ASTHA YADAV

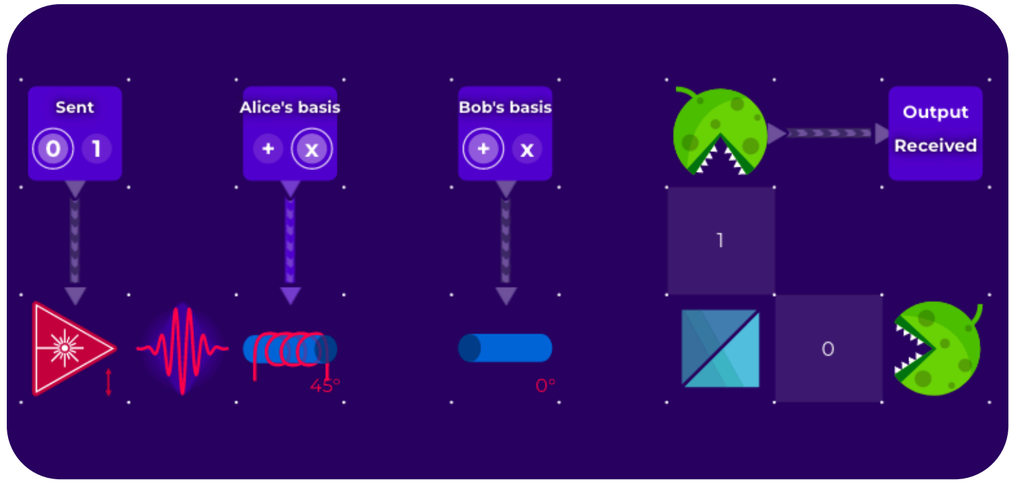
astha04.yadav@gmail.com

BB84

PROTOCOL

What is QKD & Why “BB84”

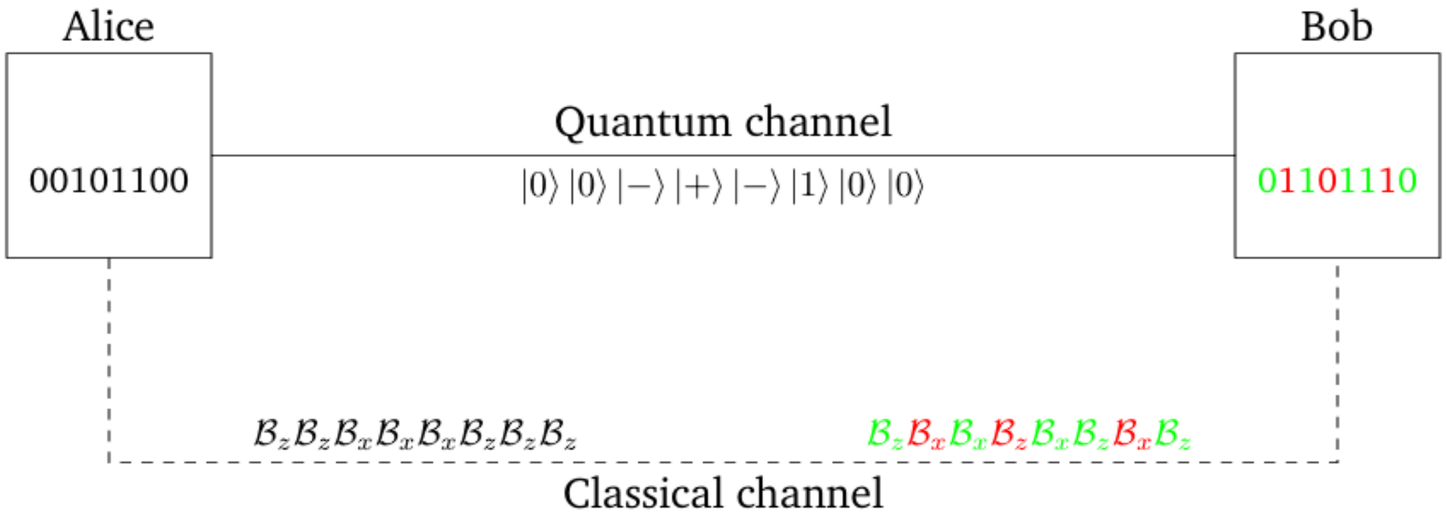
There are three things integral parts of Cryptography: Symmetric Cryptography, Asymmetric Cryptography (Public Key Cryptography), and Digital Signatures using hash functions. Cryptography works by using a pair of functions: **E** (encryption) and **D** (Decryption), and keys pace **K**, from which we get our key “**k**”. A message “**m**” is encrypted using the E function and decrypted using the D function, using the key k. Now this “**key**” is central to encryption. The whole reason the attacker can’t know the message is because they don’t know the key. Only by feeding the key into D will they get the message.
Now in **Symmetric Encryption**, both parties use same key. But how exactly will you safely generate and share this key? Only then can you use that to encrypt your messages. This is where **Quantum Key Distribution** comes in. It is a good method of key generation and communication but needs more work still. It uses the no-cloning theorem as it’s base, that’s why “quantum”.



