QISKIT FALL FEST

HACKATHON

TEAM GOTOSHDINO

WHY QKD?

The conventional cryptosystems used for data-encryption rely on the complexity of mathematical algorithms.

Quantum Computers can easily solve these complex mathematical problems, henceforth, making it easy for anyone with access to the technology to decrypt the data.

Once a powerful enough quantum computer exists, Shor’s quantum algorithm for factoring and Grover’s quantum algorithm for search can be used to rapidly factor numbers and speed up searches.

HOW DOES QKD WORK?

Quantum Key Distribution (QKD) leverages the properties of quantum mechanics to securely derive a symmetric encryption key at two locations.

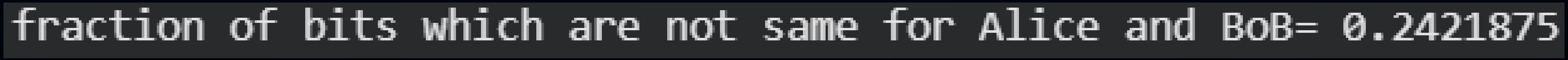
The provable security for QKD relies on quantum mechanical properties which allow detection and prevent successful eavesdropping.

Quantum objects exist in a state of superposition where the value for a property of the object can be described as a set of probabilities for different values.

Observation of the quantum object perturbs it in manner which leads it to collapse into a single measurable value.

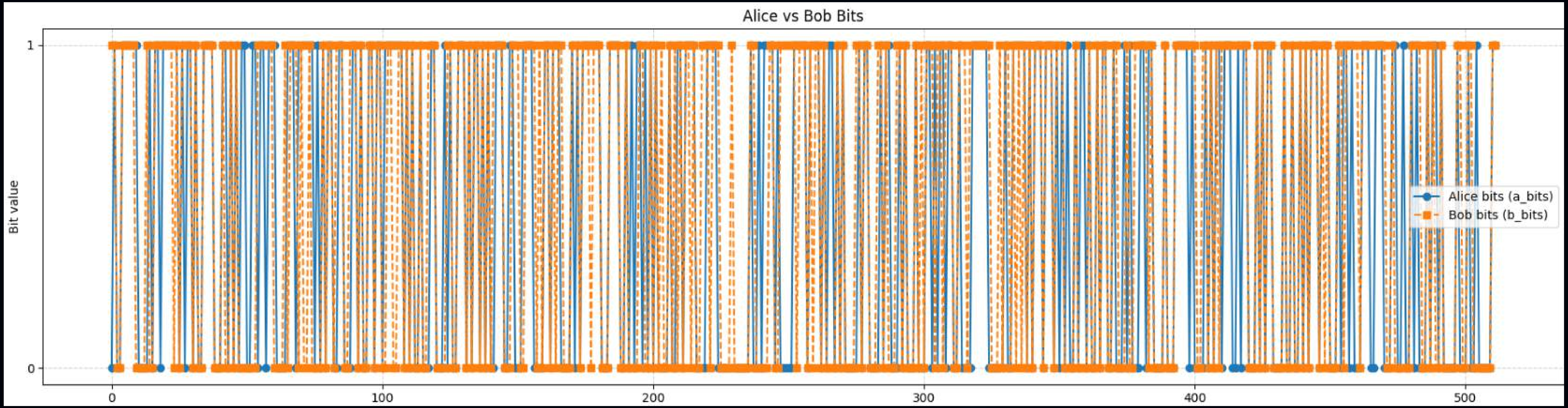
IMPLEMENTING BB84 PROTOCOL

We generated a quantum circuit with 1 classical bit and 1 quantum bit. We used random number generators to chose what bit alice wants to transmit and what base alice uses to encode this bit. We send this through a quantum channel and bob receives it using A random base again (using a random number generator again).



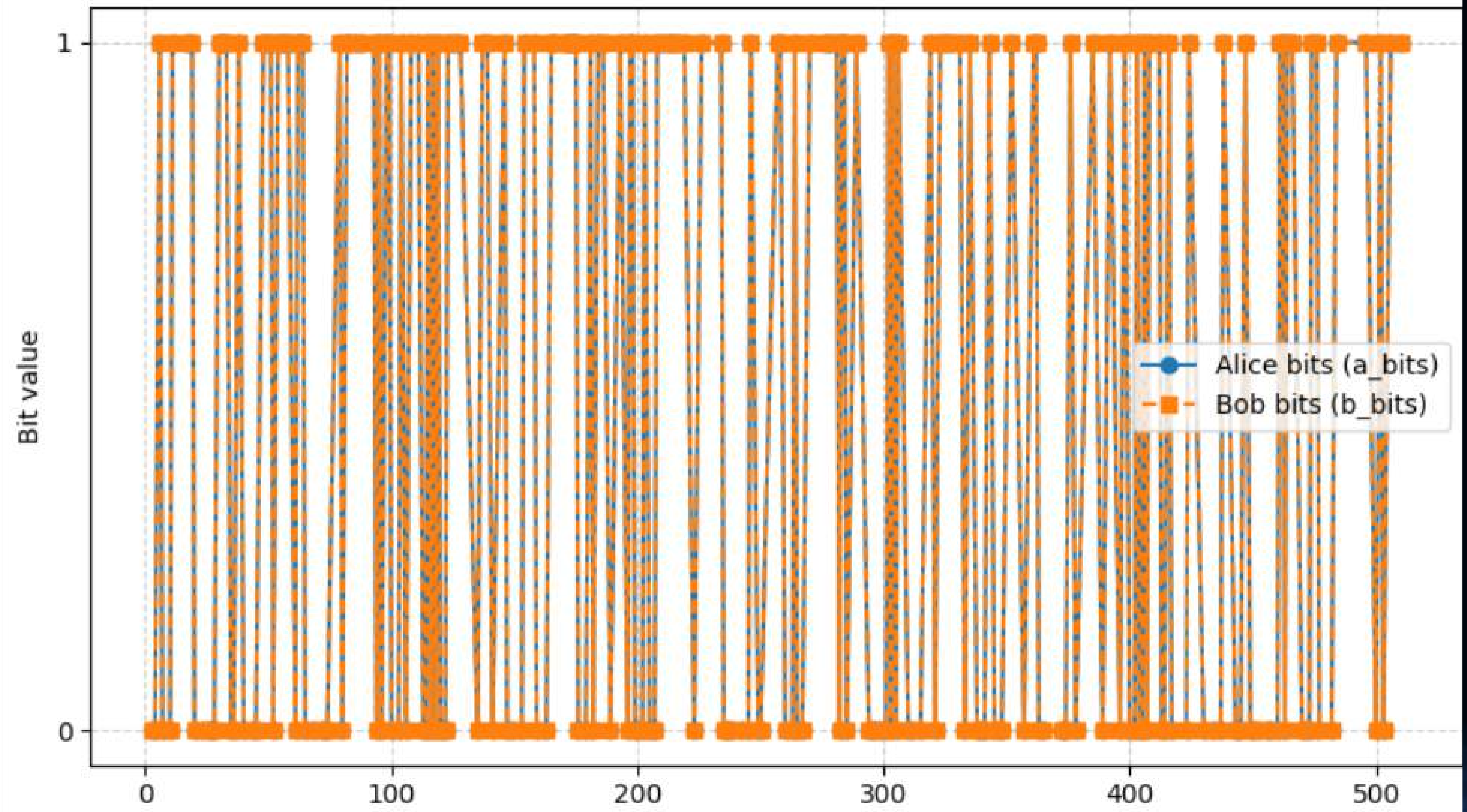
```
fraction of bits which are not same for Alice and BoB= 0.2421875
```

Since the basis used by Alice to send the Qubit might be different from the basis used by Bob to measure the Qubit (as both are decided by different RNGs), some of the bits received by Bob are different from what Alice sent by a probability of ¼ and our model correctly obtains fraction of incorrect obtained bits close to the theoretical probability.



