Implementation of Quantum Communication Protocols with IBM Qiskit [Checking Security of Protocols and Effect of Noise]

By Nirupan Basak

WORD COUNT 9057 TIME SUBMITTED 06-JUL-2022 04:53PM

PAPER ID 88087526

Implementation of Quantum Communication Protocols with IBM Qiskit

[Checking Security of Protocols and Effect of Noise]

Submitted in partial fulfillment of the requirements for the award of the degree

Master of Technology

in

Computer Science

Submitted by:

Nirupam Basak

(Roll No.: CS2004)

Guided by:

Dr. Goutam Paul

Cryptology and Security Research Unit,

Indian Statistical Institute

Kolkata 700108

Declaration

I hereby declare that the work presented in the project report entitled *Implementation of Quantum Communication Protocols with IBM Qiskit [Checking Security of Protocols and Effect of Noise]* contains my work under the supervision of Dr. Goutam Paul, ISI Kolkata. At places, where ideas are borrowed from other sources, proper references, as applicable, have been cited. To the best of my knowledge, this work does not emanate from or resemble other work created by persons other than mentioned herein.

Signature:
Date:

Certificate From Supervisor

I hereby certify that the dissertation thesis entitled *Implementation of Quantum Communication Protocols with IBM Qiskit [Checking Security of Protocols and Effect of Noise]* submitted by Nirupam Basak in partial fulfillment for the award of the degree of Master of Technology in Computer Science is completed under my supervision.

Signature:
Date:

ABSTRACT

Quantum communication is an important ingredient in future information processing technologies and transfers a quantum state from one location to another. There are three types of Quantum communication protocols: QKD (Quantum Key Distribution), QSDC (Quantum Secret Direct Communication), and QSS (Quantum Secret Sharing). Here we are implementing some protocols from QSDC and QSS in ideal simulators, noisy simulators, and real backends with IBM Qiskit. Also, we are implementing some quantum attacks (e.g. intercept-and-resend, entangle-and-measure, DoS, etc.). We are discussing security against attacks and the effect of hardware/ simulator noise before as well as after applying some error mitigation techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 World of Quantum

1.1.1 Motivation

In recent years, technology is growing very fast. Lots of information is required to process in the field of Cyber Security, Artificial Intelligence (AI), Machine Learning, Deep Learning, etc. The computational requirements have skyrocketed with the advent of the processing of Big Data. High-Performance Computing (HPC) and parallel computing techniques boosted computational complexity. But what about if we can achieve some techniques to process an infinite number of information at once? Quantum computing provides one such possibility by storing information as quantum data and thereby increasing computational power. Shor's algorithm [1] is one significant piece of evidence for this. Along with this high computational power, quantum computing brings some challenges for us, especially in the field of cyber security. In the field of cryptology, most security protocols are designed based on the computational power of the classical computer. But with its very high computational power quantum computing breaks the security in many cases. So, in the field of quantum computing, increasing the computational power is not the only thing to concentrate on. We have to focus on security also.

1.1.2 Postulates of Quantum Mechanics

Quantum computation is developed over three postulates of quantum mechanics. They are:

- **Postulate 1.** Pure state of a quantum system is represented by unit vectors of a Hilbert space. For any two-state $|\phi\rangle, |\psi\rangle$ and $\alpha, \beta \in \mathbb{C}$, $\alpha|\phi\rangle + \beta|\psi\rangle$ is also a quantum state.
- **Postulate 2. (Evolution)** Time evolution of a quantum system is governed by unitary operation.
- **Postulate 3.** Any physical quantity associated with a quantum system can be represented by a Hermitian operator acting on the Hilbert space. These physical quantities are called 'observables'.
- **Postulate 4. (Measurement Principle)** Measurement of a quantum system is an interaction that measures some observables of the system. The followings are the properties of measurement:
  - Measurement outcome will always be one of the eigenvalues of the observable $\mathcal{A}$.
  - (Born Rule) The probability of obtaining a particular eigenvalue λ_i of the observable $\mathcal{A}$ for some quantum system $|\psi\rangle$ is given by the probability distribution $P[outcome \quad is \quad \lambda_i] = |\langle\lambda_i|\psi\rangle|^2$, where $|\lambda_i\rangle$ is the normalized eigenvector of $\mathcal{A}$ corresponding to the eigenvelue λ_i and $\langle\lambda| = |\psi\rangle^\dagger$.
  - (Wave Function Collapse) During measurement system undergoes collapse of wave function and the post measurement state becomes the normalized eigenstate $|\lambda_i\rangle$ corresponding to the eigenvalue λ_i.

1.1.3 Quantum States and its Representations

Above we saw that a quantum can state can be realized as a unit vector in a Hilbert space. But there is another realization of quantum states as a density matrix.

Definition 1 (Density Matrix) *A density matrix ρ is a linear operator acting on a Hilbert space with following properties:*

- ρ *is Hermitioan* ($\rho = \rho^\dagger$)
- ρ *is normalized* ($\operatorname{tr}\rho = 1$)
- ρ *is positive operator* ($\rho \geq 0$).

Definition 2 (Pure state) *A state $|\psi\rangle$ is called a pure state if $|\psi\rangle$ is normalized vector. In this case density matrix is defined as $\rho = |\psi\rangle\langle\psi|$.*

Definition 3 (Ensemble) *Suppose a quantum system is one of several states $|\psi_i\rangle$ with respective probabilities p_i. Then we shall call $\{|\psi_i\rangle, p_i\}$ as an ensemble of states.*

Definition 4 (Density matrix of quantum state) *Let us consider a quantum state as an ensemble $\{|\psi_i\rangle, p_i\}$. Then the density matrix of the above system is defined as*

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

Note that, if ρ is a pure state then $\operatorname{tr}\rho = 1$.

Definition 5 (Mixed state) *If a system ρ is such that $\operatorname{tr}\rho^2 < 1$, then we call that state a mixed state.*

Definition 6 (Product state) *Let us consider a quantum system H containing two subsystem A, B. Then a state $|\psi\rangle_{AB} \in H$ is called a product state if $|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B$, where $|\psi\rangle_A, |\psi\rangle_B$ are two quantum states in A, B respectively. For any product state $|a\rangle \otimes |b\rangle$ we write it as $|ab\rangle$.*

Definition 7 (Entangle state) *(For a system made of two subsystems) If a state is not a product state then it is said to be an entangled state.*

1.1.4 Quantum Computation

We have given a brief introduction to quantum mechanics and quantum states. Now we will briefly discuss quantum computing. In quantum computation, the main components are qubits, quantum gates, quantum circuits, and the measurement of quantum qubits.

Qubits

Qubit is basic unit of quantum information in quantum computation. It corresponds a quantum system in two dimensional Hilbert space. We can also think about d-dimensional generalization, called qudits. Since qubits are associated with Hilbert space, basis of that space is said to be basis of the qubit also. Two usefull bases in quantum computation are computational basis and diagonal or Hadamard basis. For a single qubit, computational basis is $\{|0\rangle, |1\rangle\}$, where, $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$ and Hadamard basis is $\{|+\rangle, |-\rangle\}$, where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. For two qubit, computational can be written as $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.

Quantum Gates

According to the second postulate in §1.1.2, quantum evolutions are operations of unitary operators. In quantum computation, those operators are called quantum gates which are applied to qubits to evolve the system. If a gate is applicable on a single qubit then it is nothing but a 2×2 unitary operator. Some common single-qubit gates are I, X, Y, Z. These are called Pauli gates. Except these two useful single-qubit gates are T and Hadamard. A useful two-qubit gate is CX (controlled-X), where one qubit is the control qubit and another qubit is a target qubit where an X gate will be applied if the control qubit is in state $|0\rangle$.

Measurement of Qubits

To retrieve the information after processing we have to measure qubits. In IBM qiskit only allowed measurement is of the observable Z, which can be done by measuring qubits on a computational basis. So after measurement, we get measurement outcome as 0 or 1 and the state collapses to $|0\rangle$ or $|1\rangle$ respectively. If we measure multiple qubits we shall get a sequence of classical bits as an outcome and the system will be collapsed in state $\otimes\{|0\rangle, |1\rangle\}$. If we require to measure any other observable then we have to change the basis accordingly.

Quantum Circuits

A quantum circuit is a collection of quantum registers (possibly along with classical registers), unitary gates, and quantum measurement operations, where quantum registers can be thought of as a quantum version of classical registers corresponding to the qubits. In IBM qiskit all quantum registers are initially set in state $|0\rangle$. To perform quantum computation, we have to break the computation into several instructions. Depending on these instructions we create a quantum circuit by applying appropriate gates. Depending on our final requirement we measure proper observable to get the outcome.

1.2 Noise in Quantum Computation

Today's quantum computers fall under the category of NISQ (Noisy-Intermediate Scale Quantum) [2] devices, which have few qubits and are noisy. Generally, noises are came at the time of preparing a state, applying gates as well as at the time of measurement. We can think of different types of quantum noises. Here we are discussing a few of them.

1.2.1 Types of Noise

Bit-flip Noise

Bit-flip error is one kind of Pauli error where with a small probability, p, X gate and with probability $1-p$ identity gate is being applied. Since here error is due to flipping the qubit on computational basis, it is called bit-flip error.

Phase-flip Noise

Phase-flip error is another kind of Pauli error where with a small probability, p, Z gate and with probability $1-p$ identity gate is being applied. Here error is due to flipping qubits on Hadamard basis which is the same as flipping the phase on computational basis. And so it is called phase-flip error.

Depolarizing Noise

The depolarizing channel is defined as: $E(\rho) = (1-\lambda)\rho + \lambda \mathrm{Tr}[\rho]\frac{I}{2^n}$ with $0 \leq \lambda \leq 4^n/(4^n-1)$ where λ is the depolarizing error parameter and n is the number of qubits.

- If $\lambda = 0$ this is the identity channel $E(\rho) = \rho$
- If $\lambda = 1$ this is a completely depolarizing channel $E(\rho) = I/2^n$
- If $\lambda = 4^n/(4^n-1)$ this is a uniform Pauli error channel: $E(\rho) = \sum_j P_j \rho P_j/(4^n-1)$ for all $P_j != I$.

Thermal Relaxation Noise

Physical qubits are photons in some excited or ground state. Any particle always wants to stay in an equilibrium state. To return to an equilibrium state photons release energy. This results in an error in quantum computation. This error is called thermal relaxation error. There are two types of relaxation – one is decaying of amplitude in the wave function, and another is decaying of phase. If T_1 is decay constant for amplitude and T_2 is same for phase then the relation between them is $T_2 \leq 2T_1$ [24].

1.2.2 Quantum Error Correction and Error Mittigation

Since quantum operations are noisy, to get the appropriate result we have to correct the error or at least reduce the error significantly. There are some proposed ideas for this. There are several error-correcting codes like bit-flip code, phase-flip code, Shor code, stabilizer code, etc. But to perform error correcting operations we need lots of qubits. Since we have limited resources (access to IBM quantum machine with 5-qubit is open to all), we can't perform quantum error correction in the real backend. So we used another technique to reduce quantum error, called error mitigation.

Quantum Error Mitigation

We can divide error mitigation in two parts – measurement error mitigation and gate error mitigation.

