MASTER TEACHER'S GUIDE

Unit Title: Quantum Encryption (BB84) (Week 8)

This module serves as the final synthesis for the Tier 3 curriculum. It applies the principles of Superposition (Week 1), Measurement Bases (Week 2), and Noise/Disturbance (Week 5) to solve a real-world cryptographic problem. It proves that quantum mechanics provides unconditional security through the laws of physics.

Field	Detail
Target Audience	**Tier 3 - Undergraduate / Developer Level**
Design Principle	**Information Theoretic Security.** Concepts require students to simulate the entire QKD pipeline: Transmission $\rightarrow$ Sifting $\rightarrow$ Error Estimation $\rightarrow$ Reconciliation $\rightarrow$ Privacy Amplification.
Learning Progression	**BB84 Protocol (Bases) → No-Cloning Theorem (Security Proof) → Sifting & QBER → Classical Post-Processing.**
Duration	**1 Week** (approx. 4×60-90 minute sessions)
Teacher Guidance	Proficiency in probability theory (conditional probabilities) is essential. Emphasize that "Encryption" is not the quantum part; **Key Distribution** is. The quantum link merely generates the shared secret.

---

2. Pedagogical Framework: The Security Engine

This unit uses **Quantum Measurement** to create security and **Classical Algorithms** to distill it. The goal is to move students from "quantum circuits" to "quantum protocols."

Focus Area	Objective (The student will be able to...)	Bloom's Level
Science/Literacy	Explain why the **No-Cloning Theorem** prevents Eve from copying the key and how **Measurement Disturbance** reveals her presence (25% error rate).	**Understanding, Analyzing**
Mathematics	Calculate the expected **Quantum Bit Error Rate (QBER)** for an intercept-resend attack. Compute the final secure key rate formula $R \approx 1 - H_1(e) - H_2(e)$.	**Applying, Evaluating**

Computational Logic	Implement the full BB84 simulation in Python/Qiskit, including the classical "Sifting" handshake and "Error Reconciliation" (Cascade-like logic).	**Applying**, **Creating**

---

3. Computational Logic Refinements (Week 8)

A. The Protocol: Conjugate Bases

Concept	Explanation
Rectilinear (Z)	The standard computational basis. Used for bits 0 and 1.
Diagonal (X)	The superposition basis. Created by Hadamard gates.
Unbiasedness	Measuring a Z-state in the X-basis (or vice-versa) yields a random result. This is the core security feature.

B. Security: Detectable Disturbance

Concept	Explanation	Mathematical Description
Eve's Dilemma	Eve must measure to learn the key, but measuring in the wrong basis destroys the state.	$P(\text{Eve wrong basis}) = 0.5$
QBER	The error rate introduced by Eve. If she attacks all bits, she induces a 25% error in the sifted key.	$QBER = 0.5 \times 0.5 = 25\%$

C. Post-Processing: Distilling the Secret

Concept	Explanation	Mathematical Description
Sifting	Discarding bits where Alice and Bob used different bases.	if basis_A == basis_B: keep
Reconciliation	Fixing errors using parity checks (e.g., Cascade protocol). Leaks some info to Eve.	Binary Search for errors.
Privacy Amp.	Shrinking the key to remove Eve's partial knowledge.	$K_{final} = Hash(K_{corrected})$

---

4. Exemplary Lesson Plan: The Unbreakable Code

Module: Simulating QKD This lesson focuses on the software implementation of the protocol, simulating the actions of Alice, Bob, and Eve.

Coding Lab: BB84 Simulation (Tier3W8coding.ipynb)

Objective	**Students will write a Python program that simulates the transmission of qubits, the interception by Eve, and the classical post-processing steps to derive a shared secret key.**
Required Resources	Python Environment (Jupyter), Tier3W8coding.ipynb, Tier3W8draft.ipynb (Lecture Notes)

Step-by-Step Instructions

Part 1: The Math (Lecture Notes)

1. **Probability Tree:** Draw the tree for an intercepted bit. Alice sends $|0\rangle \rightarrow$ Eve measures X ($|+\rangle$) $\rightarrow$ Bob measures Z (0 or 1). Show the 25% error path.
2. **Basis Matching:** Prove mathematically why sifting is necessary (50% of raw data is noise).

Part 2: The Code (Qiskit Implementation)

1. **Task 1 (Alice):** Generate random bitstrings for data and bases. Encode into circuits (I, X, H, XH).
2. **Task 2 (Transmission):**
    - **No Eve:** Bob measures using random bases. Implement **Sifting**: raw_key = [b for a, b, m in zip(a_bases, b_bases, results) if a==b].
    - **With Eve:** Eve measures (random bases) and re-encodes. Bob measures Eve's qubits.
3. **Task 3 (Analysis):** Calculate the **QBER** by comparing Alice's and Bob's sifted keys. If $QBER \approx 0$, key is safe. If $QBER \approx 25\%$, drop the key.
4. **Task 4 (Privacy):** Implement a simple XOR hash to reduce the key length and amplify privacy.

Part 3: Assessment

- **Quiz Question 3:** What happens if Bob measures $|0\rangle$ in the X-basis? (Answer: Random 0 or 1).
- **Quiz Question 7:** What is the theoretical QBER if Eve intercepts every bit? (Answer: 25%).

- **Quiz Question 9:** What step removes Eve's partial information? (Answer: Privacy Amplification).

---

5. Resources for Curriculum Implementation (Week 8)

Resource Name	Type	Purpose in Curriculum
Tier3W8draft	**Lecture Notes (IPYNB)**	Formal derivation of the BB84 states, the intercept-resend attack probabilities, and the post-processing pipeline.
Tier3W8coding	**Lab Notebook (IPYNB)**	Coding tasks to simulate the full multi-agent protocol (Alice, Bob, Eve).
TierW8	**Quiz (IPYNB)**	**Knowledge Check:** 10 multiple-choice questions covering the physics of QKD and the logic of the security proof.

---

6. Conclusion and Next Steps

This **Tier 3, Week 8** module concludes the Tier 3 curriculum. Students have now journeyed from the basic linear algebra of a single qubit to the implementation of secure quantum communication protocols.

Key Takeaway: Quantum security relies on the **laws of physics** (No-Cloning, Uncertainty Principle), not on computational hardness.

Final Review: Students are now prepared to move to **Tier 4 (Advanced)**, where they will explore Quantum Error Correction and other more advanced algorithms like Shor's Algorithm.