

# **When Noise Looks Like an Attacker: Evaluating BB84 Eavesdropping Detection on IBM Quantum Processors**



**Dr Adam Gorine,  
Muhammad Suhaib**  
**UWE, Bristol**



# Why is QBER important in BB84?

---

## Hardware Noise Produces QBER due to:

- **Decoherence:** T1 and T2 processes
- **Gate infidelity:** measures how much a real quantum gate deviates from its ideal operation.
- **Readout errors** occur when the measurement system reports the wrong qubit state.
- **Calibration drift:** The gradual change of a quantum processor's operating parameters over time.
- **Environmental/device imperfections**

## Takeaway:

So Alice and Bob don't automatically assume that a high QBER means Eve is present. They need to distinguish **normal system noise** from an unusually **high error rate**.

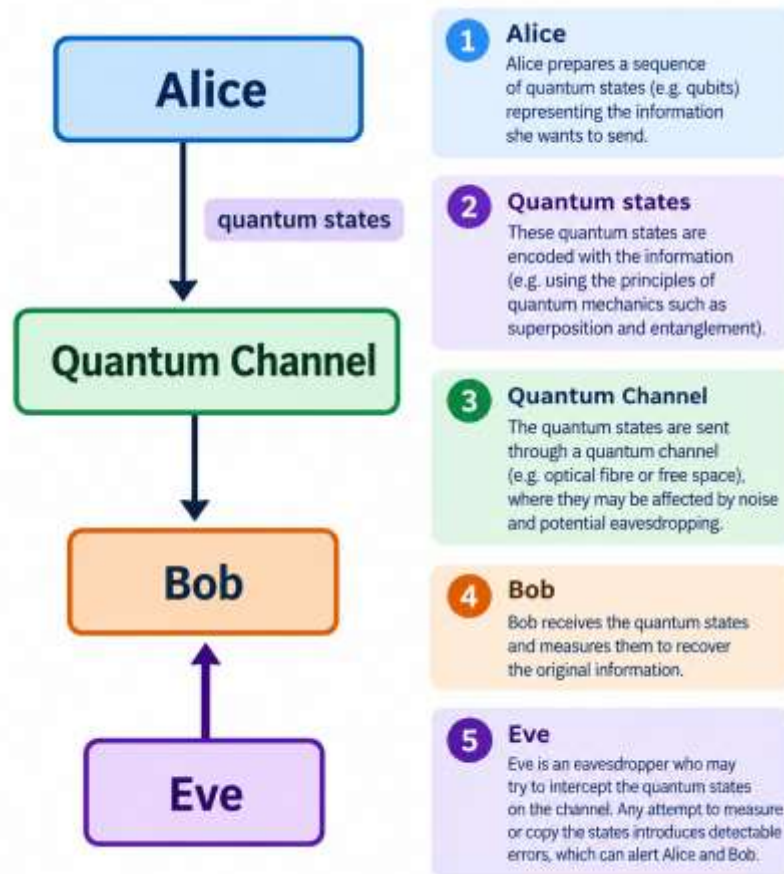
## Research Question:

# Can Current Quantum Computers Distinguish Noise from an Attack?

---

- **How does IBM superconducting hardware noise affect BB84 QBER?**
- **Can hardware-induced and attack-induced errors be statistically separated?**
- **What is the practical limit for reliable eavesdropping detection?**

# BB84: Security Through Detecting Disturbance



The diagram shows the basic **BB84-style quantum communication model**:

- **Alice** prepares quantum states (qubits) that encode information.
- The states travel through a **quantum channel**, such as optical fibre or free space.
- **Bob** receives and measures the quantum states.
- **Eve** is the eavesdropper. If she intercepts and measures the qubits, she can disturb their quantum states.
- Those disturbances can produce an increased **quantum bit error rate (QBER)**, allowing Alice and Bob to detect potential eavesdropping.

# Research Gap

---

## Previous research examined:

- BB84 through simulation
- Statistical QBER estimation
- Finite-key security
- Threshold-based detection
- BB84 implementation on IBM hardware

## Gap:

Previous studies do not provide repeated, cross-backend experiments directly comparing intrinsic hardware-noise distributions with controlled intercept-resend attack distributions.