BB84 stands for Bennett and Brassard, and the fact that they made this protocol - the first quantum cryptography protocol - in 1984.

It's the blueprint for how to use quantum states to guarantee a secure key exchange

How does it work??



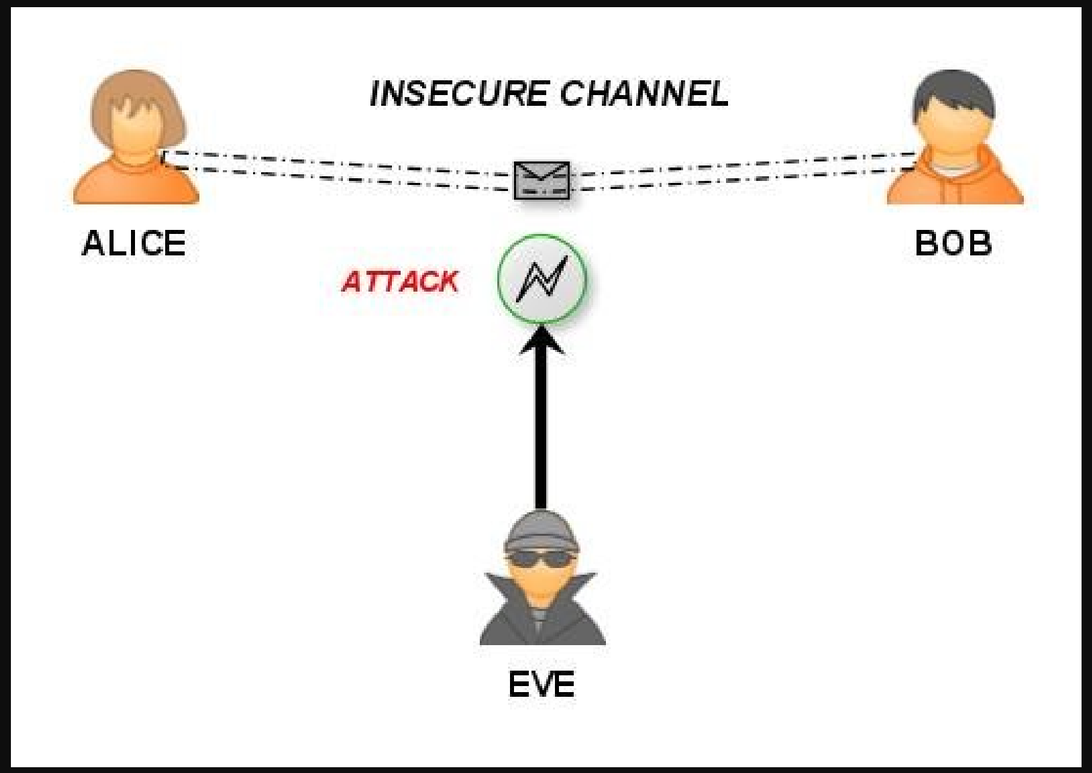
The whole process relies on two key things happening randomly, and then making one simple comparison.
First, Alice has to encode her key bits, |0>s and |1>s. For every bit, she randomly chooses one of two measurement frames, or bases: the straight Z-basis or the diagonal X-basis. For example, she'll encode a |0> either as |0> (Z-basis) or |+> (X-basis). She sends these encoded photons—her quantum key—to Bob.
Second, Bob doesn't know Alice's choice, so he randomly guesses a measurement basis for every photon he receives.

This is the crucial part: if Bob guesses the wrong basis, the quantum uncertainty principle says his measurement will randomly scramble the photon’s original state.