- **Step 1 [Measurement error mitigation]:** The idea behind this mitigation is as follows [6]:
  - If n qubit is measured then 2^n outcomes are possible. Create circuits for each outcome (with measurement) and execute them. The output will be noisy. Create a linear operator as a matrix (say, M) using these noisy outputs. Then clearly if C_{ideal} and C_{noisy} be the ideal and noisy output respectively then we have $C_{noisy} = MC_{ideal}$. Therefore we will get ideal output from noisy one by applying M^{-1} on C_{noisy} as $C_{ideal} = M^{-1}C_{noisy}$.
    Note: To prepare circuits for different outcomes we use some one qubit gates which are also noisy, so C_{ideal} is not properly *ideal*.
- **Step 2 [Gate error mitigation]:** By looking carefully one can see that the idea behind measurement error mitigation works if measurement is done at the end only. So that idea is not applicable here. To perform gate error mitigation there are several protocols like zero noise extrapolation (ZNE), probabilistic error cancellation (PEC), etc. Here we will discuss the unitary folding method of ZNE [8] which we used in this thesis work. The idea of ZNE is as follows:
  - **Noise-scaling:** Let L be the circuit we prepared to perform some quantum computation. Then by replacing the circuit with $LL^\dagger L$ we shall be able to scale up the noise. If $\lambda_i \geq 1, i \geq 1$ are scalling parameter, then prepare circuits for $\{\lambda_i\}_i$. Execute all the circuits and collect the expected value. There are two versions of noise-scaling – 1) Non-adaptive, where all λ_i are chosen beforehand; 2) Adaptive, where λ_1 is chosen beforehand (typically set to 1) and remaining λ_i are chosen by the ZNE algorithm itself.
  - **Extrapolation:** Fit above expected values in best fitted curve. Let $E(\lambda)$ be the curve. Then put $\lambda = 0$ to get required result as $E(0)$.

Chapter 2

Quantum Secret Sharing

2.1 Introduction

Secret sharing was independently proposed by Adi Shamir [3] and George Blakley [4] in 1979. It is a method to distribute a 'secret' among a group in such a way that no individual will be able to recover the secret with the information that individual has, but with information from a sufficient number of individuals, the secret can be reconstructed. In a secret sharing, there is one 'dealer' who shares the secret among n parties. If any m parties will reconstruct the secret using their 'shares' but less than m parties will not reconstruct the secret then we will call it a '(m, n)-secret sharing scheme'. Clearly if $m = 1$ then it is a trivial sharing.

In 1999 Hillery et. al. [5] proposed the first Quantum Secret Sharing (QSS) scheme which was a quantum version of the classical secret sharing scheme. They proposed two protocols, one is to share classical secrets using quantum states and another is to split quantum information. Both the protocols were $(2, 2)$-secret sharing. In the same paper, they proposed one possible generalization of the first protocol (sharing classical message) up to $(3, 3)$-secret sharing. Then Xiao et. al. [9] proposed a (n, n)-secret sharing protocol which is a generalized version of Hillery's first protocol. In the last 20 years Zhang, Guo, Panigrahi, Hsu, Liao [12, 13, 10, 11, 14, 15] proposed some more QSS schemes.

Here we Implemented QSS protocols proposed by Hillery et. al. [5] with IBM qiskit. In 2020, Dintomon et. al. [16] implemented first protocol in IBMQ-5-Tenerife. We implemented both the protocols in several IBM backends as well as in simulators. Then checked their security against different attacks. Also, we created a noise model using parameters from 'ibmq_jakarta' to check the effect of gate errors. Then we applied error mitigation on IBM Backend result as well as noisy simulator result. We also implemented the generalized QSS proposed by Li Xiao et. al. [9].

2.2 Splitting of Classical Message

In the paper, [5] author proposed a protocol based on Greenberger–Horne–Zeilinger (GHZ) state. There are three parties (one dealer, and two individual receivers) involved in this protocol, each has one qubit from the above entangled state. They measured their qubits randomly and then two individuals together construct the dealer's secret.

2.2.1 Brief of the protocol

- **Step 1:** Alice (dealer), Bob, and Charlie each have one particle from a GHZ triplet that is in state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

- **Step 2:** They each choose at random a direction (x or y) and measure their particle in that direction and announce the directions in which they have measured.
- **Step 3:** They throw away the results where an odd number of measurements is performed in the y direction.
- **Step 4:** Bob and Charlie derive the secret (result) of Alice depending on the measurements of their two qubits and the mathematical details provided below.

Let us define eignstates of x and y as

$$|+x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |-x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$|+y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \quad |-y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$$

Then $|\Psi\rangle$ becomes

$$\begin{aligned}
|\Psi\rangle &= \frac{1}{2}\left[(|+x\rangle_c|+x\rangle_b + |-x\rangle_c|-x\rangle_b)(|+x\rangle_a) + (|+x\rangle_c|-x\rangle_b + |-x\rangle_c|+x\rangle_b)(|-x\rangle_a)\right] \\
&= \frac{1}{2}\left[(|+y\rangle_c|+x\rangle_b + |-y\rangle_c|-x\rangle_b)(|-y\rangle_a) + (|+y\rangle_c|-x\rangle_b + |-y\rangle_c|+x\rangle_b)(|+y\rangle_a)\right] \\
&= \frac{1}{2}\left[(|+x\rangle_c|+y\rangle_b + |-x\rangle_c|-y\rangle_b)(|-y\rangle_a) + (|+x\rangle_c|-y\rangle_b + |-x\rangle_c|+y\rangle_b)(|+y\rangle_a)\right] \\
&= \frac{1}{2}\left[(|+y\rangle_c|+y\rangle_b + |-y\rangle_c|-y\rangle_b)(|-x\rangle_a) + (|+y\rangle_c|-y\rangle_b + |-y\rangle_c|+y\rangle_b)(|+x\rangle_a)\right]
\end{aligned}$$

where a, b, c in subscript denotes Alice, Bob and Charlie's qubit respectively. From this we can make measurement outcome table 2.1:

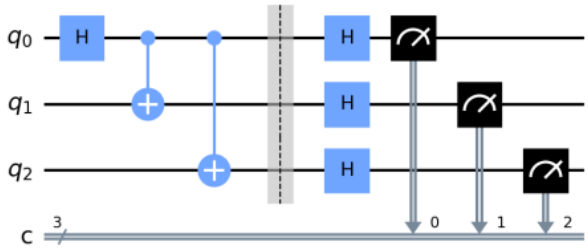
		Charlie			
		$\|+x\rangle$	$\|-x\rangle$	$\|+y\rangle$	$\|-y\rangle$
Bob	$\|+x\rangle$	$\|+x\rangle$	$\|-x\rangle$	$\|-y\rangle$	$\|+y\rangle$
	$\|-x\rangle$	$\|-x\rangle$	$\|+x\rangle$	$\|+y\rangle$	$\|-y\rangle$
	$\|+y\rangle$	$\|-y\rangle$	$\|+y\rangle$	$\|-x\rangle$	$\|+x\rangle$
	$\|-y\rangle$	$\|+y\rangle$	$\|-y\rangle$	$\|+x\rangle$	$\|-x\rangle$

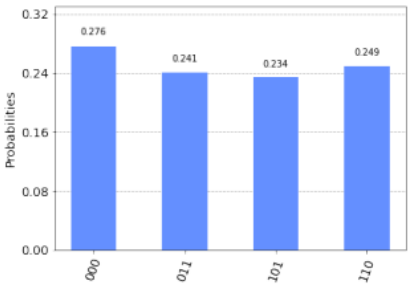
Table 2.1: Relation between outputs (QSS)

This table shows what will be Alice's output depending on Charlie and Bob's output. Select the row corresponding to Bob's measurement output and the column corresponding to Charlie's measurement output. Alice's measurement output is given by the intersection of the above row and column. Also notice that an odd number of y directions of measurements are not there in the table and they are thrown away in step 3.

2.2.2 Implementation with Qiskit

We imnplemented above protocol with IBM qiskit. Here is one of the circuits and corresponding histogram:





The first part is the creation of the GHZ triplet and the last part is measurement. Here everyone measures on Hadamard (x) basis.

2.2.3 Attack Implementation

Since there is no communication except sharing the entanglement, the attack is only possible at the time of entanglement sharing. We implemented three common attacks performed by Eve: Intercept-and-resend, Entangle-and-measure, and Daniel-of-service (DoS).

Intercept-and-resend attack

In this case, some eavesdropper creates interceptions and resends corresponding resulting qubits to the destination. Then, Alice, Bob, and Charlie follow the protocol. To create interception we used measurement on a randomly chosen z (computational) or x (Hadamard) basis. We ran the protocol 500 times and got the following plot of interception detection rate:

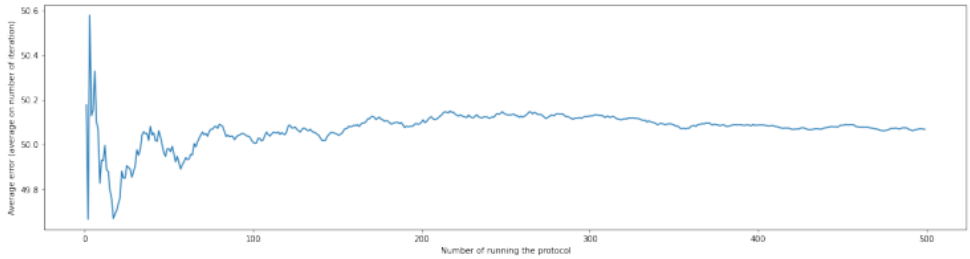
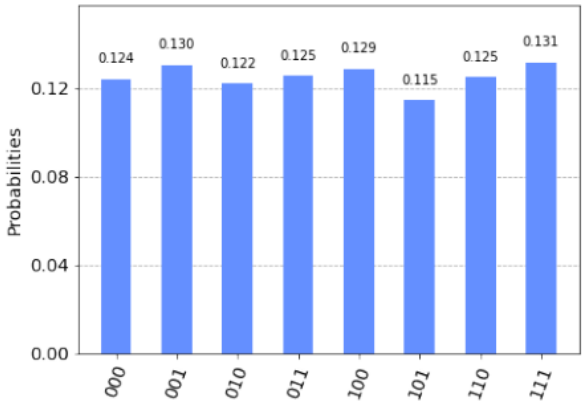


Figure 2.1: Intercept-and-resend-attack (QSS)

This graph shows how the average error (due to interception) changes with the number of iterations. The average error we got was 50.0666% with a variance of 1.2960.

Here is one of the histogram



Entangle-and-measure attack

At first, the GHZ triplet will be created according to the protocol. Then to create entanglement for Eve we used a controlled-NOT gate controlled by Bob and Charlie's qubit and target as two different ancilla qubits.

We ran the protocol in 'qasm_simulator' 500 times and generates following graph:

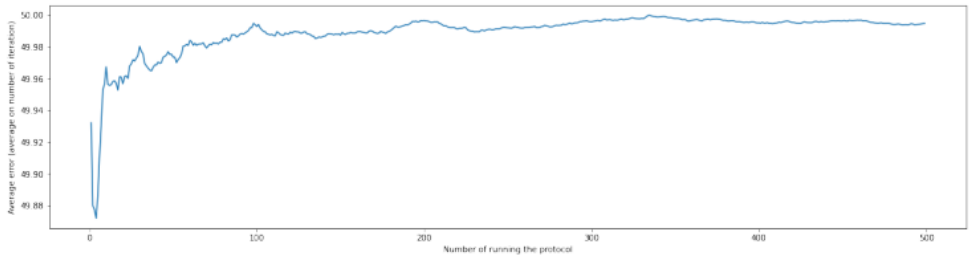
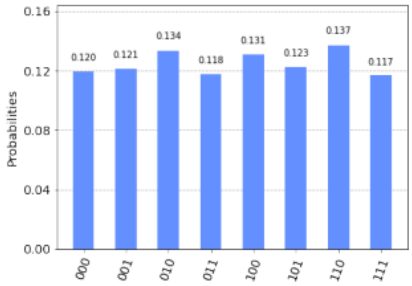


Figure 2.2: Entangle-and-measure-attack (QSS)

This graph shows how the average error (due to entanglement) changes with the number of iterations. The average error we got was 49.9950% with a variance of 0.0100.