•

# Quantum Hardware Configuration

---

## Three-Stage Experimental Framework

### 1. Simulation validation:

*Validate the BB84 protocol implementation in a controlled simulation environment to establish a baseline quantum bit error rate (QBER) and verify circuit correctness under configurable noise conditions.*

### 2. IBM Quantum execution

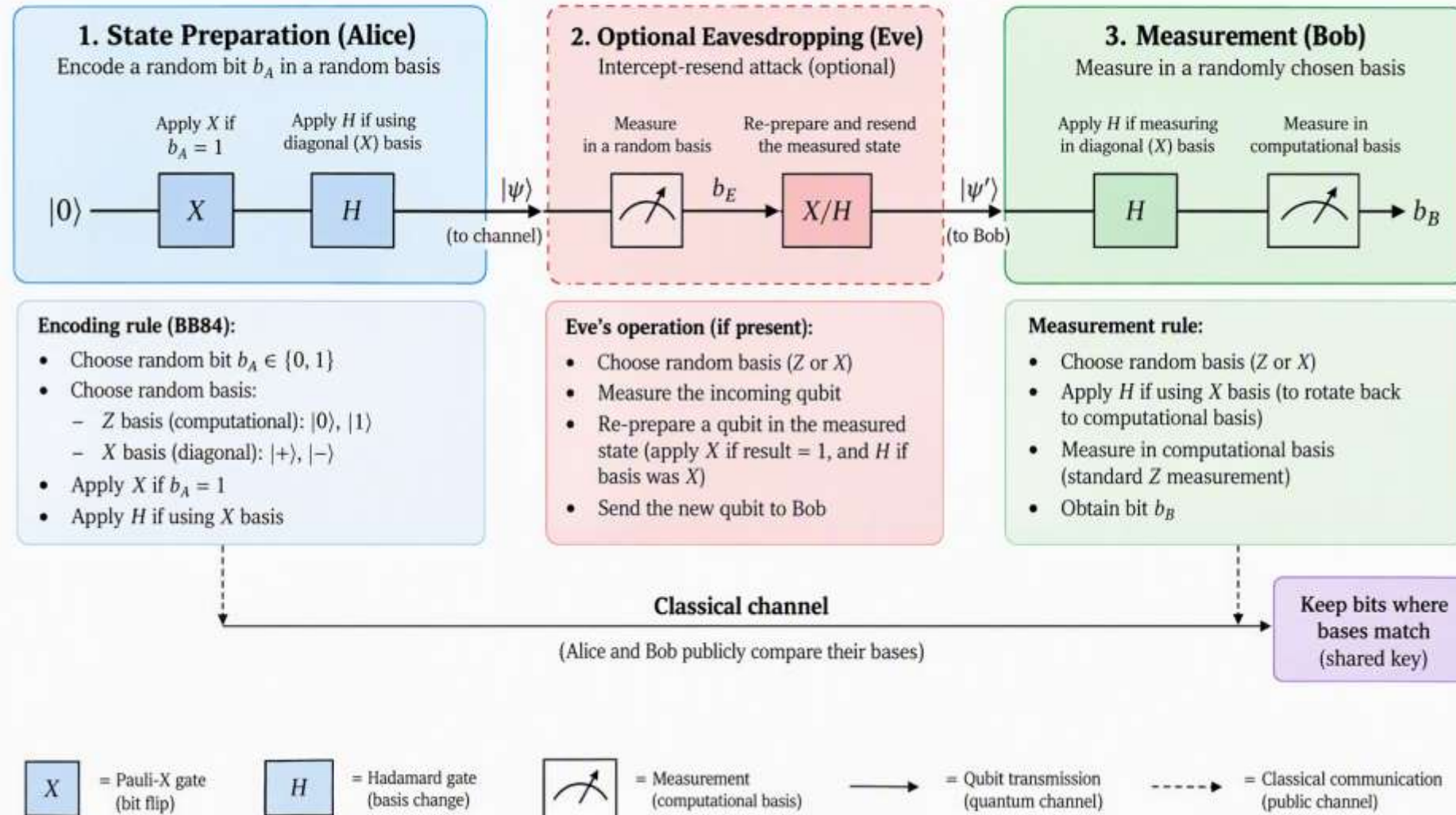
*The validated circuits are executed on IBM Quantum superconducting processors to assess the impact of practical hardware effects, including gate errors, decoherence, and measurement uncertainty.*