The above graph compares the bits sent by alice to the bits received by alice. We can clearly observe that the plot for alice and bob's bits does not coincide.

Now, Bob and Alice will share through the public channel what basis of decryption and encryption they used respectively. Note that they are not sharing the bits - they are only sharing the basis used. Both of them will drop the bits for which they used different basis. We, henceforth, obtain a filtered key and the basis of decryption and encryption match for all bits.



The above graph compares the bits sent by Alice and bits received by Bob after filtering. We can see many bits were dropped and the plot for Alice and Bob's bits coincides.

This was all done till now using an ideal simulator. In the case of a non-ideal simulator, there will be noise introduced in the transmission process through the quantum channel. We accounted for four types of noises:

- Gate Noise (Depolarisation)
- Measurement Noise (Bit Flip)
- Amplitude Decay Effect (Loss of Photon)
- Phase Damping Noise

Hence, the key sent by Alice and the key received by Bob will still not be same even after filtering as, even though Bob used the correct basis to measure the Qubit, due to noise during transmission, the bit sent by Alice might get modified.

This results in mismatch of bits sent and received even though the basis used are same for Bob and Alice.

QBER

QBER (Quantum Bit Error Rate) is the fraction of bits in the sifted key where Alice’s and Bob’s bits disagree.
It directly quantifies noise/eavesdropping in the quantum channel.

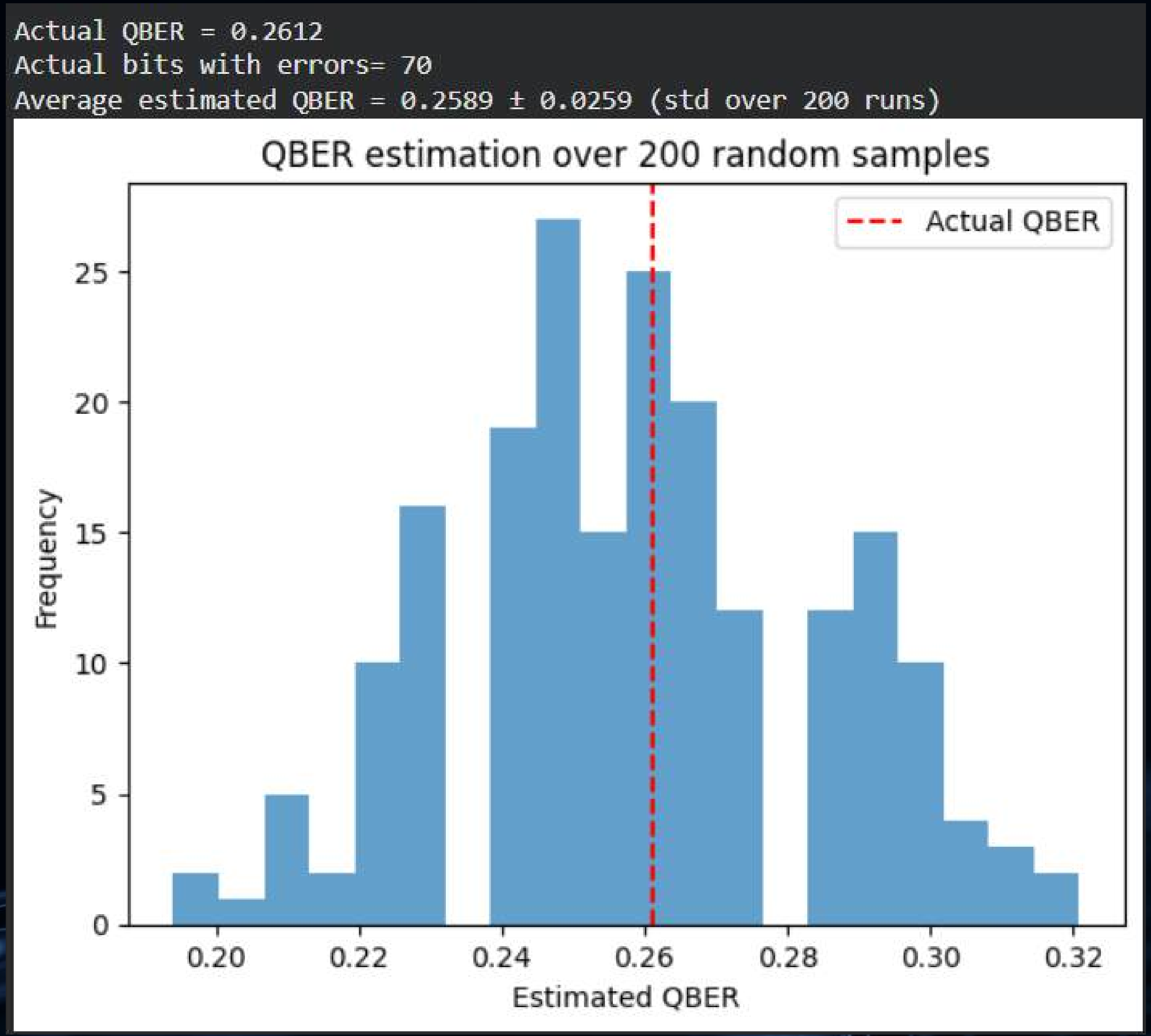
The procedure to calculate QBER involves Alice and Bob choosing a subset of their sifted key and disclosing it to each other using the classical channel.
The number of different bits in the choosen subset divided by the length of the subset gives us the QBER value.
We would observe that the QBER value in ideal case (without noise) will be equal to 0 since there are no bit mismatches.

Once they succesfully calculate the QBER value, they will drop the bits from subset chosen from the sifted key as it was revealed to everyone using the classical channel and is no longer private.

QBER ANALYSIS

In the code, a random subset of the key is selected to check the QBER and then that subset is removed from the Key

In order to prove that the QBER from a random subset is close to actual QBER in entire key our code, we calculated QBER for multiple random subsets of the key and plotted the distribution.