Here is one of the histogram



DoS attack

After creating the GHZ triplet some eavesdropper applies an identity operator or some random unitary operator on the qubits both with the probability of 0.5. Then the remaining part of the protocol will be continued. We know that any 2×2 complex unitary matrix can be represented as

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\phi}\sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)}\cos(\frac{\theta}{2}) \end{pmatrix}$$

where θ, ϕ, λ are three parameters. This concept is used in the above code.

We ran the protocol in 'qasm_simulator' 500 times and generates following graph:

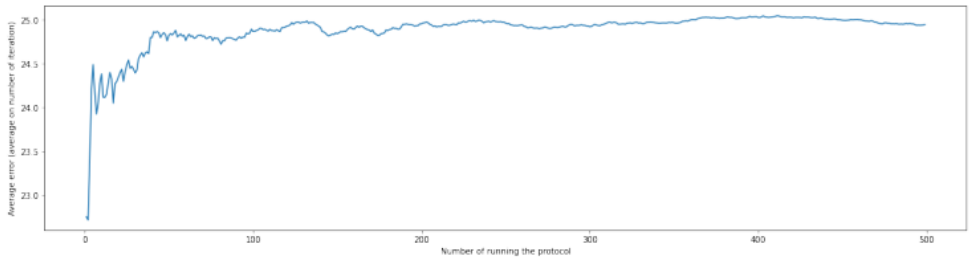
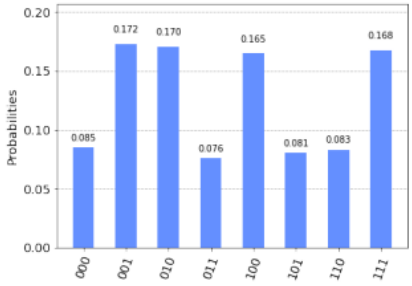


Figure 2.3: DoS attack (QSS)

This graph shows how the average error (due to attack) changes with several iterations. The average error we got was 24.9481% with a variance of 4.1115.

Here is one of the histogram



2.2.4 Execution on IBM backends

Choosing Initial Layout

In real backends maintained by IBM, not all qubits are connected with each other. There is some specific coupling map (see fig. 2.4 for 7-qubit backends). If we apply some 2-qubit gate between two qubits that are not connected then some swap gate will be applied to change the positions of the virtual qubits. So we have to avoid this type of 2-qubit gates. Also, the noise of the qubits, measurements as well as coupling between qubits is not the same. So we have to use those qubits and couplings which are less noisy. Roughly, in IBM backends a 2-qubit gate has a 10% error of measurement, and a single-qubit gate has a 1% error of measurement. So we will focus on those qubits which have less measurement error and those coupling which has less 'CX' error (controlled-NOT gate error). Also, we will try to avoid swap gates. All of these can be done by choosing a proper initial layout for our circuit. The initial layout is a vector denoting a one-one map between virtual qubits in our circuit and physical qubit in the backend. At the time of transpilation, this vector is used to select the physical qubits from the backend where the transpiled circuit will be assembled. Here we used the initial layout as $[5, 3, 6]$ for 'ibmq_jakarta'

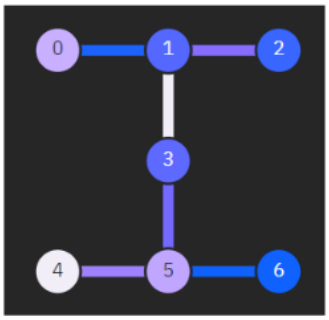


Figure 2.4: Layout of IBM backend 'ibmq_jakarta' [Circle denotes qubits, line denotes coupling, color denotes error; white: high error, blue: less error.]

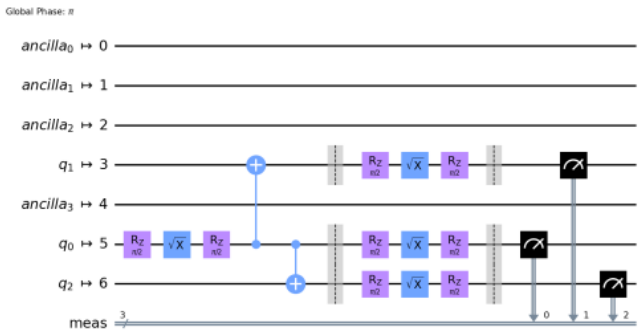
which means Alice, Bob, and Charlie's qubit will be mapped to qubit $5, 3, 6$ in the backend respectively.

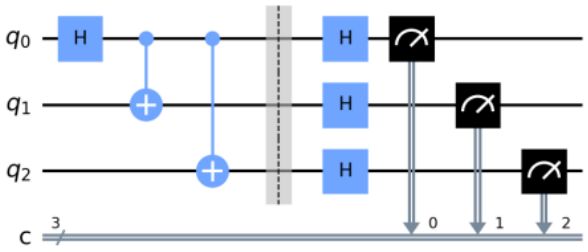
Execution

We executed the protocol on four 7-qubit IBM backends: 'ibm_perth', 'ibmq_lagos', 'ibmq_casablanca', 'ibmq_jakarta'. Error statistics for 100 iterations on above backends are as follows:

	ibm_perth	**ibmq_lagos**	**ibmq_casablanca**	**ibmq_jakarta**
mean error	0.2397	0.0649	0.0979	0.1152
variance of error	0.5998	1.2369	0.4330	1.2198

Here is the transpiled (due to backend configuration) circuit (left) [original circuit is given in right as reference]:





The histograms from four backends are:

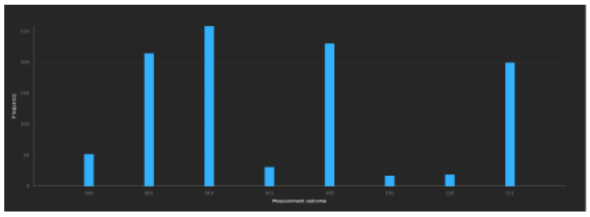


Figure 2.5: 'ibm_perth' histogram

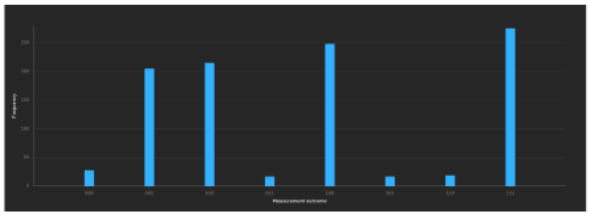


Figure 2.6: 'ibmq_lagos' histogram

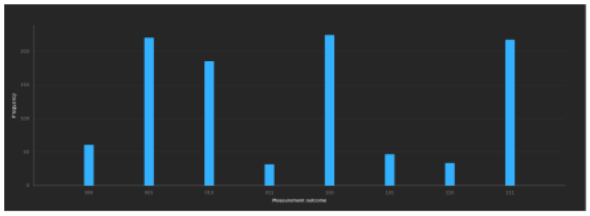


Figure 2.7: 'ibmq_casablanca' histogram

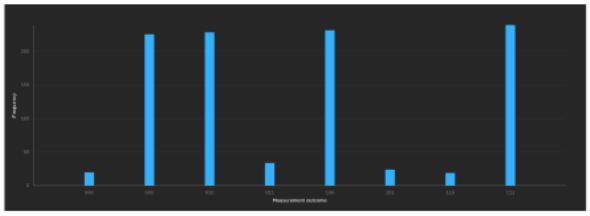


Figure 2.8: 'ibmq_jakarta' histogram

Here is the graph how average error changes over iteration number in four backends:

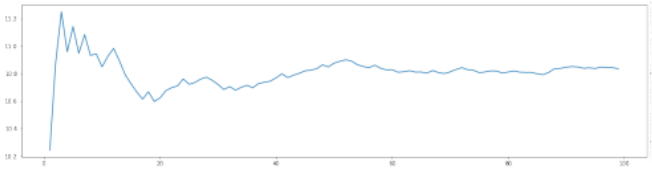


Figure 2.9: 'ibm_perth' graph

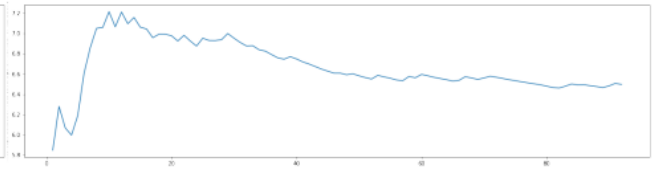


Figure 2.10: 'ibmq_lagos' graph

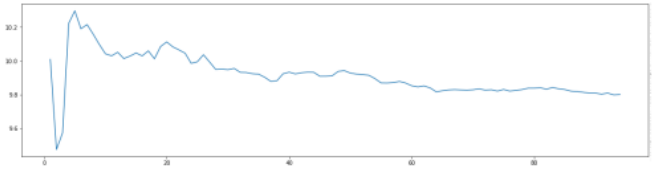


Figure 2.11: 'ibmq_casablanca' graph

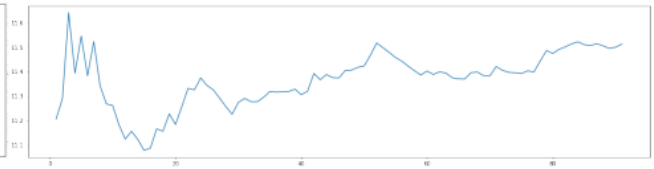


Figure 2.12: 'ibmq_jakarta' graph

2.2.5 Simmulation with Noise Models

Thermal Relaxation error is the most realistic error to create a realistic noise model. But thermal relaxation error is a coherent error, whereas a real machine generates incoherent errors also. So in our model, we will use thermal relaxation error followed by depolarising error (which is an incoherent error) as suggested in qiskit documentation [7]. So let us deduce depolarizing error parameters in presence of Thermal Relaxation error.

Denoting depolarizing channel as $E\_dep = (1-p) * I + p * D$, where I is the identity channel and D is the completely depolarizing channel, we have

$$\begin{aligned} 1 - error &= F(E\_dep * E\_relax) = (1-\lambda) * F(I * E\_relax) + \lambda * F(D * E\_relax) \\ &= (1-\lambda) * F(E\_relax) + \lambda * F(D) \\ &= F(E\_relax) - \lambda * (dim * F(E\_relax) - 1)/dim \end{aligned}$$

where F denotes the average fidelity.

Thus we get,

$$depol\_param = dim * \frac{error - relax\_infid}{dim * relax\_fid - 1} \tag{2.1}$$

where

$depol\_param =$ depolarizing error parameter λ
$dim =$ dimension
$error =$ gate error (from backend)
$relax\_infid =$ gate infidelity due to thermal ralexation error
$relax\_fid = 1 - relax\_infid$

Thus we created a noise model using parameters from '*ibmq_jakarta*' and (2.1).