### 3. Statistical evaluation

*Measurement outcomes are statistically analysed to quantify QBER distributions, confidence intervals, and discrepancies between simulated noise models and the hardware's actual behaviour.*

# BB84 Quantum Key Exchange Experiment

Single-qubit circuit with state preparation, optional eavesdropping, and measurement



# BB84 Implementation on IBM Quantum Hardware

---

```
Qiskit
|
IBM Quantum Runtime
|
Superconducting Quantum Processor
|
Measurement Results
|
QBER Analysis
|
Statistical Testing
```

- IBM superconducting devices
- $N = 27\text{--}54$  qubits
- 300+ experimental runs

# Synthetic intercept-resend attack

## How was Eve Modelled?

- A physical eavesdropper cannot directly intercept qubits inside IBM's closed superconducting quantum processor.
- Therefore, we implemented a **synthetic intercept-resend attack** that reproduces Eve's behaviour within the quantum circuit.

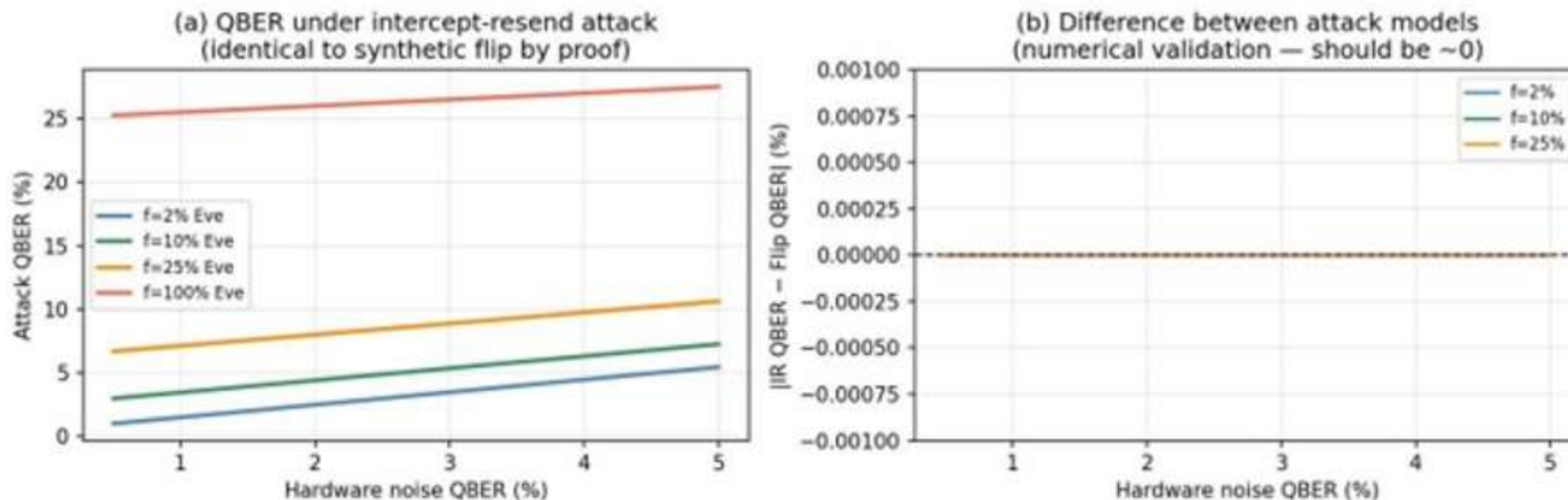
## Eve's Operation

1. Randomly selects a measurement basis.
2. Measures the qubit.
3. Resets the qubit.
4. Re-prepares a new qubit based on the measurement result.
5. Forwards the new state to Bob.

**Purpose:** Enable realistic attack experiments on IBM Quantum hardware.

# Validation of the Synthetic Intercept-Resend Attack

Synthetic Attack Equivalence Proof — Statistical Validation



# Experimental methodology

---

## **Three-Stage Experimental Framework**

- 1. Simulation validation**
- 2. IBM Quantum execution**
- 3. Statistical evaluation**