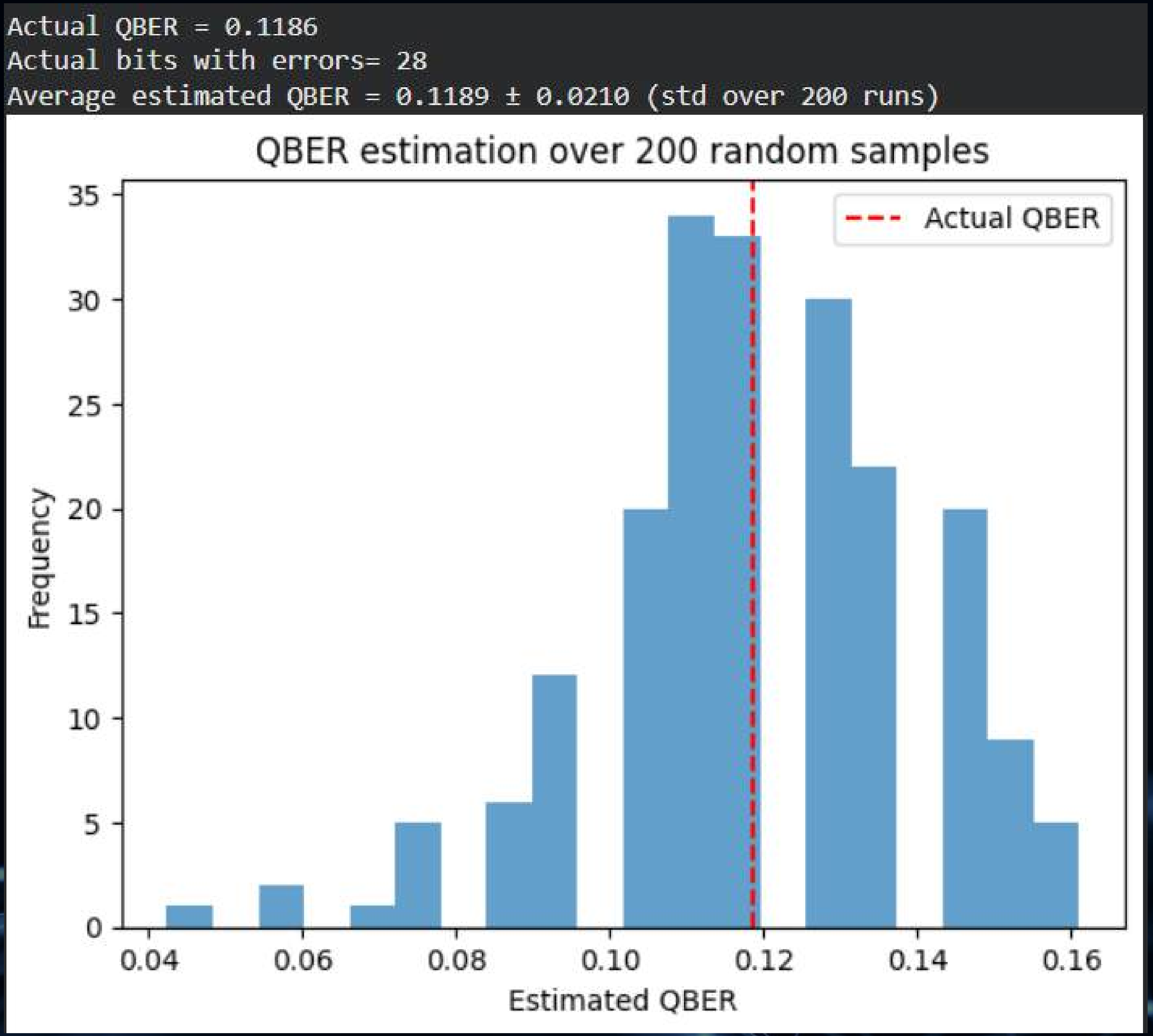


```
Actual QBER = 0.2612
Actual bits with errors= 70
Average estimated QBER = 0.2589 ± 0.0259 (std over 200 runs)
```

We can clearly observe that actual QBER is close in value to the average estimated QBER that was calculated by iterating the random subset generation and QBER calculating process 200 times.

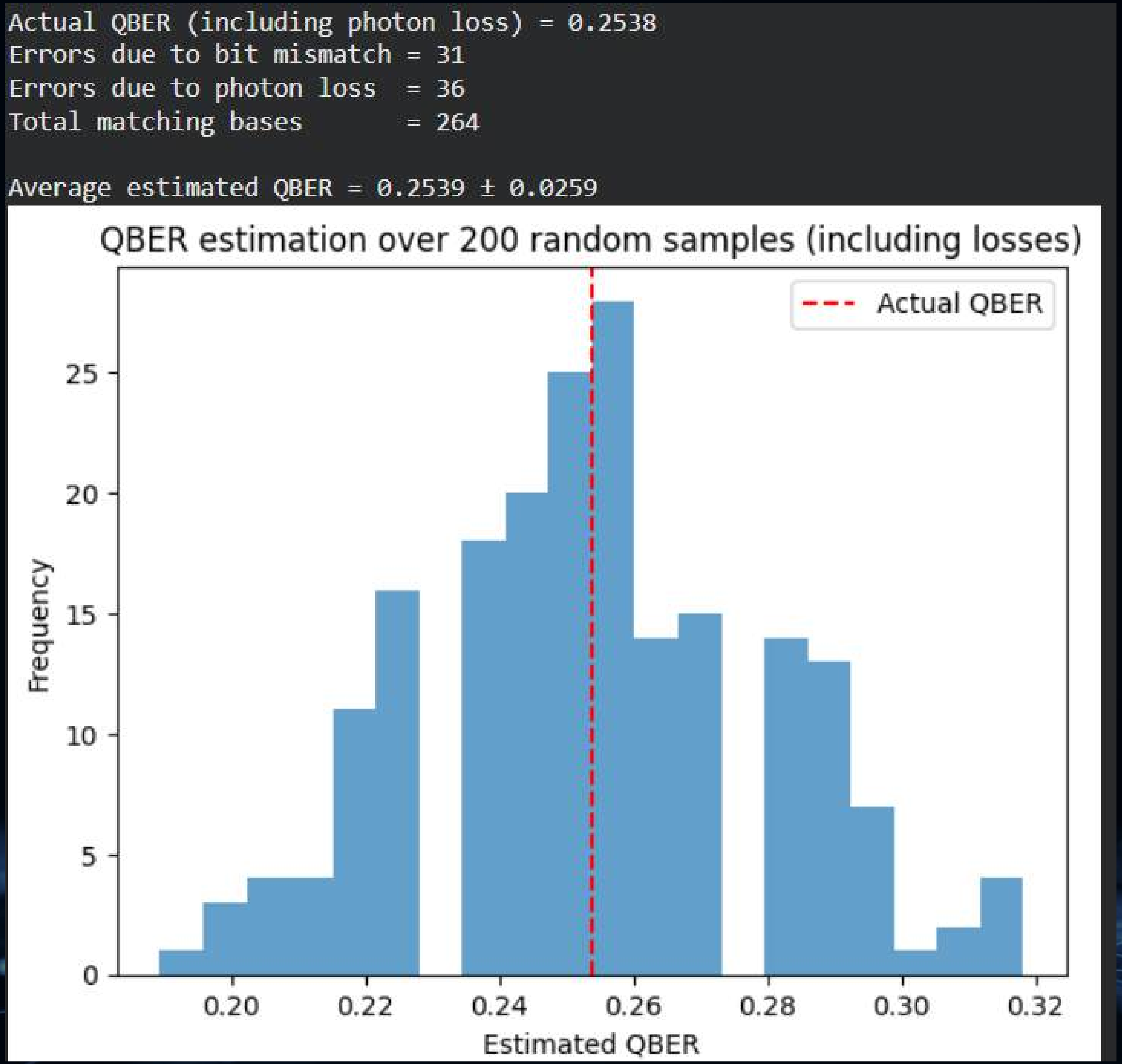
Also, to show that QBER is an indicator of noise/errors in the key, our code calculated the QBER for a system with and without Amplitude Decat Effect (Photon Loss). We observed that the QBER for system having Photon Loss is higher than the QBER for system not having Photon Loss.

This proves that QBER is indeed the measure of noise/error in the key.



```
Actual QBER = 0.1186
Actual bits with errors= 28
Average estimated QBER = 0.1189 ± 0.0210 (std over 200 runs)
```

QBER of System without Amplitude Decay Effect (Loss of Photon)



```
Actual QBER (including photon loss) = 0.2538
Errors due to bit mismatch = 31
Errors due to photon loss  = 36
Total matching bases       = 264

Average estimated QBER = 0.2539 ± 0.0259
```

QBER of a system with Amplitude Decay Effect (Loss of Photon)

Left graph has lesser QBER than right graph.