Finally, after all the photons are sent, Alice and Bob get on a public, classical channel and compare their bases.[2] They throw away all the bits where their bases didn't match. The bits they keep—the "sifted key"—should be identical. If that sifted key is clean, they have their secret. If it's noisy, they know someone was listening.

n	A's bit	A's basis	A's qubit	B's basis	B's qubit	B's bit	Key
1	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_z$	$\|0\rangle$	0	0
2	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_x$	$\|+\rangle$ / $\|-\rangle$	0/1	
3	1	$\mathcal{B}_x$	$\|-\rangle$	$\mathcal{B}_x$	$\|-\rangle$	1	1
4	0	$\mathcal{B}_x$	$\|+\rangle$	$\mathcal{B}_z$	$\|0\rangle$ / $\|1\rangle$	0/1	
5	1	$\mathcal{B}_x$	$\|-\rangle$	$\mathcal{B}_x$	$\|-\rangle$	1	1
6	1	$\mathcal{B}_z$	$\|1\rangle$	$\mathcal{B}_z$	$\|1\rangle$	1	1
7	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_x$	$\|+\rangle$ / $\|-\rangle$	0/1	
8	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_z$	$\|0\rangle$	0	0

Noise/Eavesdropper



The most important feature of QKD is that it makes eavesdropping detectable. Let's call our eavesdropper "Eve". Eve can't copy a quantum state without disturbing it (No-Cloning Theorem). If she intercepts a photon, she has to guess the measurement basis, just like Bob.

Since Alice and Bob's bases only match 50% of the time, Eve will inevitably choose the wrong basis about 50% of the time. When she measures in the wrong basis, she scrambles the bit Alice sent. This disturbance leads to errors in the final key Bob measures.

n	A's bit	A's basis	A's qubit	E's basis	E's qubit	B's basis	B's qubit	B's bit	Key
1	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_z$	$\|0\rangle$	0	0
2	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_x$	$\|-\rangle$	1	
3	1	$\mathcal{B}_x$	$\|-\rangle$	$\mathcal{B}_z$	$\|1\rangle$	$\mathcal{B}_x$	$\|+\rangle$	0	0
4	0	$\mathcal{B}_x$	$\|+\rangle$	$\mathcal{B}_x$	$\|+\rangle$	$\mathcal{B}_z$	$\|1\rangle$	1	
5	1	$\mathcal{B}_x$	$\|-\rangle$	$\mathcal{B}_x$	$\|-\rangle$	$\mathcal{B}_x$	$\|-\rangle$	1	1
6	1	$\mathcal{B}_z$	$\|1\rangle$	$\mathcal{B}_z$	$\|1\rangle$	$\mathcal{B}_z$	$\|1\rangle$	1	1
7	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_x$	$\|-\rangle$	1	
8	0	$\mathcal{B}_z$	$\|0\rangle$	$\mathcal{B}_x$	$\|+\rangle$	$\mathcal{B}_z$	$\|0\rangle$	0	0

To detect this, Alice and Bob publicly reveal a small sample of their sifted key. They check for mismatches. The rate of error is called the Quantum Bit Error Rate (QBER). If the calculated QBER is above a safe threshold (say, 10%), they know someone was listening—Eve's interference created too much noise. At that point, they simply abort the entire key exchange and try again later. They never use the corrupted key.