# Implementation and Results

---

## Iteration 1. Simulation Baseline

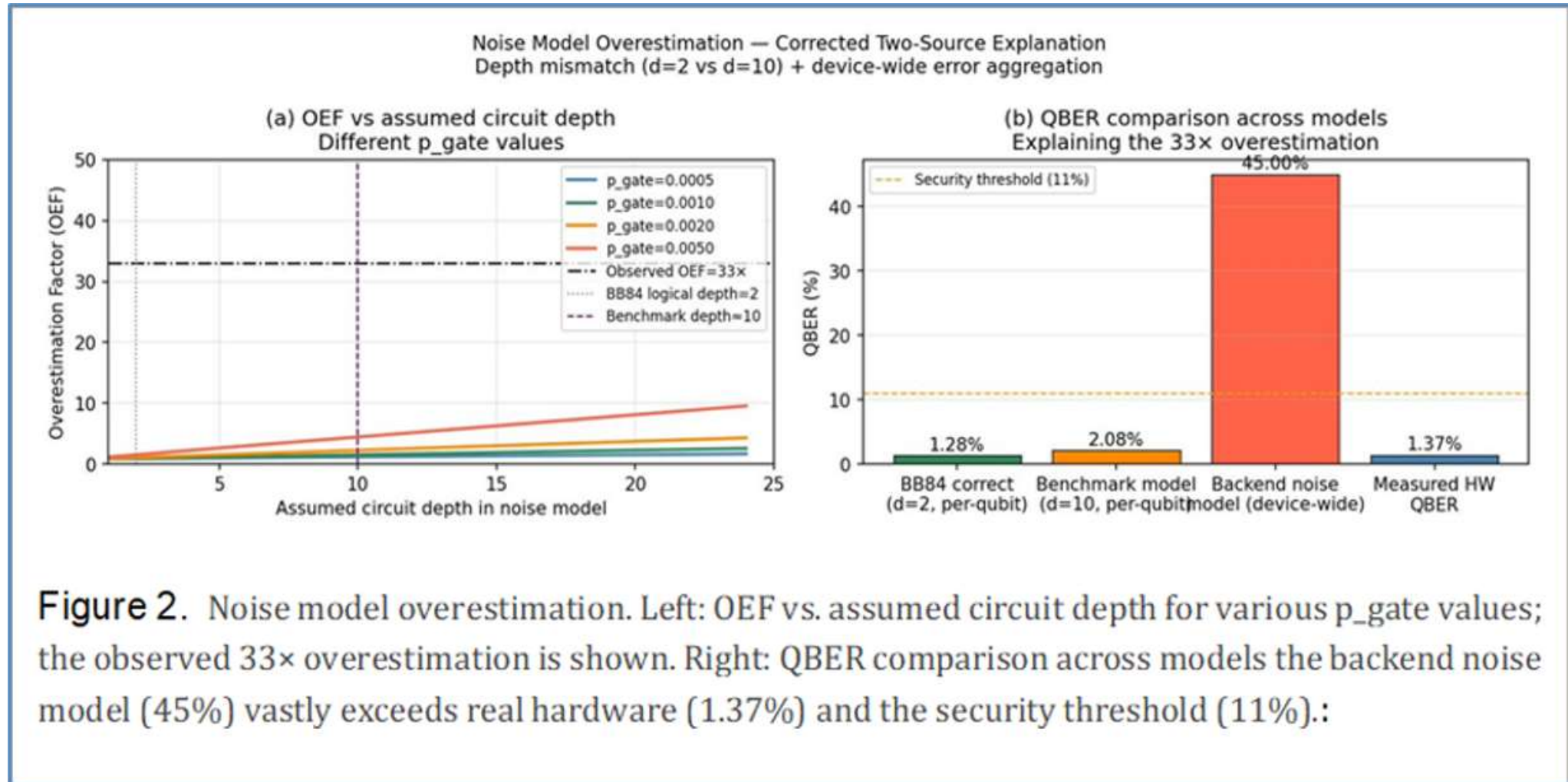
- We evaluated the BB84 protocol using the AER Simulator to establish a baseline under noise-free conditions.
- In the absence of an attack, the simulation yielded a QBER of 0%.
- When subjected to a full intercept-resend attack, QBER increased to 25%, aligning with the theoretical expectation for BB84.
- The sifting retention rate was 50%, consistent with the probability of Alice and Bob selecting matching measurement bases.