We executed our circuits with this noise model and got average error as 12.7945% whereas in real backend error was 11.52%. Here is the graph how average error changes with iteration number:

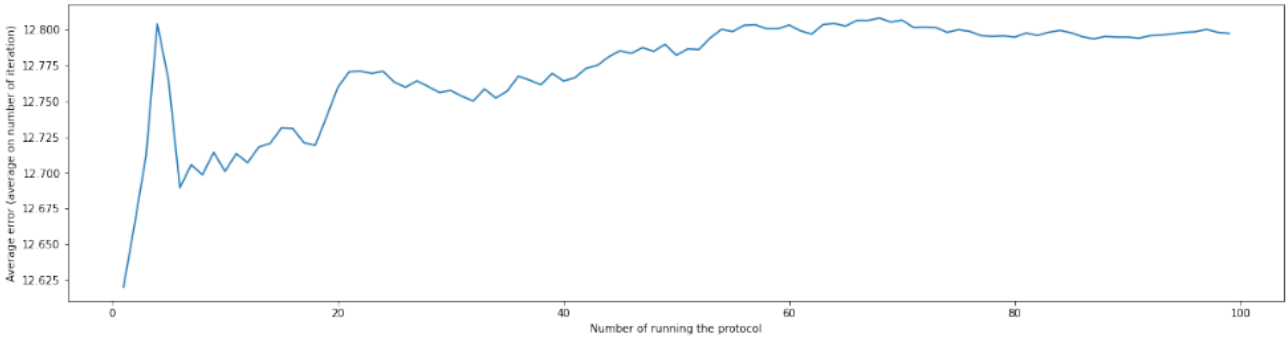
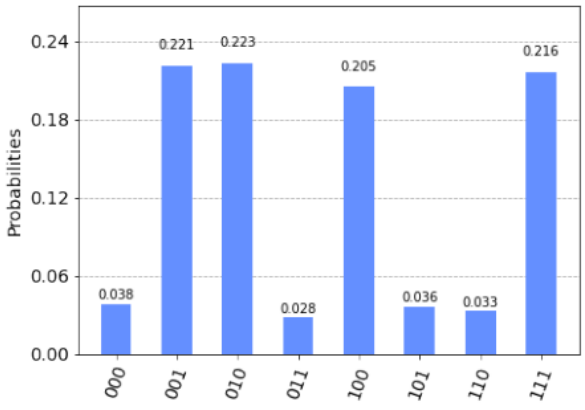


Figure 2.13: Noisy simulator graph (QSS)

Histogram for some random circuit is given by



2.2.6 Error Mitigation

Now our task is to mitigate the error. To do so we are focusing on measurement error mitigation and circuit error mitigation. Let us see the scaled circuits:

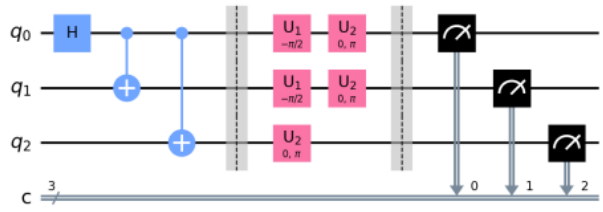


Figure 2.14: Without scalling

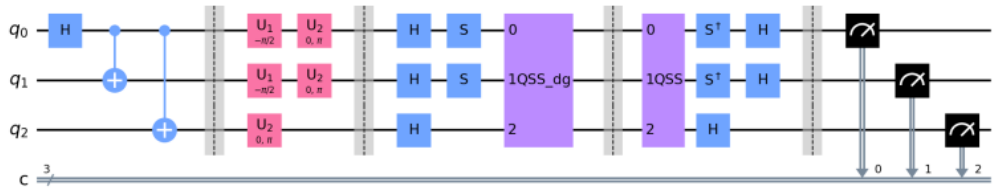


Figure 2.15: 3X scalling

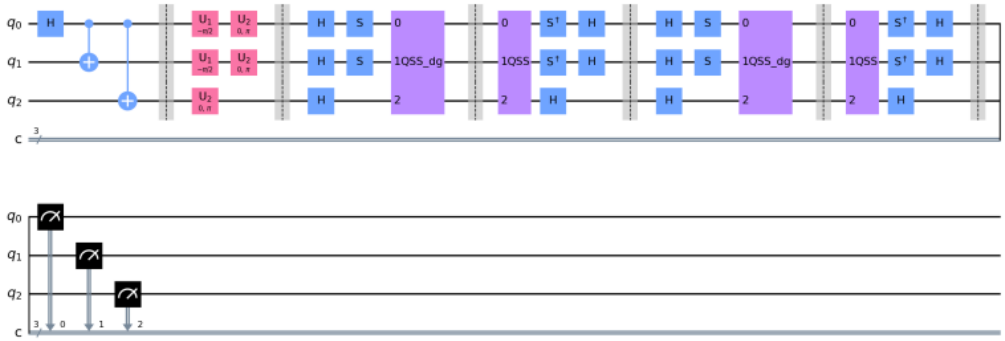


Figure 2.16: 5X scalling

On IBM Backend

We executed the protocol with the above error mitigation technique on 'ibmq_jakarta'.
Here is the graph how average error changes over iteration number (total 100 iteration):

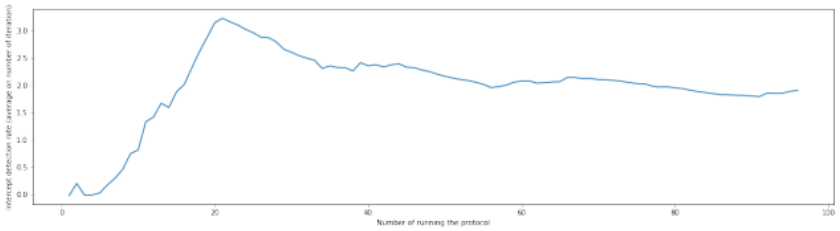


Figure 2.17: Backend mitigated error graph (QSS)

The average error is reduced from 11.5235 (without mitigation) to 1.9231% (with mitigation).
Histogram without mitigation and with mitigation:

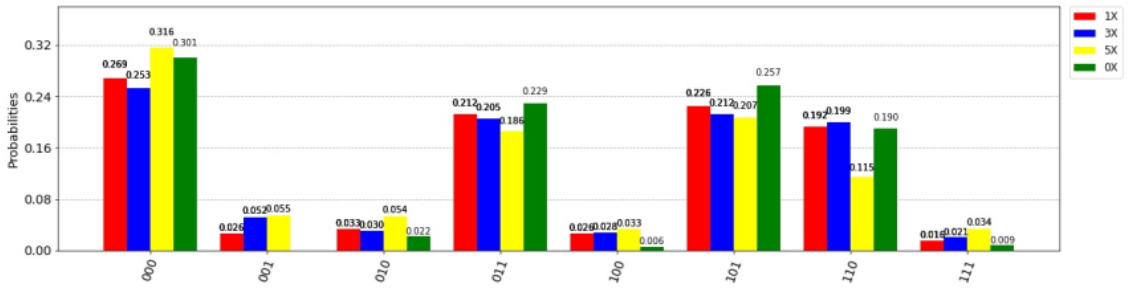


Figure 2.18: Backend error mitigation histogram (QSS)

On noisy simulator

We executed the protocol with the above error mitigation technique on a noisy simulator (see §2.2.5).
Here is the graph how average error changes over iteration number (total 100 iteration):

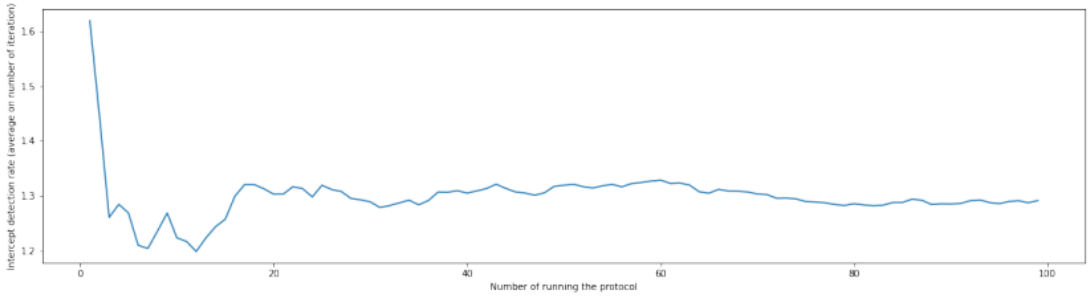


Figure 2.19: Noisy simulator mitigated error graph (QSS)

The average error is reduced from 12.7945% (without mitigation) to 1.2931% (with mitigation).
Histogram without mitigation and with mitigation:

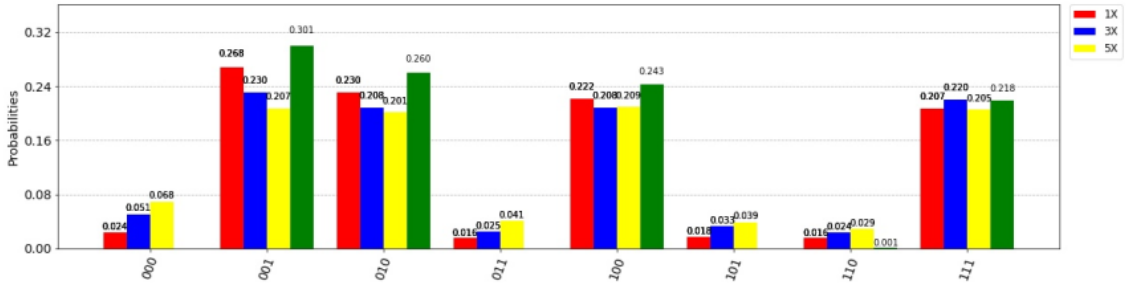


Figure 2.20: noisy simulator error mitigation histogram (QSS)

2.3 Quantum Information Splitting

This protocol is also proposed by Hillery et. al. [5] and uses the GHZ state. Three parties (one dealer, and two individual receivers) are involved here. Dealer has a secret quantum state $|\Psi\rangle$ to share. They will perform a modified version of teleportation protocol to construct the secret at any one individual's end.

2.3.1 Brief of the protocol

- **Step 1:** Alice, Bob, and Charlie each have one particle from a GHZ triplet (A, B, C respectively) that is in state

$$|\Phi^+\rangle_{ABC} = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

  Alice has one more particle (A' say) in some arbitrary quantum state $|\Psi\rangle_{A'} = \alpha|0\rangle + \beta|1\rangle$.

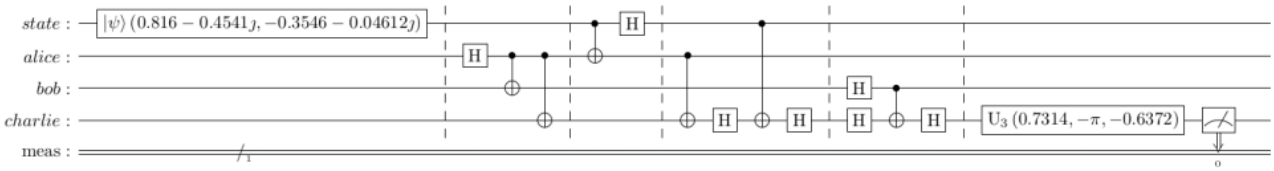
- **Step 2:** Alice measures her two qubits on a Bell basis and chooses Bob or Charlie randomly and declares the output. Say, Bob is chosen, then $|\Psi\rangle$ will be prepared in Charlie's qubit.

- **Step 3:** Charlie (not chosen by Alice) applies an X-gate depending on the measurement output of Alice's qubit A and a Z-gate depending on the measurement output of particle A' when the measurement outputs are one (similar to the teleportation protocol). Now Bob and Charlie produce the state $|\Psi\rangle$ together at Charlie's end.

- **Step 4:** Bob (chosen by Alice) measures his qubit on a Hadamard basis. Bob sends his measurement output to Charlie. Charlie applies a Z-gate depending on the measurement output of Bob when the measurement output is one. Now Charlie has the state $|\Psi\rangle$.

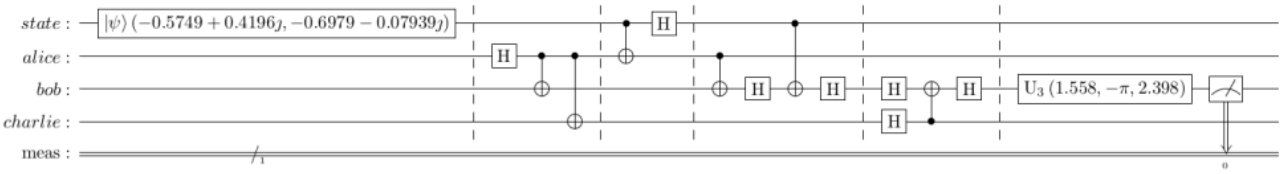
2.3.2 Implementation with Qiskit

Here is the circuit when Alice chooses Bob at step 2: In the first part we are initializing A' and in the last part to verify



whether $|\Psi\rangle$ is created correctly or not at Charlie's end we applied the inverse of the gate we used to initialize A'. Measurement output 0 with probability 1 means the protocol works correctly.