Now after finally being done by calculating QBER, we discard the bits disclosed to do so. Now we move on to our next step of Error Reconciliation.

ERROR RECONCILIATION

Objective of Error Reconciliation is to exchange minimum classical information to correct bit mismatches between Alice’s and Bob’s keys without revealing the key itself to an eavesdropper.

We have achieved this using Cascade Protocol in our code.

CASCADE PROTOCOL

Alice and Bob both divide their keys into blocks.

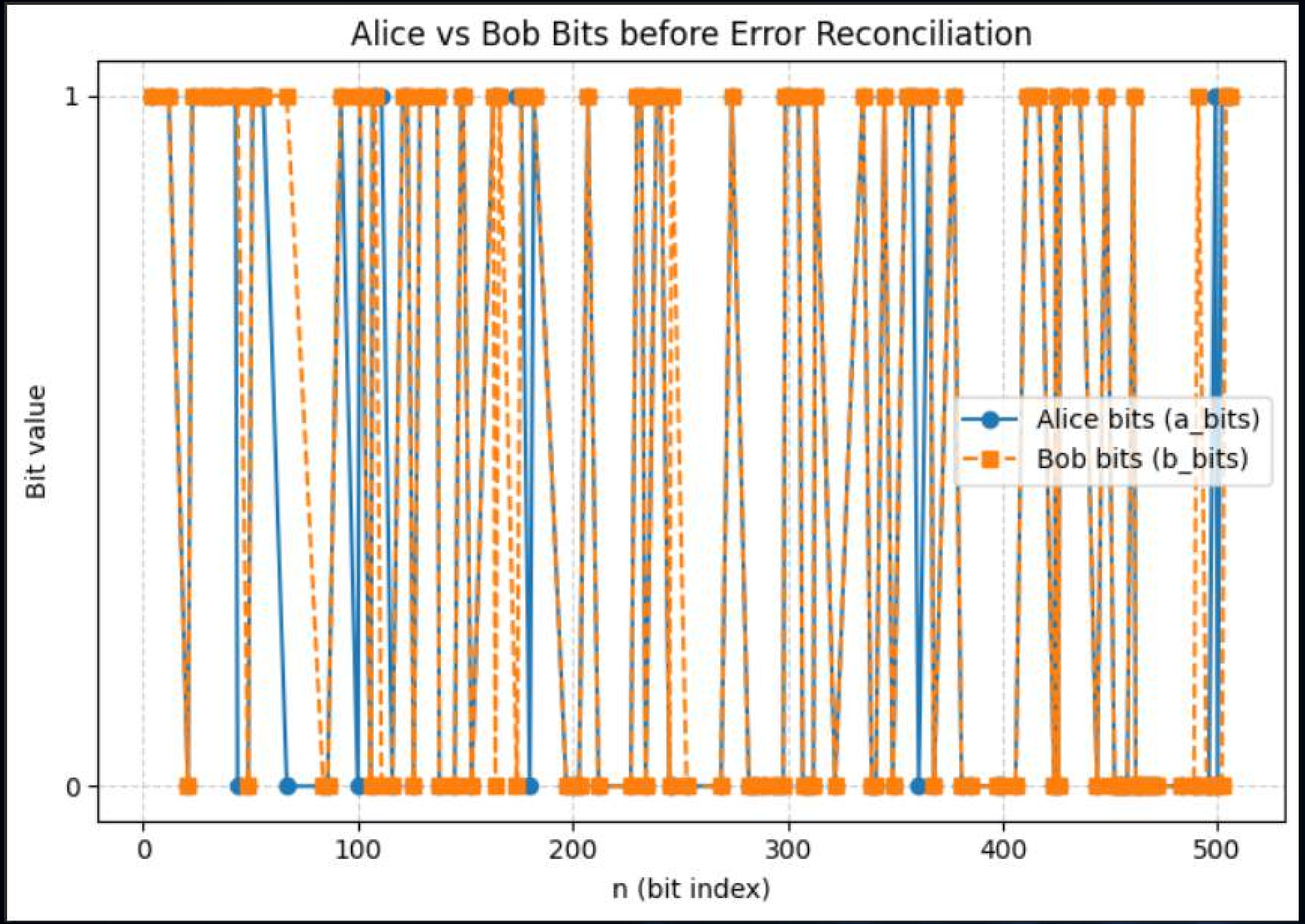
They publicly compare the parity (even/odd) of each block.

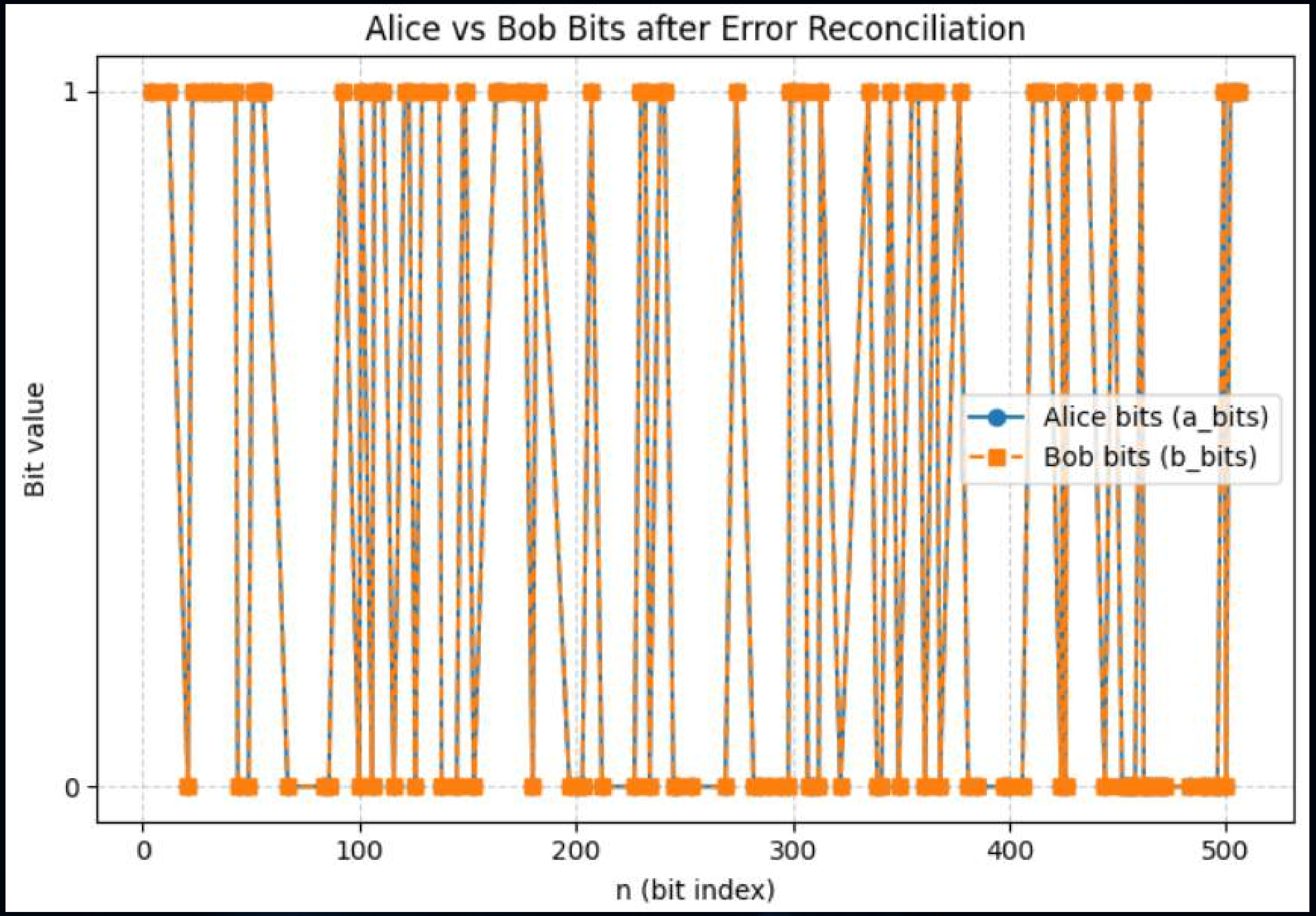
A mismatch in parity means an odd number of errors in that block.