# Implementation and Results

## Iteration 2. Noise Model vs Real Hardware

- We compared three execution modes—ideal simulation, the IBM Qiskit backend noise model, and real IBM\_KINGSTON hardware.
- We found a major difference between simulated and experimental QBER values, as shown in Figure 2.
- The backend noise model predicted a QBER of about 45%, but the actual hardware showed a baseline QBER of  $1.19\% \pm 0.57\%$  (N = 27, 20 trials).
- This means the noise model overestimated QBER by roughly 33 times.

# Iteration 2. Noise Model vs Real Hardware



# Iteration 3: Multi-Trial Statistical Distinguishability

## Experimental design

- Multi-trial experiments were conducted across the evaluated IBM Quantum backends.
- Each configuration included **20 to 50 independent trials**.
- Qubit counts ranged from  $N = 27$  to  $N = 54$ .
- Hardware-only QBER distributions were compared with synthetic intercept-resend attack distributions.
- Distribution separation was assessed using the **two-sample Kolmogorov-Smirnov test**.

## Main observation

As the number of allocated qubits and hardware QBER variability increased, the empirical separation between the hardware-only and attack distributions generally decreased.

# Scaling Study Results

| Backend      | $N$ | Trials | Hardware QBER, mean $\pm$ SD | KS $D$ | 2% Eve detected? |
|--------------|-----|--------|------------------------------|--------|------------------|
| ibm_kingston | 27  | 20     | 1.19% $\pm$ 0.57%            | 1.00   | Yes              |
| ibm_fez      | 27  | 50     | 1.66% $\pm$ 1.37%            | 0.32   | Yes              |
| ibm_fez      | 36  | 50     | 1.85% $\pm$ 0.99%            | 0.34   | Yes              |
| ibm_kingston | 36  | 50     | 1.53% $\pm$ 0.99%            | 0.28   | Yes              |
| ibm_kingston | 36  | 20     | 1.88% $\pm$ 0.99%            | 0.35   | No               |
| ibm_kingston | 45  | 20     | 1.26% $\pm$ 0.50%            | 0.35   | No               |
| ibm_fez      | 54  | 50     | 2.90% $\pm$ 3.47%            | 0.24   | No               |

# Key findings

- The clearest separation occurred at  $N = 27$  on **ibm\_kingston**, where  $D=1.00$
- At  $N = 54$  on **ibm\_fez**, hardware **QBER** variability increased to **3.47%**, and the **KS statistic** decreased to  **$D = 0.24$**
- Trial count affected the decision: at  $N = 36$  on **ibm\_kingston**, detection succeeded with **50 trials** but failed with **20 trials**.
- The results indicate that detection depends on the **backend, qubit allocation, QBER variability and number of trials**.

# Seed 61 Case Study

## Near-identical hardware and attack QBER

For `ibm_fez`,  $N = 54$ , seed 61:

- **Hardware-only QBER:** 25.60%
- **Synthetic attack QBER:** 25.54%
- **Absolute difference:** 0.06 percentage points

## Interpretation

- In this individual trial, hardware noise produced a QBER close to the theoretical signature of a full intercept-resend attack.

# Scaling Mechanism

## Near-identical hardware and attack QBER

For `ibm_fez`,  $N = 54$ , seed 61:

- **Hardware-only QBER:** 25.60%
- **Synthetic attack QBER:** 25.54%
- **Absolute difference:** 0.06 percentage points

## Interpretation

- In this individual trial, hardware noise produced a QBER close to the theoretical signature of a full intercept-resend attack.

# Iteration 4: 44-Trial Validation

Can a 10% attack be distinguished from hardware noise?

## Experimental configuration

- Backend: **ibm\_kingston**
- Physical qubits:  $N = 36$
- Independent trials: **44**
- Shots per circuit: **512**
- Eve interception fraction:  $f = 0.10$
- Attack model: **Synthetic intercept-resend**

## Results:

- Hardware QBER:  **$0.896\% \pm 0.123\%$**
- Attack QBER (10% Eve):  **$0.873\%$**
- $\Delta\text{QBER} = 0.873\% - 0.896\% = -0.023$  percentage points