If Alice chooses Charlie at step 2 then the circuit will be as following:



2.3.3 Attack Implementation

Intercept-and-resend attack

We ran the protocol 600 times and got the following plot of interception detection rate:

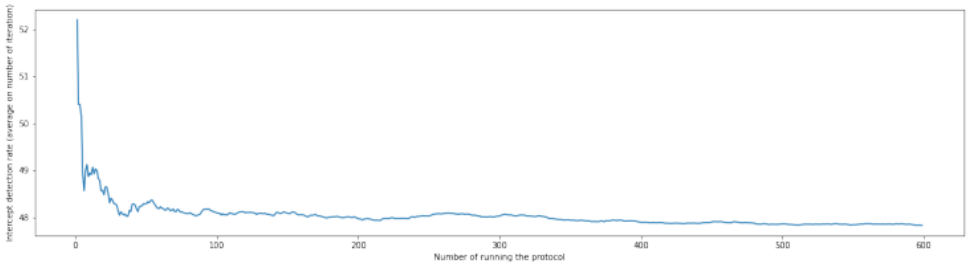
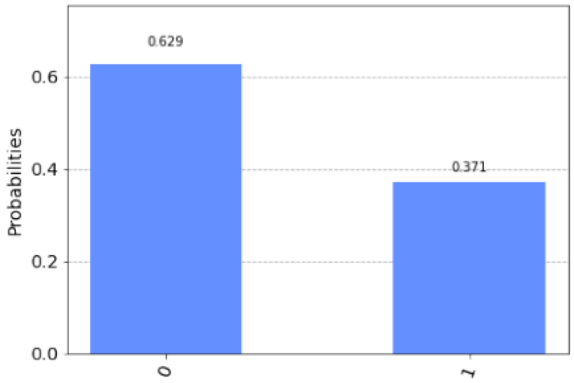


Figure 2.21: Interception-and-resend-attack graph (QIS)

This graph shows how the average error (due to interception) changes with the number of iterations. The average error we got was 47.832%.

Here is histogram for one of the circuits



Entangle-and-measure attack

We ran the protocol in 'qasm_simulator' 500 times and generates following graph:

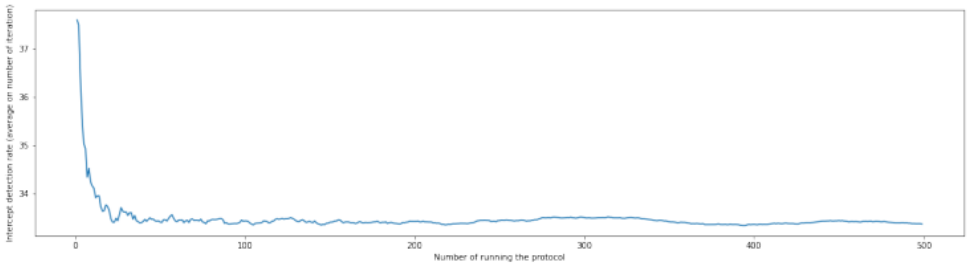


Figure 2.22: Entangle-and-measure-attack graph (QIS)

This graph shows how the average error (due to entanglement) changes with the number of iterations. The average error we got was 33.376%.

DoS attack

We ran the protocol in 'qasm_simulator' 600 times and generates following graph:

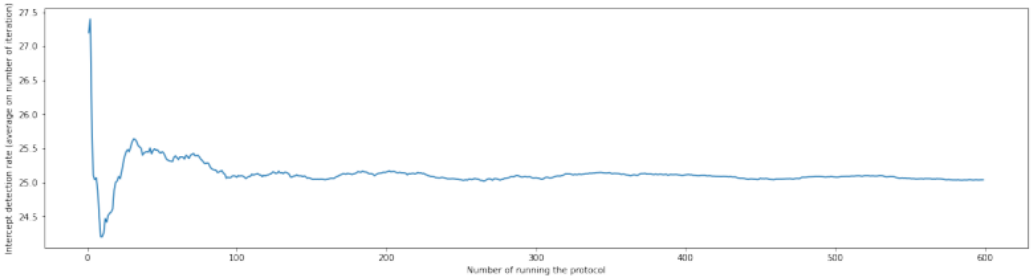


Figure 2.23: DoS attack graph (QIS)

This graph shows how the average error (due to attack) changes with several iterations. The average error we got was 25.037%.

2.3.4 Execution on IBM backends

We executed the protocol on 'ibmq_jakarta (1.0.25)' [backend version is inside the parenthesis]. Here is the graph how average error changes over iteration number (total 100 iteration):

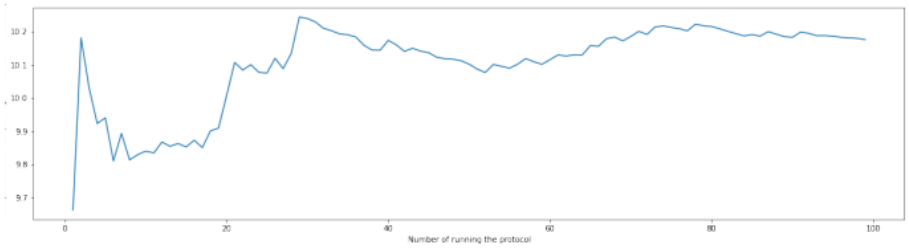
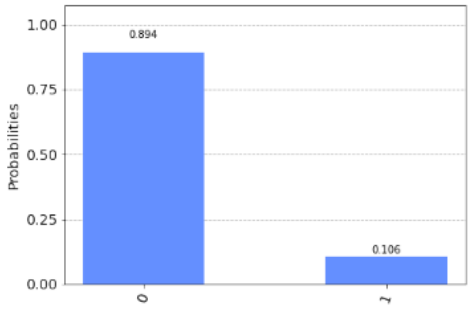


Figure 2.24: 'ibmq_jakarta' error graph (QIS)

The average error we received is 10.1752% with a variance of 0.5857.
Here is histogram for one circuit



2.3.5 Simmulation in Noise Model

The graph how average error changes over iteration number (total 100 iteration) on the noisy simulator generated in section §2.2.5:

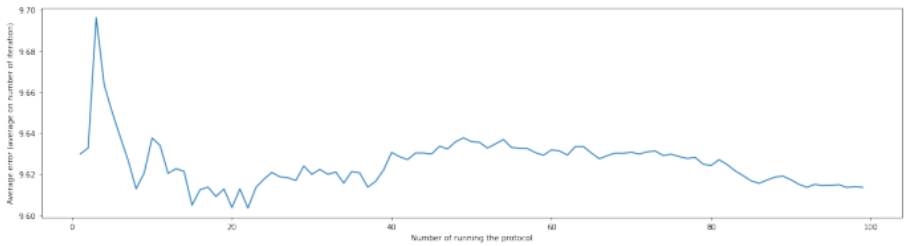
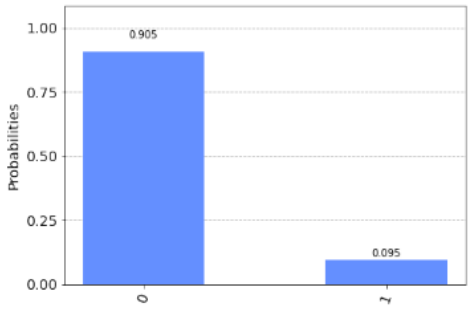


Figure 2.25: Noisy simulator error graph (QIS)

The average error we received is 9.6153% with a variance of 0.0175. Time is taken in this simulation (9800 circuits): 56 hours 46 minutes 18.695701122283936 seconds.

Here is histogram for one circuit



2.3.6 Error Mitigation

On IBM Backend

We executed the protocol with the above error mitigation technique on 'ibmq_jakarta (1.0.25)' [backend version is inside the parenthesis].

Here is the graph how average error changes over iteration number (total 100 iteration):

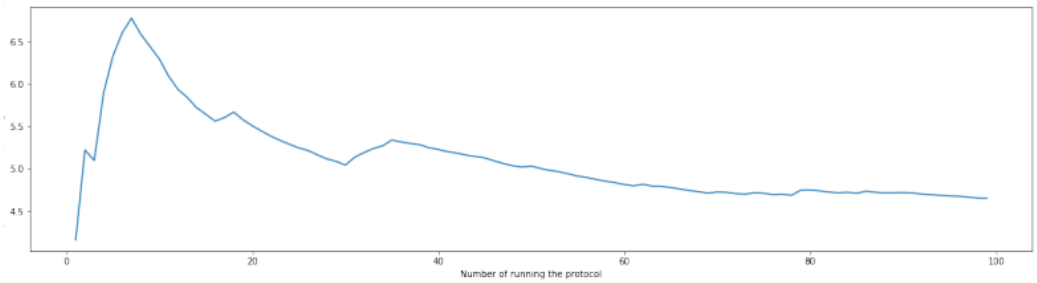
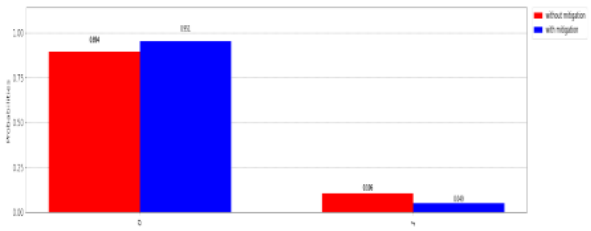


Figure 2.26: Backend mitigated error graph (QIS)

Average error is reduced from 10.1752 (without mitigation) to 4.6444% (with mitigation).

Histogram without mitigation and with mitigation:



On noisy simulator

We executed the protocol with the above error mitigation technique on a noisy simulator (see §2.2.5).
Here is the graph how average error changes over iteration number (total 125 iteration):

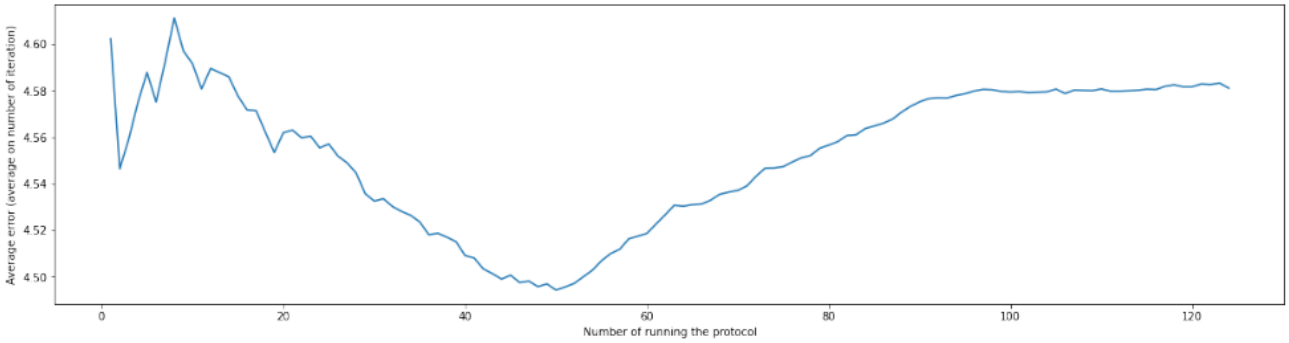
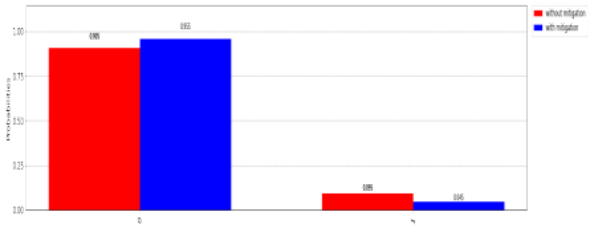


Figure 2.27: Noisy simulator mitigated error graph (QIS)

The average error is reduced from 9.6153% (without mitigation) to 4.5806% (with mitigation).
Histogram without mitigation and with mitigation:



2.4 Generalization of Quantum Secret Sharing

This is generalization to n-party of the protocol [5] discussed in 2.2. The generalization is given by Li Xiao et. al. [9] for n-party. The protocol is written briefly in the following section.