They use binary search to locate and flip the erroneous bits.

Then, in subsequent passes, they reshuffle their keys randomly to catch any errors that affected parities indirectly.(both shuffle their keys in the same way and the way of permuting is made public through the classical channel)

Once we are done with correcting all the mismatching bits, we return back to our original permutation of bits and the key thus obtained is our FINAL QUANTUM KEY and is exactly same for both Alice and Bob.





Alice and Bob's bits before Error Reconciliation. We can see some bits mismatch due to noise.

After Error Reconciliation, all the mismatching bits now match.

Hence, we finally, succesfully implemented BB84 protocol -:

1. We showed how finding QBER value for a random subset of the key gives us the correct measure of errors in key.
2. We also compared QBER values for systems with and without Amplitude Effect. Our observations were in compliance with the fact that QBER should increase with more noise effects.

3. We performed Error Reconciliation using Cascade Protocol and made the mismatching bits identical to what Alice intended to send.

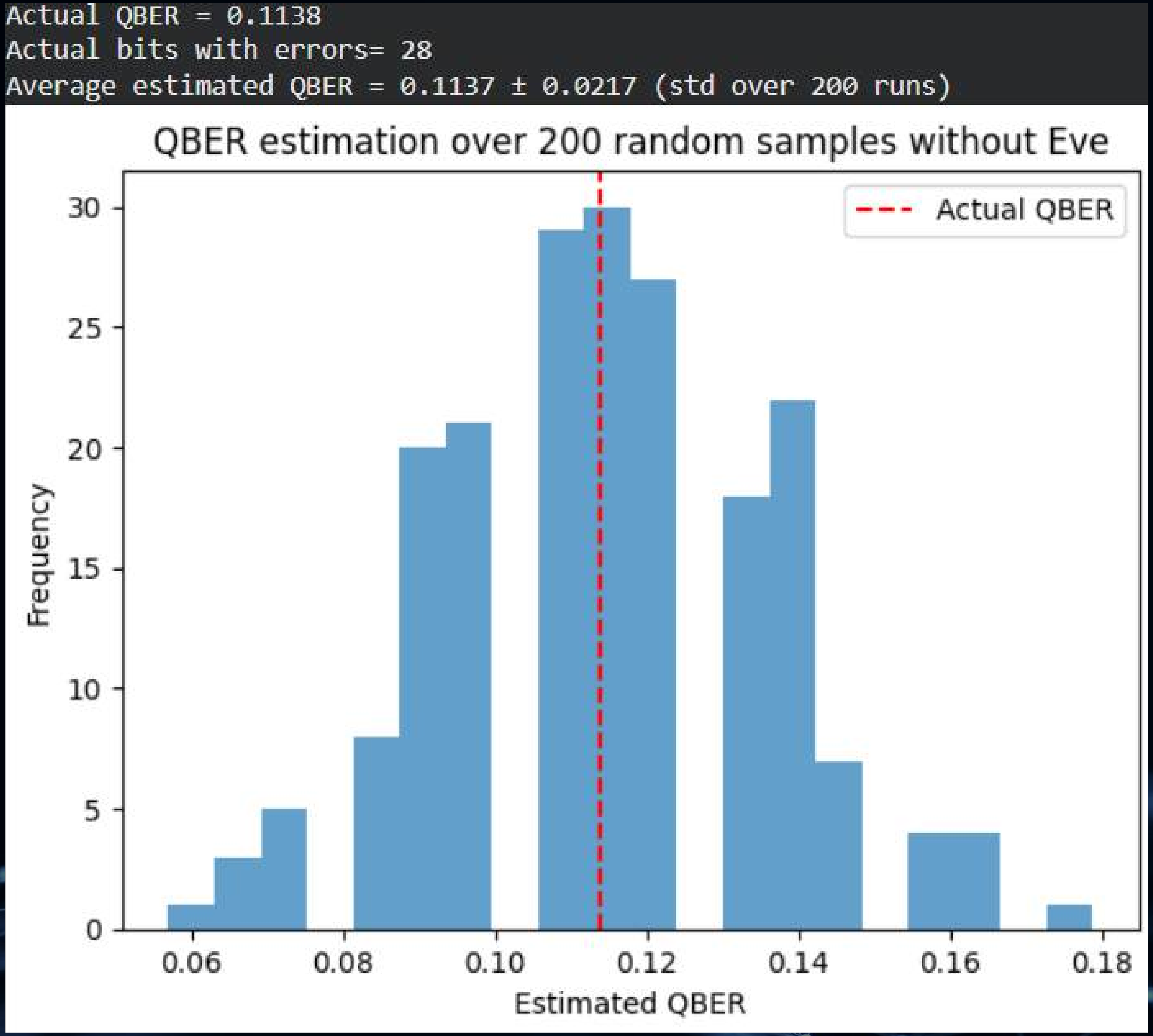
SECURITY OF BB84 PROTOCOL WITH EVE'S INTERCEPTION

Now, we tested the security of BB84 protocol with Eve's interception. For simplicity, we have ignored the Amplitude Decay Effect in this.

Eve will now intercept the signal coming from Alice and measure the Qubits using random basis of her choice. Her chosen basis may or may not match the basis of encryption that Alice choses, hence, Eve’s measurement may or may not be equal to what Alice sent.