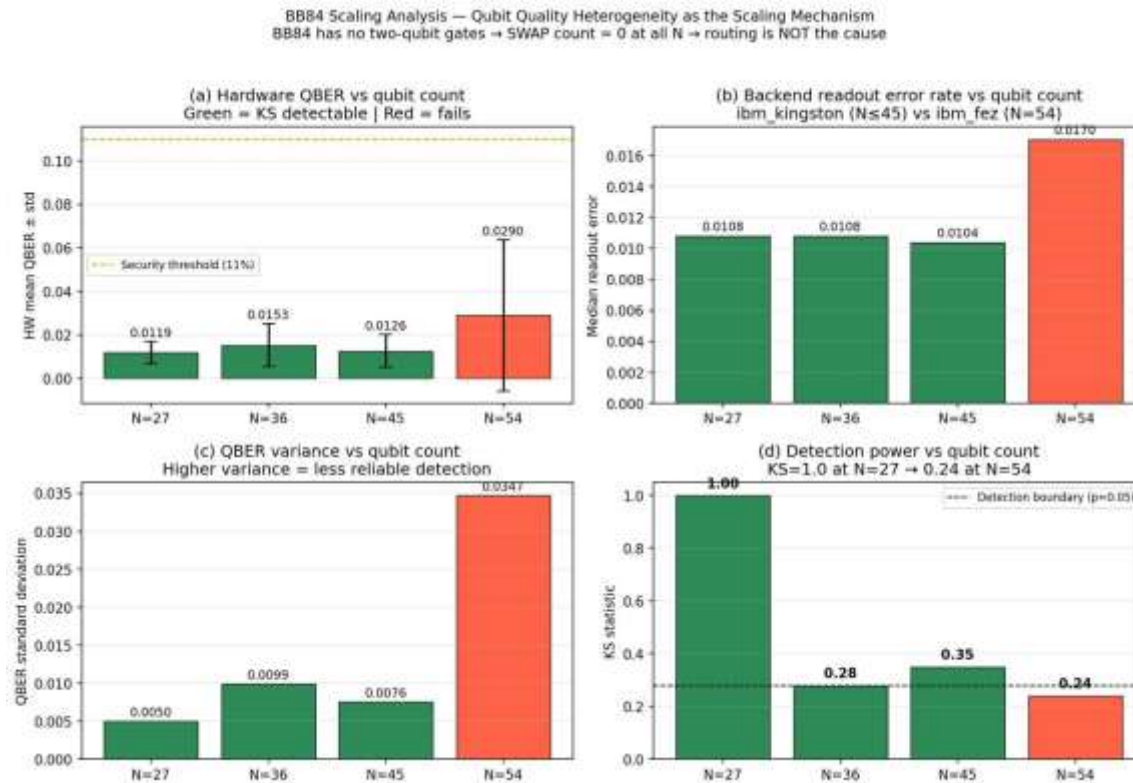
## Research Question

Can the QBER produced by a 10% intercept-resend attack be statistically distinguished from the backend's normal hardware QBER?

No statistically significant difference was detected between the hardware-only and 10% attack QBER distributions.

# Scaling analysis qubit calibration heterogeneity as the mechanism

**Figure 3.** Scaling analysis qubit calibration heterogeneity as the mechanism. (a) Hardware QBER mean  $\pm$  std vs. qubit count; red = detection fails. (b) Backend readout error rate ibm\_fez N=54 is 57% higher than ibm\_kingston. (c) QBER standard deviation grows six-fold from N=27 to N=54. (d) KS detection power drops from 1.00 at N=27 to 0.24 at N=54. Note: SWAP count = 0 at all N routing is not the cause.



# Practical Security: Finite-Key Analysis

---

## Main result

- For the experimental configuration:

### Minimum sifted key length

- 305 bits
- corresponding to approximately:
- 610 transmitted qubits

### At 305 sifted bits:

- Estimated QBER upper value: **16.5%**
- Finite-key rate: approximately **0.002**
- This represents the reported crossover point.
- Below 305 sifted bits:

**Finite-key secret-key rate is zero under the study's parameters.**

# Conclusions

---

## Key Finding 1

- **QBER is a distribution, not a fixed value.**
- Hardware characteristics significantly affect observed BB84 error rates.
- The experimental results indicate that the BB84 quantum bit error rate (QBER) should be regarded as a hardware-dependent statistical distribution rather than a fixed value.

## Key Finding 2

- **N = 27:** Hardware and attack distributions were clearly separable.
- **N = 54:** Hardware variability caused substantial overlap between attack and non-attack conditions.

## Takeaway

Current quantum hardware can generate QBER patterns that resemble **eavesdropping attacks**, making attack detection increasingly dependent on hardware quality and statistical analysis rather than fixed security thresholds.