2.4.1 Brief of the Protocol

- **Step 1:** Alice, Bob, Charlie,... each has one particle from a GHZ multiplet that is in state

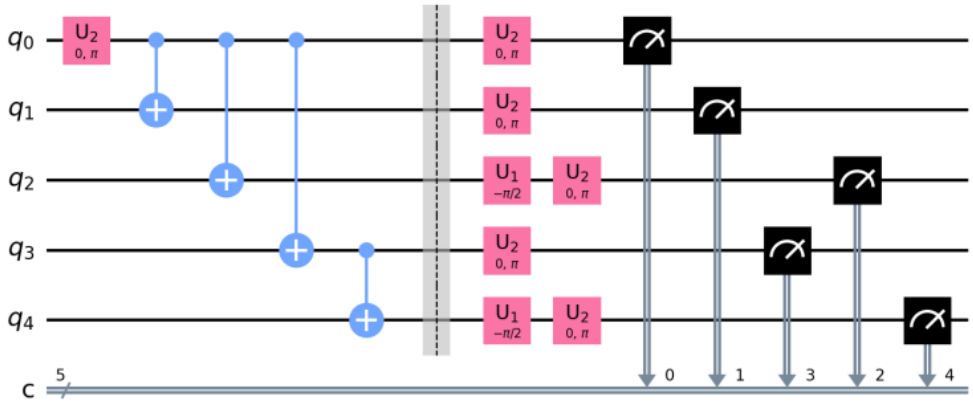
$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00...0\rangle + |11...1\rangle)$$

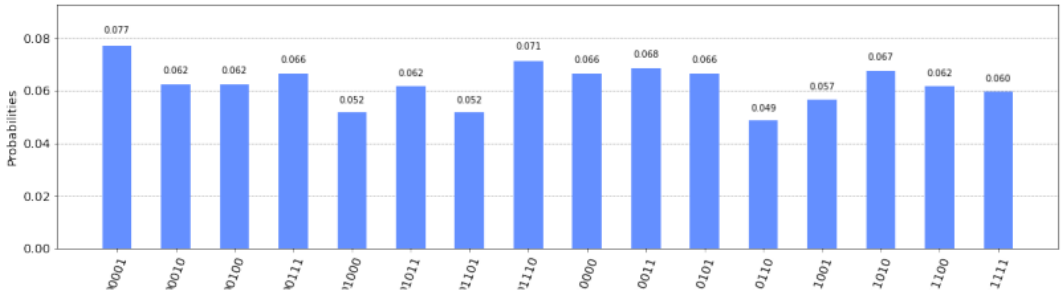
- **Step 2:** Everyone measures their qubits randomly either on x or y basis and declares their measurement basis.
- **Step 3:** If there are an odd number of measurements on y basis then discard the corresponding measurement result otherwise continue.
- **Step 4:** To get Alice's message back, the remaining n-1 person perform the following:
  - when the number of parties measures in y basis is $4k$ for some positive integer k then XOR of their (n-1 party) measurement results will give Alice's message.
  - when the number of parties measures in y basis is $4k+2$ for some non-negative integer k then the complement of the XOR of the measurement results will give Alice's message.

.

2.4.2 Implementation with Qiskit

Here is one of the circuit and corresponding histogram for 5 (delear + 4 individual):





Here q_0, q_1 and q_3 chosen x axis and other two chosen y axis. The histogram shows possible outcomes. To find the number of total possible outcomes let's consider $q_1, ..., q_4$. Each of them will get two outcomes (up and down) on their respective basis. So there are 2^4 possible outcomes. But for each of these outcomes, q_0 will collapse in one specific outcome - either up or down depending on others' outcomes. So total number of possible outcomes will be $2^4 \times 1 = 16$. You can see there are 16 bars in the histogram.

2.4.3 Attack

Following table shows percentage of detecting attacks for different number of parties:

	Intercept-and-resend attack	Entangle-and-Measure attack	DoS attack
4-qubit	50.0013%	50.0001%	24.9993%
5-qubit	49.8162%	50.0153%	24.8501%
6-qubit	49.9836%	49.8903%	25.0109%
7-qubit	49.9962%	49.9963%	25.0010%

In each case, we can detect the intercept-resend attack and entangle-measure attack with approximately .5 probability while the DoS attack is detected with a probability of .25 approximately.

2.4.4 Real Backend and Noisy simulator

Following table shows percentage of error recieved from 'ibmq_jakarta' and 'noisy simulator' (see §2.2.5):

	4-qubit	5-qubit	6-qubit	7-qubit
ibmq_jakarta	14.9116%	30.1990%	41.2327%	53.8105%
noisy simulator	15.9576%	19.6508%	23.4379%	27.0475%

From this table, you can see that the error difference between the real backend and noisy simulator is increasing from the 5-qubit implementation. This is happening because we only considered gate noise, but in the backend, there are errors due to the environment and also the qubits are noisy. Also, note that when the number of qubits is 5 or more then an error is very high. In the next section, we will see that after mitigating the error 5-qubit protocol will have less than 10% error.

2.4.5 Error Mitigation

Following table shows percentage of error recieved from 'ibmq_jakarta' and 'noisy simulator' (see §2.2.5):

	4-qubit	5-qubit
ibmq_jakarta	5.2366%	7.3901%
noisy simulator	2.4680%	4.9805%

Chapter 3

Quantum Secure Direct Communication

3.1 Introduction

Quantum secure direct communication (QSDC) and quantum key distribution (QKD) are quantum-based communication schemes. BB84, which is one of the first QKD protocols, was proposed in 1984 [**?**]. In QKD, first, a key is generated, then the message is encrypted using that key and transmitted. In contrast to QKD, QSDC sends a secret message directly via quantum channel without setting up any prior key [21, 22]. If both sides' communication is allowed in a QSDC then it is called quantum dialogue (QD) protocol.

A measurement-device independent quantum dialogue (MDI-QD) protocol was proposed by A. Maitra [20] in 2017. In 2020 Das and Paul [19] improved that protocol. In 2021, Das and Paul [23] proposed a quantum conference protocol which is an extended version of the above protocol. Here we implemented these protocols with IBM qiskit. Then we checked their security against different attacks. Also, we simulated them in noise model (see §2.2.5). Then we applied error mitigation on IBM Backend result as well as noisy simulator result.

Here we implemented the MDI-QD protocol from [19, 20] and the conference protocol proposed by Das and Paul in [23] for quantum direct communication in qiskit. And then checked their security against different attacks. We implemented these algorithms with IBM qiskit, and executed them in the ideal simulator, real IBM backends, and different noisy simulators. Also, we created a noise model using parameters from 'ibmq_jakarta' and executed protocols in using these noise models also. We are working on error mitigation for real backend and noisy simulators.

3.2 Measurement-Device Independent Quantum Dialogue

There are three algorithms (one in [20] and two modifications in [19]) with some classical modifications. The quantum part is the same for all three algorithms. So the qiskit implementation, security check against quantum attacks, and error due to noise will be the same for all of the three algorithms. We will first see all of the three algorithms briefly then will continue with only one. Here algorithm 1 is proposed by A. Maitra [20] and algorithms 2, and 3 are modifications of algorithm 1 provided in [19].

3.2.1 Brief of the protocols

Algorithm 1

1. Alice and Bob share n bit key string $k = k_1 k_2 \dots k_n$.
2. They encode their messages in Q_A and Q_B using table 3.1.

Key	Message bit	Encoded qubit
0	0	$\|0\rangle$
0	1	$\|1\rangle$
1	0	$\|+\rangle$
1	1	$\|-\rangle$

Table 3.1: Encoding Table (MDI-QD)

3. Send Q_A and Q_B to some third party who will measure them on Bell basis and announce the results.
4. Alice and Bob guess each others message bits using table 3.2.

Self qubit	Guess when measurement output is			
	$\|\Phi^+\rangle$	$\|\Phi^-\rangle$	$\|\Psi^+\rangle$	$\|\Psi^-\rangle$
$\|0\rangle$	0	0	1	1
$\|1\rangle$	1	1	0	0
$\|+\rangle$	0	1	0	1
$\|-\rangle$	1	0	1	0

Table 3.2: Guess each other's message bit (MDI-QD)

5. They discard the cases where the measurement outcome is $|\Phi^+\rangle$ or $|\Psi^-\rangle$.
6. Alice and Bob randomly choose $\frac{\gamma n}{2}(\gamma < 1)$ number of positions and announce the guesses for respective positions to estimate error.
7. If an error is tolerable then they continue with remaining (not chosen above) bits. Otherwise, they abort.

Algorithm 2

1. Alice and Bob share n bit key string $k = k_1 k_2 \dots k_n$ and calculate $c = \oplus_{i=1}^{n} k_i$.
2. They encode their codes in Q_A and Q_B using table 3.1.
3. Send Q_A and Q_B to some third party who will measure them on Bell basis and announce the result. Alice and Bob store the result.
4. Alice and Bob randomly choose $\gamma n(\gamma < 1)$ number of measurement results.
5. They guess each other's message bits (using table 3.2) for the above-chosen bits and announce guesses to estimate error.
6. If error is tolerable then they continue with remaining (not chosen above) bits. Otherwise, they abort. Let $\{M_{new}[i]\}_{i=1}^{n'}$ be the remaining sequence of measurement results and updated key be $k_{new} = k_1 k_2 \dots k_{n'}$, where $n' = (1-\gamma)n$.
7. Generate $X = \{X[i]\}_{i=1}^{n'}$ as
$$X[i] = \begin{cases} 1, & \text{if } M_{new}[i] = |\Phi^-\rangle \text{ or } |\Psi^+\rangle; \\ 0, & \text{otherwise} \end{cases}$$
8. Generate $Y = \{Y[i]\}_{i=1}^{n'}$ as
$$Y[i] = \begin{cases} 0, & \text{if } X[i] = 1; \\ k_{new}[j], & \text{if } c = 1 \text{ an } X[i] \text{ is the } j^{th} \text{ zero in the sequence } X; \\ \overline{k_{new}[j]}, & \text{if } c = 0 \text{ an } X[i] \text{ is the } j^{th} \text{ zero in the sequence } X \end{cases}$$
9. If $X[i] \oplus Y[i] = 1$ then Alice and Bob consider i^{th} measurement result $M_{new}[i]$ and guess other's message bit using table 3.2. Else they discard $M_{new}[i]$.