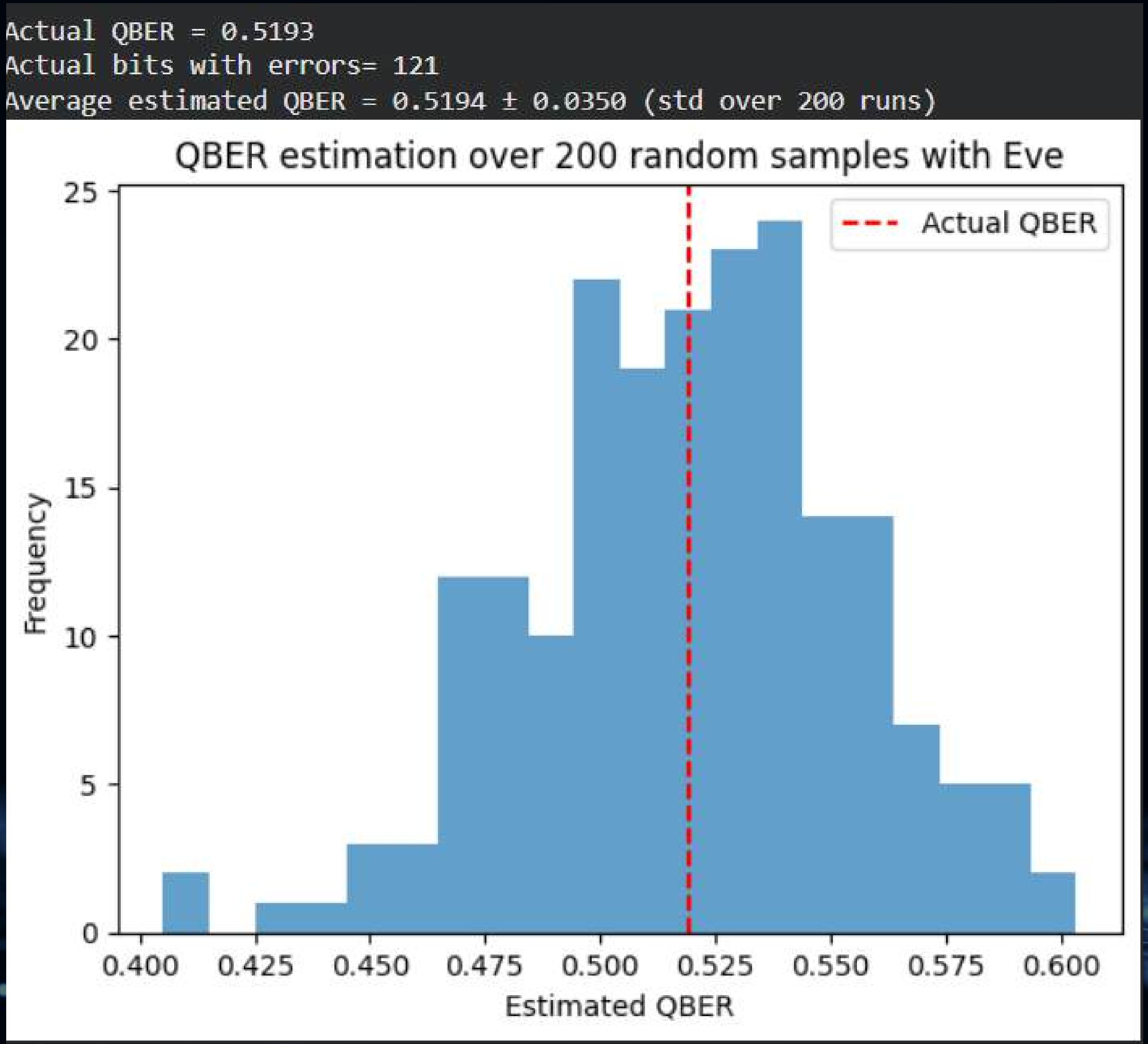
Since Eve measures the Qubit, she needs to create a copy to send to Bob or Bob will straightaway detect interception since signal from Alice won’t reach him.

Due to No Cloning theorem, Eve cannot make an exact copy of the Qubit that was sent originally by Alice. Since she is randomly chosing basis to measure qubits, here measurements will not be accurate.
She will send whatever she measured, encrypting it using whatever basis she used to measure it, to Bob.
This will directly lead to an increased QBER, when Alice and Bob will compare their bits for QBER calculation.



```
Actual QBER = 0.1138
Actual bits with errors= 28
Average estimated QBER = 0.1137 ± 0.0217 (std over 200 runs)
```

QBER without Eve Interception



```
Actual QBER = 0.5193
Actual bits with errors= 121
Average estimated QBER = 0.5194 ± 0.0350 (std over 200 runs)
```

QBER with Eve Interception

We can see a significant jump in QBER rate in the presence of Eve i.e. from 0.1138 to 0.5193. As Alice and Bob see this unusual jump in QBER, they detect the presence of Eve and immediately stop the QKD process and start over from scratch.

Hence, we have succesfully demonstrated the BB84 protocol is highly effective to do QKD and provides a great deal of security against eavesdroppers. We demonstrated that the presence of an eavesdropper is efficiently detected in this protocol, making it reliant. We dealed with noisy channels and eliminated their effects.

E91

E91 AND IMPLEMENATION

E91 is another Quantum Key Distribution protocol prosed in the year **1992** by **Ekert.**

This protocol makes us of Entangled nature of Qubits and CHSH inequality of **Bell's theorem** to prove the Quantum nature of the system.

In the protocol, a common source produces **Entangled** set of Photons which is being sent to Alice and Bob.

Alice and Bob use randomly selected Bases to measure the Entangled pair. After sending all the qubits they compare the Basis selection and analyse the outcomes to find the reliability of the Channel.

BELL'S THEOREM

Einstein called Quantum entanglement of particles as Spooky action at a distance. He proposed that Quantum particles carry some additional variables that dicatates how they behave with different basis.

Einstein stated that the measurement of one entangled pair will not affect the measurement of the other though they may be infinitely apart. He stated that entangled pairs just carry some hidden variables that dictate them how to collapse under different basis.

However that is not the case and this can be verified by analysing the correlation of the entangled qubits at different basis. Presence of hidden variables lead to bound correlation.

BASIS SELECTION

There are totally 4 different types of Basis. They measure along 0^0 , 22.5^0 , 45^0 and 67.5^0 and can be called as A1, B1, A2, B2.

Alice can select any of the bases in the set {A1, B1, A2} in equal probability and Bob can do the same in the set {B1, A2, B2}.

Hence Bob and Alice can only select the same Basis with Probability of 2/9.

CHSH INEQUALITY

CHSH Inequallity for the given system is
<A1,B1> - <A1,B2> + <A2,B1> + <A2,B2> <=2

Any system that carries the additional variables will follow the Inequality. But Quantum entangled systems will not follow the inequality and have higher values.

The expectation of all these values is 0 if we map bits(1,0) to (1,-1). Hence the Corelation coefficient is just Expectation of product.

Now by the statistical analysis of the base-mismatched data we know about reliability of the system.

CHSH RESULT

For the ideal circuit, without any noise the LHS of CHSH inequality goes to **2.9457**

For a circuit with the presence of Eve, the CHSH inequality does not hold and the LHS value goes to -**0.1559**

For Bit-flip error of 5% and photon loss of 5% the LHS value goes to **1.8074**.

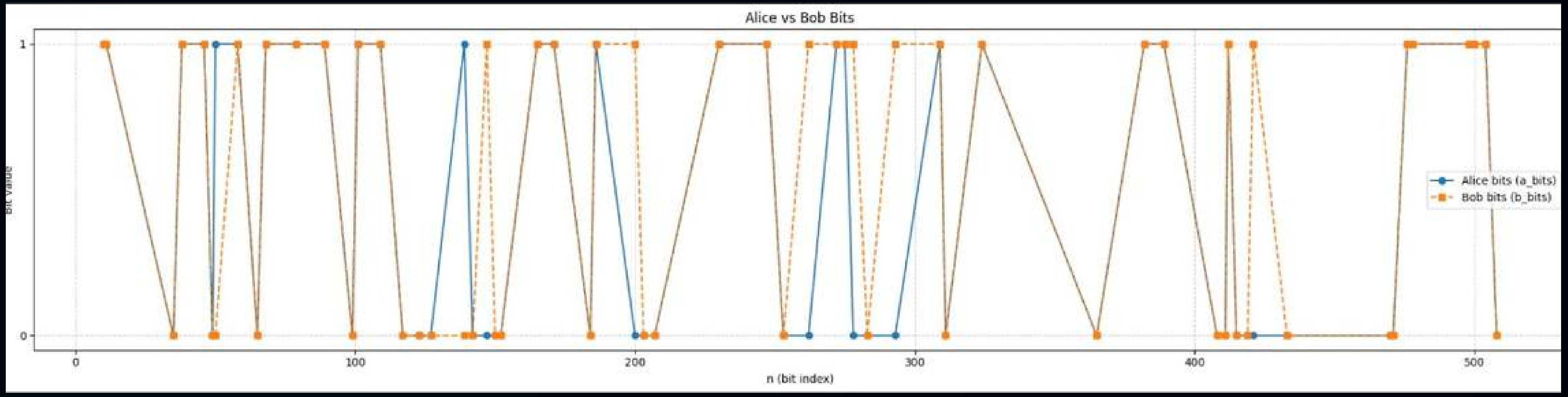
For Bit-flip error of 5% and photon loss 3% of the LHS value goes to **2.127**.