ANALYZING RESULTS

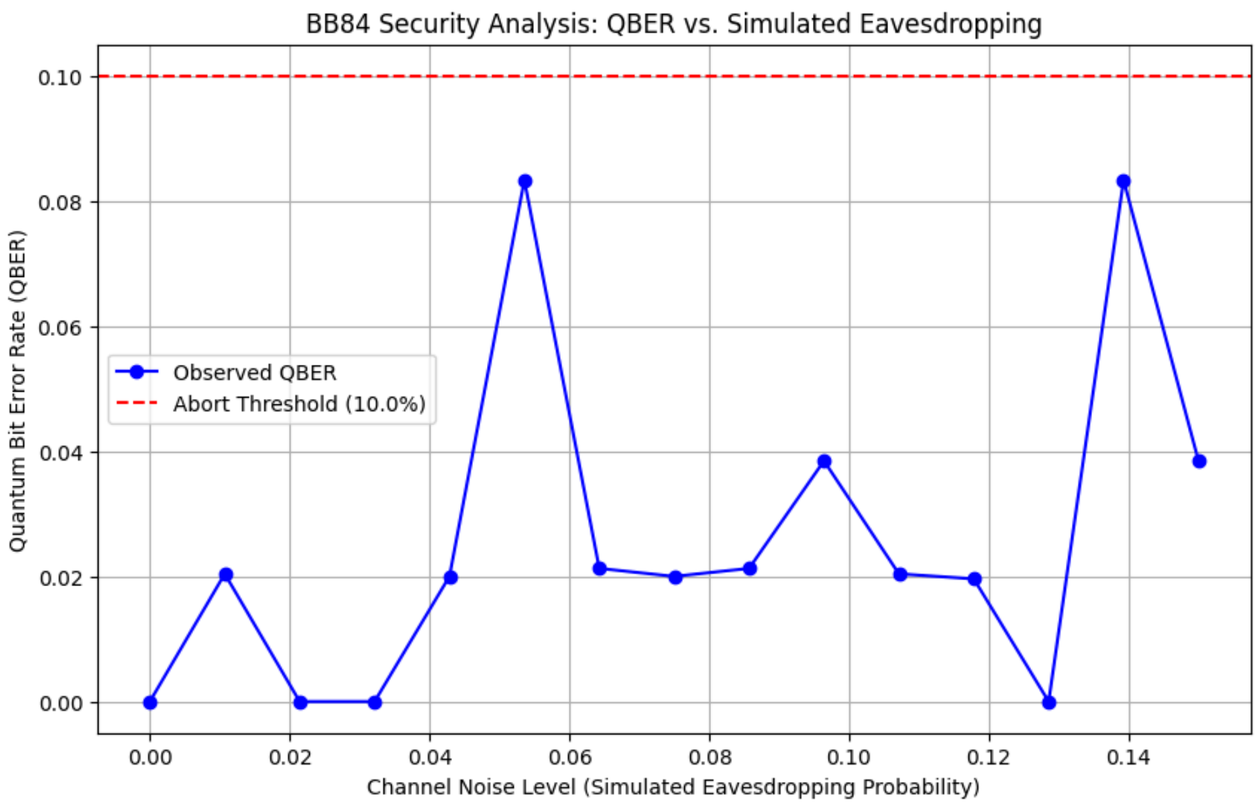
This graph plots the Observed QBER (the blue line) against the increasing level of Simulated Eavesdropping (the noise probability we built into the channel).

1. If the QBER we calculate is below this line, we conclude the key is safe and can proceed.
2. If the QBER crosses above this line, it means Eve has disturbed the quantum states too much, and we abort the entire exchange.

What Our Data Shows:
No exchange was aborted across the entire range of simulated noise (up to 15%).

- The highest QBER we measured was approximately 8.3% (at noise levels 0.054 and 0.139.
- Crucially, even at the highest simulated noise level of 15.0% the resulting QBER was only 3.8%

Conclusion on Eavesdropping
A lower threshold would have successfully detected the spikes at 0.054 and 0.139. This highlights the importance of choosing a realistic and sensitive abort threshold in real-world QKD systems.



COMPARISON TO PQC ALGORITHM

We've shown that BB84 can generate a secret key and detect Eve. But the challenge also asks us to compare it to Post-Quantum Cryptography (PQC), which is a very different beast. They both aim for security in a quantum world, but they work on opposite principles.

The PQC Approach
A typical PQC candidate, like CRYSTALS-Kyber, is based on Lattice Cryptography. Its security doesn't rely on physics; it relies on hard math. PQC uses complex mathematical problems that are considered impossible for even a powerful quantum computer (running an algorithm like Shor's) to solve in a reasonable amount of time. This is computationally secure.

Feature	BB84 QKD	CRYSTALS-Kyber (PQC)
Security Basis	Physics (No-Cloning Theorem).	Computational Hardness (Lattice Math).
Resilience Guarantee	Information-Theoretically Secure (ITS). It's provably unbreakable, regardless of future computing power, because any attack is *detected*.	Computationally Secure. It's believed to be secure until a breakthrough algorithm or computer proves otherwise.
Use Case	Key Distribution only.	Key Encapsulation and Encryption.
Hardware	Requires Specialized Quantum Hardware (photon sources/detectors).	Runs on Standard Classical Computers and existing infrastructure.

Thank You

- Astha Yadav
- astha04.yadav@gmail.com