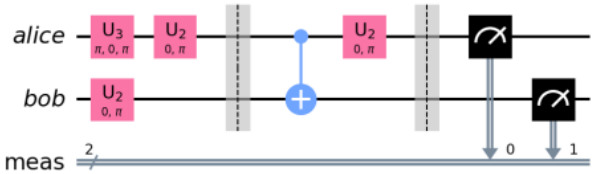
Algorithm 3

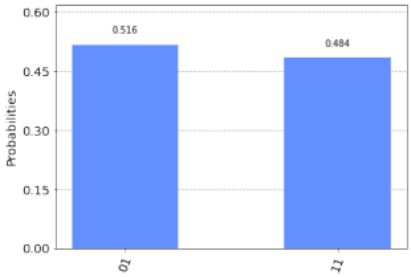
1. Perform first 7 steps of algorithm 2.
2. Generate $Y = \{Y[i]\}_{i=1}^{n'}$ and $Z = \{Z[i]\}_{i=1}^{n'}$ as
$$Y[i] = \begin{cases} 0, & \text{if } X[i] = 1; \\ k_{new}[j], & \text{if } c = 1 \text{ an } X[i] \text{ is the } j^{th} \text{ zero in the sequence } X; \\ \overline{k_{new}[j]}, & \text{if } c = 0 \text{ an } X[i] \text{ is the } j^{th} \text{ zero in the sequence } X \end{cases}$$
$$Z[i] = \begin{cases} 0, & \text{if } X[i] = 1; \\ k_{new}[j], & \text{if } c = 0 \text{ an } X[i] \text{ is the } j^{th} \text{ zero in the sequence } X; \\ \overline{k_{new}[j]}, & \text{if } c = 1 \text{ an } X[i] \text{ is the } j^{th} \text{ zero in the sequence } X \end{cases}$$

3. If $X[i] \oplus Y[i] = 1$ then Alice and Bob consider i^{th} measurement result $M_{new}[i]$ and Bob guess Alice's message bit using table 2 above. Else they discard $M_{new}[i]$.
4. If $X[i] \oplus Z[i] = 1$ then Alice and Bob consider i^{th} measurement result $M_{new}[i]$ and Alice guess Bob's message bit using table 1 above. Else they discard $M_{new}[i]$.

3.2.2 Implementation with Qiskit

We implemented above protocol with IBM qiskit. Here is one of the circuits and corresponding histogram:





$U_3(\pi, 0, \pi)$ is the X gate and $U_2(0, \pi)$ is H hate. Here the key is 1, Alice's message bit is also 1 but Bob's message bit is 0. So the qubits are prepared in $|-\rangle$ (Alice) and in $|+\rangle$ (Bob) state. Let us now see some measurement output we got and compare them with the expected output. Here is a table containing the first few inputs and corresponding output we got:

key	Alice	Bob	Enc_basis	Expected	measurement
1	1	0	x	[[0, 1], [1, 1]]	[1, 1]
0	0	0	z	[[0, 0], [0, 1]]	[0, 1]
0	1	0	z	[[1, 0], [1, 1]]	[1, 0]
0	1	0	z	[[1, 0], [1, 1]]	[1, 0]
0	1	1	z	[[0, 0], [0, 1]]	[0, 0]
0	1	0	z	[[1, 0], [1, 1]]	[1, 0]
1	1	1	x	[[0, 0], [1, 0]]	[0, 0]
1	1	0	x	[[0, 1], [1, 1]]	[0, 1]
0	1	0	z	[[1, 0], [1, 1]]	[1, 1]
0	1	0	z	[[1, 0], [1, 1]]	[1, 1]
1	1	0	x	[[0, 1], [1, 1]]	[1, 1]
0	0	0	z	[[0, 0], [0, 1]]	[0, 0]
0	1	1	z	[[0, 0], [0, 1]]	[0, 0]
1	0	1	x	[[0, 1], [1, 1]]	[1, 1]
1	0	0	x	[[0, 0], [1, 0]]	[1, 0]

Table 3.3: Output table (MDI-QD)

Let us take any two (here red and green marked row is taken) to analyze measurement output. In the red row key is zero, so Alice's bit 1 is encoded as $|1\rangle$ and Bob's bit 0 encoded as $|0\rangle$. Now from table 3.1, we can see that if the key is 0 and Alice's bit is 1 then Bob's bit 0 forces the quantum state to be either $|\Psi^+\rangle$ or $|\Psi^-\rangle$ after Bell measurement. So measurement output will be 10 or 11 with half probability. Here we got 11 which is one of the desired outputs.

Now let us consider the green marked row. Here the key is one, Alice's bit is 0 and Bob's bit is 1. So Alice's prepared qubit will be $|+\rangle$ and Bob's qubit will be $|-\rangle$. Then after measurement state should be either $|\Phi^-\rangle$ or $|\Psi^-\rangle$ with half probability. So measurement output should be 01 or 11. Here we got 11 which is one of the desired outputs.

3.2.3 Attack Implementation

Intercept-and-resend attack

We ran the protocol 500 times and got the following plot of interception detection rate:

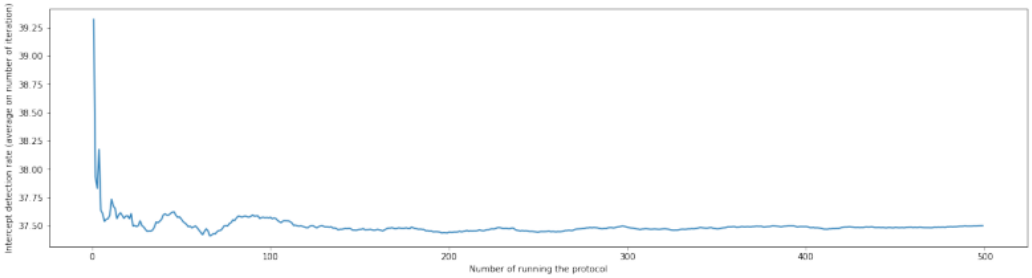
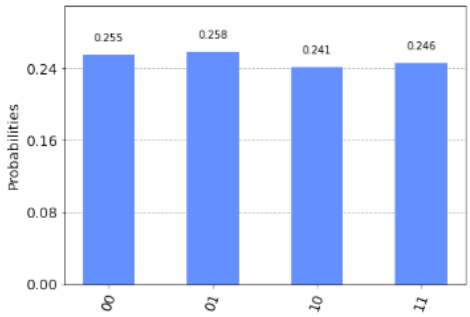


Figure 3.1: Intercept-and-resend attack graph (MDI-QD))

This graph shows the average error (due to interception) against the number of iterations. We average error as 37.5016% with a variance of 0.8824.

Here is histogram for one of the circuits



Entangle-and-measure attack

We ran the protocol in 'qasm_simulator' 500 times and generates following graph:

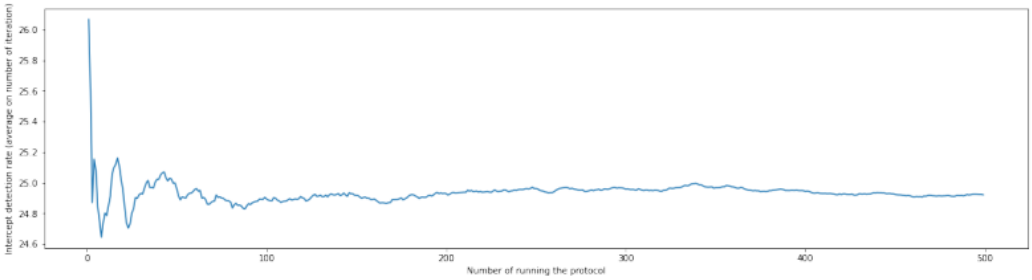


Figure 3.2: Entangle-and-measure attack graph (MDI-QD))

This graph shows the average error (due to interception) against the number of iterations. We average error as 24.9238% with a variance of 1.3435.

DoS attack

We ran the protocol in 'qasm_simulator' 500 times and generates following graph:

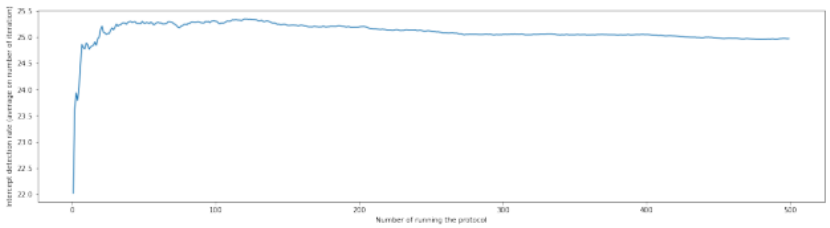


Figure 3.3: DoS attack graph (MDI-QD))

This graph shows the average error (due to interception) against the number of iterations. We average error as 24.9735% with a variance of 1.5312.

3.2.4 Execution on IBM backends

We executed the protocol on four 7-qubit IBM backends: 'ibm_perth (1.1.14)', 'ibmq_casablanca (1.2.49)', 'ibmq_jakarta (1.0.25)' [backend versions are inside the parenthesis]. Error statistics for 100 iterations on above backends are as follows:

	ibm_perth	**ibmq_casablanca**	**ibmq_jakarta**
mean error	0.0457	0.0582	0.0467
variance of error	0.2567	0.2686	3.2533

Here is the graph how average error changes over iteration number in 'ibmq_casablanca':

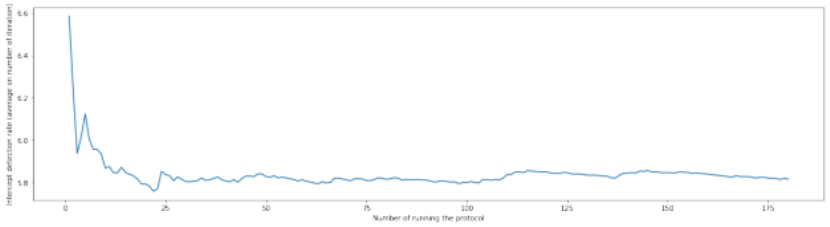


Figure 3.4: 'ibmq_casablanca' error graph (MDI-QD))

3.2.5 Simmulation in Noise Model

The graph how average error changes over iteration number (total 100 iteration) on the noisy simulator generated in section §2.2.5:

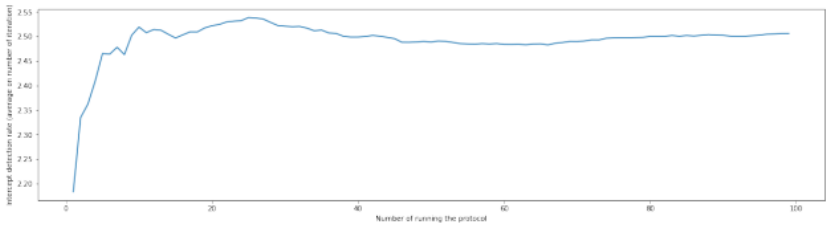


Figure 3.5: noisy simulator error graph (MDI-QD)

The average error we received is 9.6153% with a variance of 0.0175.

3.3 Quantum Conferrence

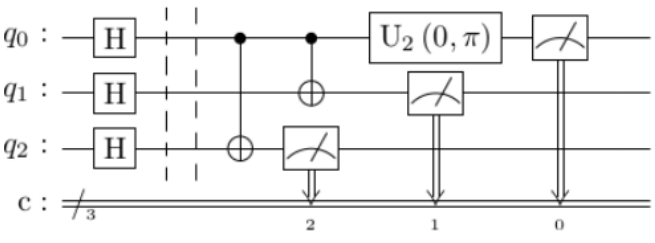
Using a similar approach as proposed in [19], the author proposed [23] a three-party quantum conference protocol with the help of an untrusted fourth party. Let us see the proposed protocol briefly in the following section.