CONCLUSION ON E91

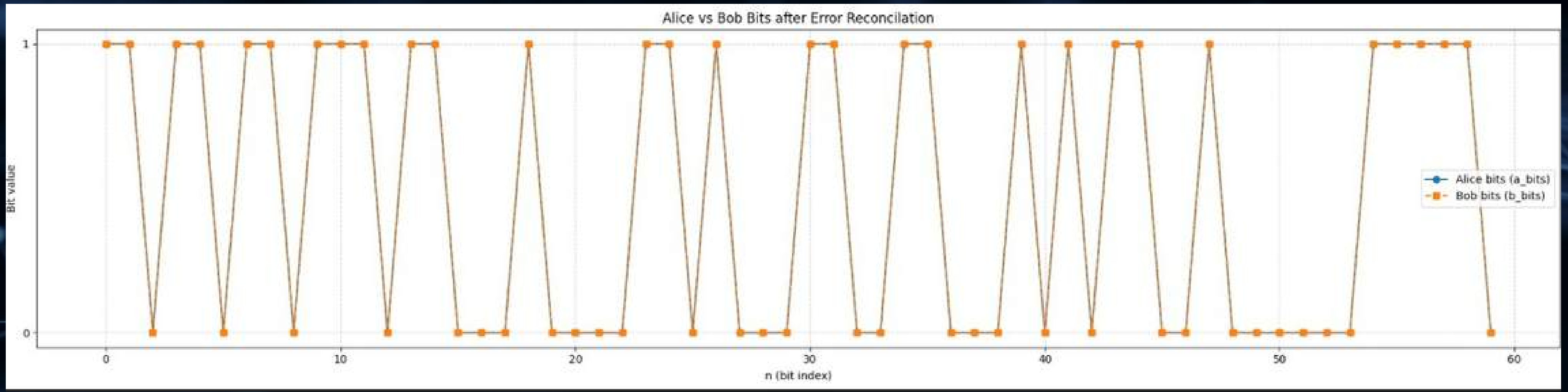
This clearly says that E91 protocol is reliable only in Channels with very less error rate.

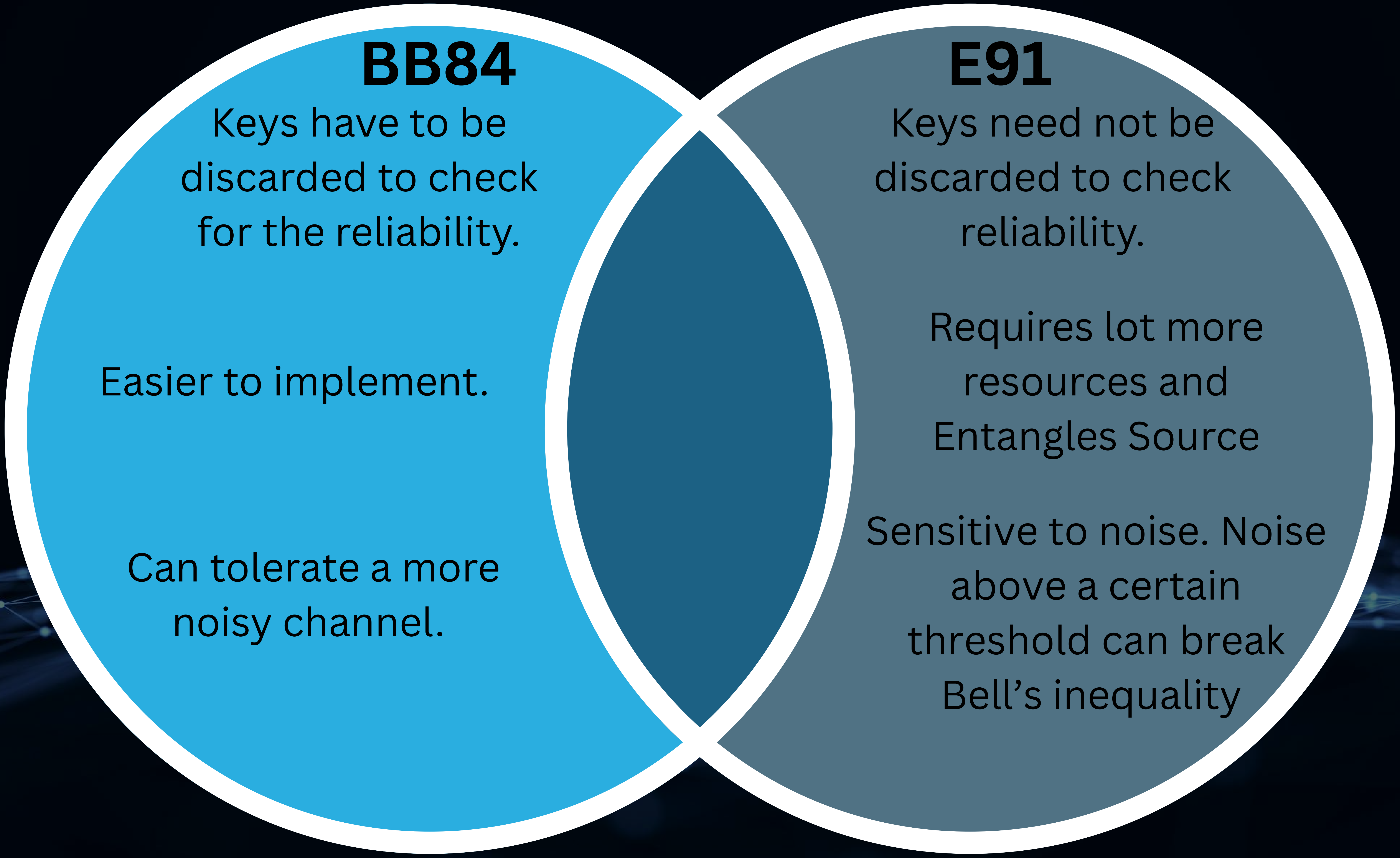
After checking the reliability of the channel, we use Cascade method to remove the error qubits that is present in Alice’s and Bob’s base-matched values in order to obtain the final key that can be used for Encryption.

Before Cascading



After Cascading





BB84

Keys have to be discarded to check for the reliability.

Easier to implement.

Can tolerate a more noisy channel.

