooWideForTarget error (requires Petabytes of RAM).

The Solution: Serial Simulation

We implemented a "Serial Simulation" engine. We process qubits one-by-one (or pair-by-pair), simulating sequential transmission. This scales to 1000+ bits with minimal RAM.

Conclusion

- **Security Guaranteed:** We successfully demonstrated that security is guaranteed by physical laws, not mathematical complexity.
- **Eve Detection:** Both protocols effectively detected eavesdroppers via error spikes (BB84) or broken entanglement (E91).
- **Scalable Tool:** The developed tool is robust, scalable, and provides a clear visual demonstration of quantum cryptography concepts.

References

Quantum Cryptography: Public Key Distribution and Coin Tossing

Bennett, C. H., & Brassard, G. (1984). Proceedings of IEEE International Conference on Computers, Systems and Signal Processing.

Quantum Cryptography Based on Bell's Theorem

Ekert, A. K. (1991). Physical Review Letters, 67(6), 661–663.

Qiskit: An Open-source Framework for Quantum Computing

Qiskit contributors. (2023). IBM Quantum.

Questions?

Thank you for your attention.

https://github.com/shyam-003/bb84_with_dashboard.git