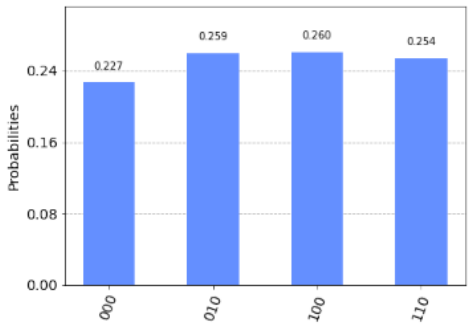
3.3.1 Brief of the protocol

1. Alice, Bob and Charlie share m bit key string $k = k_1k_2 \dots k_m$. Let the m-bit messages of Alice, Bob and Charlie be $a = a_1a_2 \dots a_m, b = b_1b_2 \dots b_m, c = c_1c_2 \dots c_m$ respectively.
2. They encode their messages in Q_A, Q_B, and Q_C using table 3.1.
3. Alice, Bob, and Charlie choose some random permutation and apply those on Q_A, Q_B, Q_C and get q_A, q_B, q_C respectively.
4. Send Q_A, Q_B, Q_C to some fourth party (UFP).
5. Alice, Bob, and Charlie choose δm number of common positions on Q_A, Q_B, Q_C. where $\delta \ll 1$. For these selected positions they perform the following steps:
   - They tell the positions and bases of the corresponding qubits to UFP.
   - UFP measures those states on proper basis and announces the results.
   - Alice, Bob, and Charlie reveal corresponding qubits and compare with UFP's result.
   - If the estimated error is not tolerable then they abort, otherwise, they continue.
6. Alice, Bob and Charlie tell the permutations they applied to UFP. UFP applies inverse permutation to get back Q_A, Q_B, Q_C.
7. They discard the qubits which are choosen above and discard corresponding key values.
8. UFP measures each three-qubit state (one qubit from each of Q_A, Q_B, Q_C) in $\mathcal{B}_3$ basis and announces the result, where $\mathcal{B}_3 = \{|\Phi_i^{\pm}\rangle\}_{i=0}^{3}$ with
$$|\Phi_0^{\pm}\rangle = \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle), \quad |\Phi_1^{\pm}\rangle = \frac{1}{\sqrt{2}}(|001\rangle \pm |110\rangle)$$
$$|\Phi_1^{\pm}\rangle = \frac{1}{\sqrt{2}}(|010\rangle \pm |101\rangle), \quad |\Phi_3^{\pm}\rangle = \frac{1}{\sqrt{2}}(|011\rangle \pm |100\rangle)$$
9. Alice, Bob and Charlie randomly choose $\gamma m'$ number of measurement results to estimate error, where $\gamma \ll 1$, $m' = (1-\delta)m$. If error is tolerable then they continue, otherwise abort.
10. They discard corresponding results, qubits and keys.
11. They will construct each other's message using remaining results.

3.3.2 Implementation with Qiskit

Here is one of the circuit and corresponding histogram:





Here the key value was 1 so everyone prepares their state in Hadamard basis and everyone has message 0 so prepared state is $|++\rangle$. After $\mathcal{B}_3$ measurement possible outcomes are $|\Phi_i^+\rangle$ for $i = 0, 1, 2, 3$. Histogram shows possible outcomes.

3.3.3 Attack

There are two channel: **channel I**, which they used to send Q_A, Q_B, Q_C to UFP; **channel II**, which they use at the last step to construct each other's messages. Table 3.4 and 3.5 show effect of attacks on these channels.

	Intercept-and-resend attack	**Entangle-and-Measure attack**	**DoS attack**
Channel I	24.8891%	25.0051%	24.9875%
Channel II	25.0730%	25.1845%	25.0860%

Table 3.4: Checking for channels

Attack	**Intercept-and-resend attack**	**Entangle-and-Measure attack**	**DoS attack**
Only channel I	50.0433%	25.0329%	50.0006%
Only channel II	12.4757%	12.4567%	12.5019%
Both channels	66.0339%	31.0565%	56.9548%

Table 3.5: Final message integrity checking

We can see from table 3.4 that attacks on different channels can be detected with probability .25. Also from table 3.5 one can see how much the message is affected by different attacks. The effect on message for attacks on channel II is approximately 12.5% and the effect of attacks on channel I is higher.

3.3.4 Real Backend and Noisy simulator

Following table shows percentage of error recieved from 'ibmq_jakarta' and 'noisy simulator' (see §2.2.5):

	Channel I	**Channel II**	**Message Integrity**
ibmq_jakarta	3.2427%	0.9808%	31.7487%
noisy simulator	2.2206%	1.2725%	30.0053%

From this table, we can see that the error of different channels is low but this low error affects messages with the probability of more than .3.

3.3.5 Error Mitigation

Following table shows percentage of error recieved from 'ibmq_jakarta' and 'noisy simulator' (see §2.2.5):

	Channel I	**Channel II**	**Message Integrity**
ibmq_jakarta	0.3679%	0.6288%	23.0308%
noisy simulator	0.1946%	0.2494%	20.4136%

Still, more than 20% of the message got corrupted. In this protocol small error in the channel is highly affecting message integrity.

Chapter 4

Conclusion

In this work, we saw that the protocols we implemented have some drawbacks in terms of implementation or security, or quantum noise. Both the QSS protocols proposed by Hillery [5] are very simple. Also, the generalization given by Xiao [9] is very simple. Although both the protocols proposed by Hillery [5] work well, the generalization doesn't perform well for more than four parties in IBM's real backend. Even error mitigation (ZNE) is not working perfectly for this protocol with six or more parties. We were not able to use Quantum error correction because that requires lots of qubits and we don't have access to backends with more than 7 qubits. For the MDI-QD protocol proposed by A. Maitra and modified by Das and Paul [20, 19], the probability of detecting attacks is around .25, which is low for security. Later Das and Paul modified the protocol to improve security and generalized the protocol [23] to get a conference protocol using two different channels. Security is improved but although both channels have very little noise, that affects message integrity on a large scale.

Bibliography

[1] Peter W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, Nov. 20–22, 1994.

[2] John Preskill, *Quantum Computing in the NISQ era and beyond*, Quantum 2, 79 (2018).

[3] Adi Shamir, *How to Share a Secret*, Communications of the ACM, Volume 22, Issue 11, November 1979.

[4] George Blakley, *Safeguarding Cryptographic Keys*, Managing Requirements Knowledge, International Workshop on (AFIPS). 48: 313–317, 1979.

[5] Mark Hillery, Vladimír Bužek and André Berthiaume, *Quantum Secret Sharing*, PHYSICAL REVIEW A, Volume 59, Issue 3, number 1829, march 1999.

[6] *Qiskit Textbook: Measurement Error Mitigation*.

[7] *Qiskit Documentation: Error from Backend*.

[8] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, W. J. Zeng, *Digital Zero Noise Extrapolation for Quantum Error Mitigation*, 2020 IEEE International Conference on Quantum Computing and Engineering, 306, October, 2020.

[9] Li Xiao, Gui Lu Long, Fu-Guo Deng, Jian-Wei Pan, *Efficient Multi-party Quantum Secret Sharing Schemes*, PHYSICAL REVIEW A, Volume 69, Issue 5, number 052307, May, 2004.

[10] Sreraman Muralidharan and Prasanta K. Panigrahi, *Perfect teleportation, quantum-state sharing, and superdense coding through a genuinely entangled five-qubit state*, PHYSICAL REVIEW A, Volume 77, Issue 3, number 032321, March 2008.

[11] Sreraman Muralidharan and Prasanta K. Panigrahi, *Quantum-information splitting using multi-partite cluster states.*, PHYSICAL REVIEW A, Volume 78, Issue 6, number 062333, 2008.

[12] Zhan-jun Zhang, Yong Li, and Zhong-xiao Man, *Multiparty quantum secret sharing*, PHYSICAL REVIEW A, Volume 71, Issue 4, number 044301, 2005.

[13] Guo-Ping Guo and Guang-Can Guo, *Quantum secret sharing without entanglement*, Physical Letters A, 310(4):247–251, 2003.

[14] Jung-Lun Hsu, Song-Kong Chong, Tzonelih Hwang, and Chia-Wei Tsai, *Dynamic quantum secret sharing*, Quantum information processing, 12(1):331–344, 2013.

[15] Ci-Hong Liao, Chun-Wei Yang, and Tzonelish Hwang, *Dynamic quantum secret sharing protocol based on ghz state*, Quantum information processing, 13(8):1907–1916, 2014.

[16] Dintomon Joy, M Sabir, Bikash K. Behera, Prasanta K. Panigrahi, *Implementation of quantum secret sharing and quantum binary voting protocol in the IBM quantum computer*, Quantum Information Processing 19, Article number: 33, 2020.

[17] I. D. K. Brown, S . Stepney, A . Sudbery, and S . L. Braunstein, *Searching for highly entangled multi-qubit states*, J. Phys. A: Math. Gen. 38, 1119 (2005).

[18] Bennett CH, Brassard G, *Quantum Cryptography: public key distribution and coin tossing*, Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India. IEEE, New York, USA, 1984, pp175-179.

[19] Nayana Das, Goutam Paul, *Two Efficient Measurement Device-Independent Quantum Dialogue Protocols*, International Journal of Quantum Information, Vol. 18, No. 07, 2050038 (2020), November, 2020.

[20] A. Maitra, *Measurement Device-Independent Quantum Dialogue*, Quantum Information Processing, Vol. 16, issue 12, Article 305, December, 2017.

[21] Long GL, Liu XS, *Theoretically efficient high-capacity quantum-key distribution scheme*, Phys Rev A 2002; 65: 032302.

[22] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Phys Rev Lett 1993; 70: 1895-1899.

[23] Nayana Das, Goutam Paul, *Secure Multi-Party Quantum Conference and Xor Computation*, Quantum Information and Computation, Vol.21, No.3-4, March 2021.

[24] Malcolm H. Levitt, *Spin Dynamics: Basics of Nuclear Magnetic Resonance*, 2nd edition, John Wiley & Sons, New York 2008, ISBN 0-470-51117-6, Section 11.9.2.

Implementation of Quantum Communication Protocols with IBM Qiskit [Checking Security of Protocols and Effect of Noise]

ORIGINALITY REPORT

5%
SIMILARITY INDEX

PRIMARY SOURCES

1	Nayana Das, Goutam Paul. "Two efficient measurement device independent quantum dialogue protocols", International Journal of Quantum Information, 2020 Crossref	128 words — 1%
2	qiskit.org Internet	90 words — 1%
3	library.isical.ac.in:8080 Internet	38 words — < 1%
4	iitk.ac.in Internet	36 words — < 1%
5	Chandrashekar, Adarsh. "Characterizing Noise in IBM-Q Devices Using Unitarity Randomized Benchmarking", Indian Statistical Institute - Kolkata ProQuest	33 words — < 1%
6	Xiaoqing Tan, Lianxia Jiang, Qiong Zhang. "Controlled Quantum Teleportation with Identity Authentication", 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, 2013 Crossref	28 words — < 1%

7	Masanori Ohya, I. Volovich. "Mathematical Foundations of Quantum Information and Computation and Its Applications to Nano- and Bio-systems", Springer Science and Business Media LLC, 2011 Crossref	20 words —	< 1%
8	Deng, F.G.. "An efficient quantum secret sharing scheme with Einstein-Podolsky-Rosen pairs", Physics Letters A, 20050606 Crossref	19 words —	< 1%
9	R. Ursin, F. Tiefenbacher, T. Jennewein, A. Zeilinger. "Anwendungen der Quantenkommunikation auf der Erde und im Weltraum", e & i Elektrotechnik und Informationstechnik, 2007 Crossref	16 words —	< 1%
10	Huawang Qin, Wallace K. S. Tang, Ray-Lin Tso. "Hierarchical quantum secret sharing based on special high-dimensional entangled state", IEEE Journal of Selected Topics in Quantum Electronics, 2020 Crossref	15 words —	< 1%
11	Mahn-Soo Choi. "A Quantum Computation Workbook", Springer Science and Business Media LLC, 2022 Crossref	15 words —	< 1%
12	ebin.pub Internet	15 words —	< 1%
13	export.arxiv.org Internet	15 words —	< 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 14 WORDS
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 14 WORDS