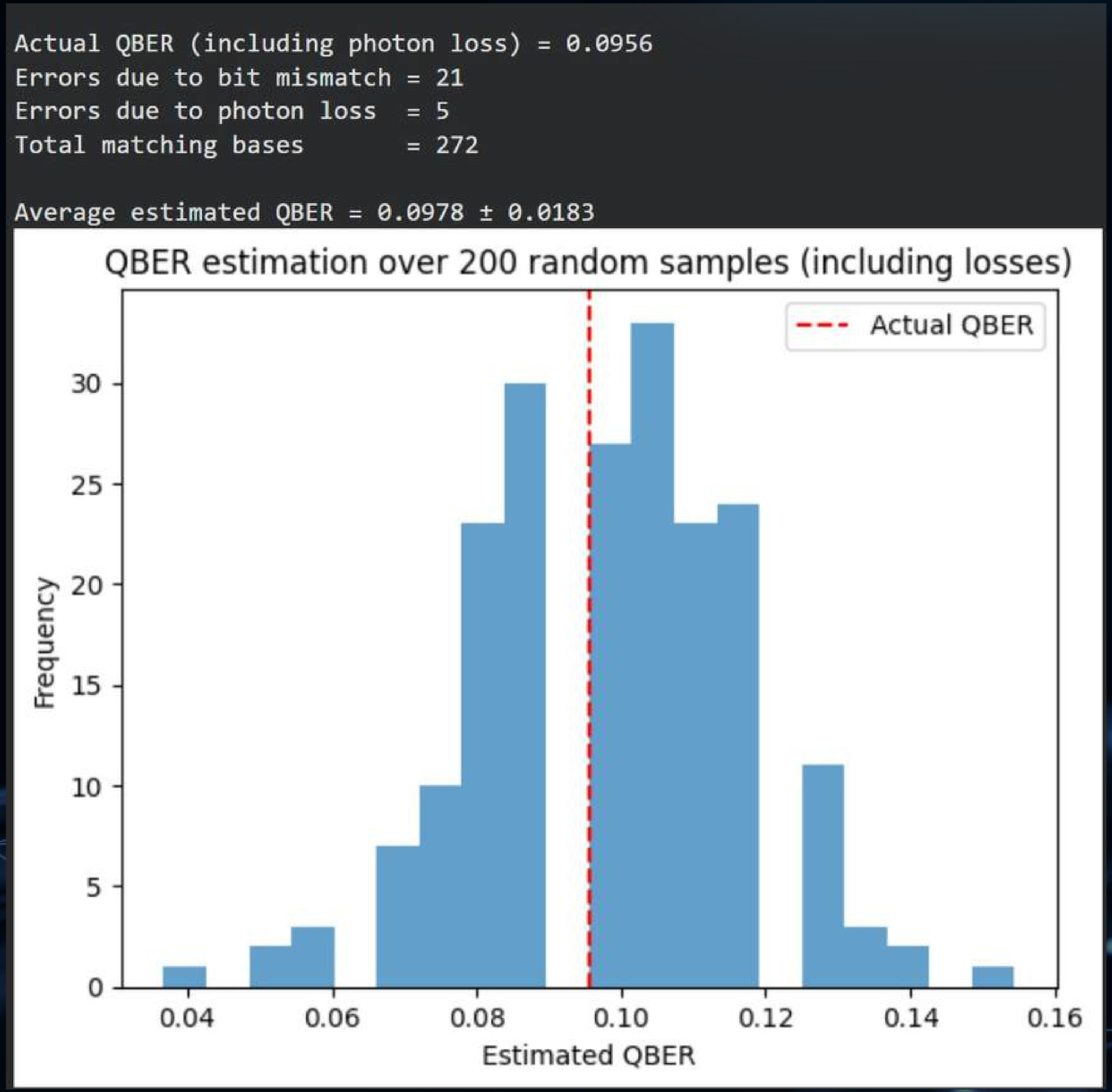
E91

Keys need not be discarded to check reliability.

Requires lot more resources and Entangles Source

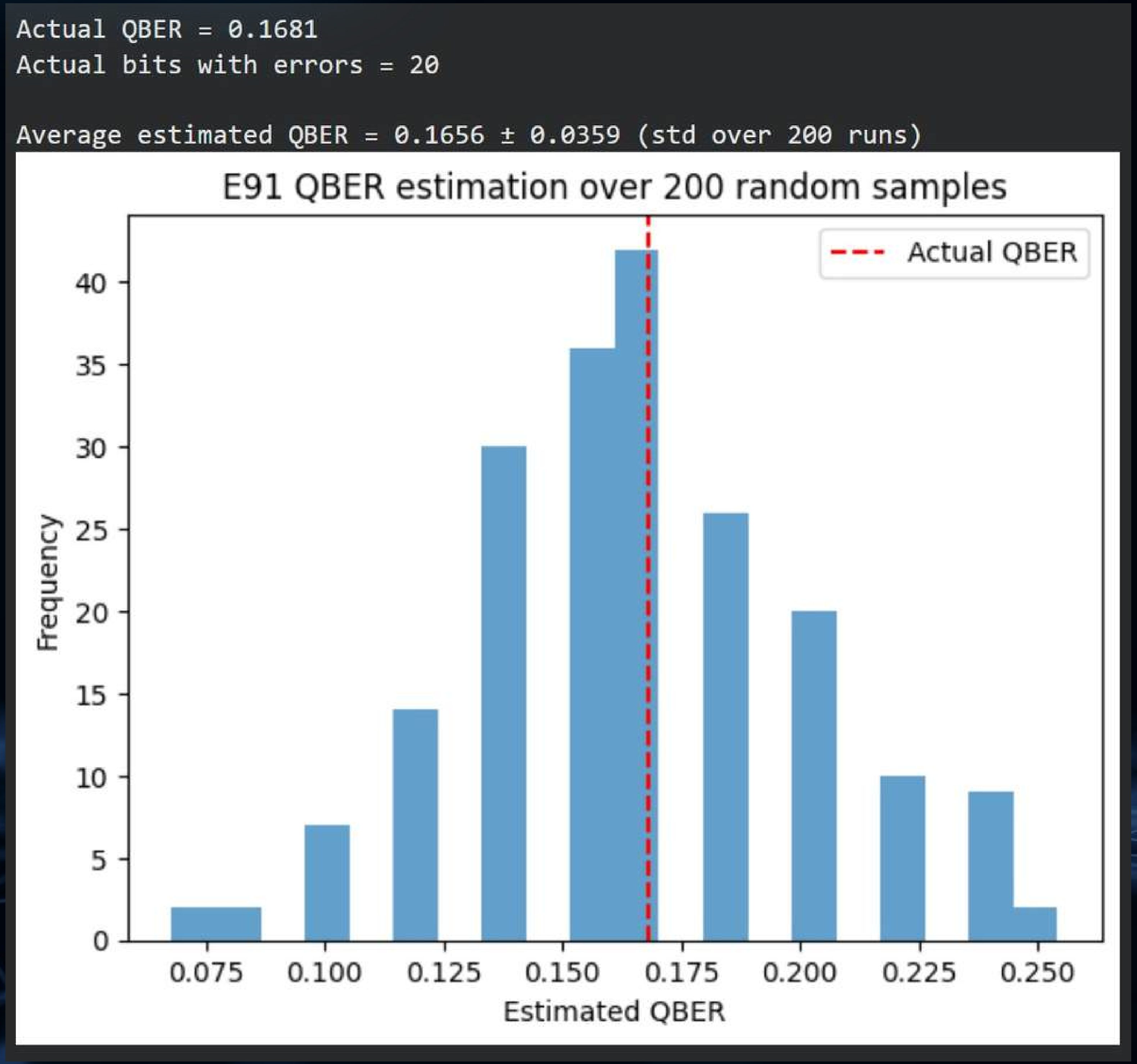
Sensitive to noise. Noise above a certain threshold can break Bell’s inequality

ERROR METRIC



```
Actual QBER (including photon loss) = 0.0956
Errors due to bit mismatch = 21
Errors due to photon loss  = 5
Total matching bases       = 272

Average estimated QBER = 0.0978 ± 0.0183
```



```
Actual QBER = 0.1681
Actual bits with errors = 20

Average estimated QBER = 0.1656 ± 0.0359 (std over 200 runs)
```

As expected the QBER in BB84 is less than E91 for given noise conditions and have hence run QKD protocols in QISKIT!!

TEAM GOTOSHDINO

THANK YOU