Error & Noise Analysis in a Quantum Key Exchange

Tashfeen, Ahmad
Oklahoma State University, Honours College
Computer Science Department
Oklahoma, Stillwater 74074
Email: tashfee@okstate.edu

***Abstract*—Over the course of human history the idea of secure communication has motivated different fields of science and one of them reside in both Computer Science and Mathematics. Oxford's English Dictionary defines cryptography as the art or practice of writing in code or cipher; the science of encryption. From the primitive shift ciphers during the time of Caesar to the Enigma in the second world war the race of creating an unbreakable code and then trying to break it continues. With the advent of Quantum Mechanics, interest in Quantum Computers has shown the so far theoretical consequences of a true parallel machine. Many classical ciphers relay on the difficulty of the underlying mathematics. RSA cryptosystem relies on the fact that for a large enough number, it is computationally in-feasible for a classical computer to calculate it's prime factors and Diffie Hellman public key exchange relies on the fact that search for a primitive root module large prime is also computationally improbable [1]. However, this is not the case for a theoretical quantum computer. In this paper, we shall look at a proposed quantum key exchange that takes benefits of the Quantum Mechanics itself and try to solve the unaddressed details of noise-error in the process of key exchange.**

I. Introduction

F.G. Deng et al., in their paper *Two-step quantum direct communication protocol using the Einstein-Podolsky-Rosen pair block* [2] have shown one possible way of exchanging a private key between two parties using quantum particles. Before we get started with the proposed Quantum Key Exchange, let's familiarize ourselves with Quantum Mechanics enough for the purposes of this paper. Quantum physics has become essential to understand the properties of solids, atoms, nuclei, sub-nuclear particles and light. A quantum particle is defined with the help of a probabilistic wave function (usually denoted using a Ψ). We will not go deeper into the mathematical rigour of how it works, but the general idea is that this function is a complex-valued, probability amplitude, which maps probabilities to the possible results of measurements made on an isolated quantum system. This function was the core of the debate behind the interpretation and completeness of quantum principles during 1930-1960. The residue of this debate still makes it hard to find a concrete interpretation of of quantum theory.

The heart of the debate lies in the way one resolves the EPR paradox. The acronym stands for the names of people who worked with what we now call EPR pairs; Einstein, Boris Podolsky, and Nathan Rosen. Not so much of a paradox any-more, but understanding the properties of the EPR pairs are important for the sake of this paper. An EPR pair is a pair of particles connected by the absurd property of quantum entanglement. Entanglement takes place when the two given particles of any EPR pair are naturally correlated such that any action on one particle shows a change in the other, as if they shared the same space [3]. Albeit, in reality they can be meters apart. Most papers call this transfer of information from one place to another as quantum teleportation. The bits of information transferred during such a teleportation are called qubits (quantum bits). I. Marcikic et al., showed that qubits (in form of EPR pairs) were transferred from one laboratory to another where the distance among the labs was $55meter$, and the connection was made using $2km$ of standard telecommunication fibre [4]. The EPR particles are what we define in terms of a wave-function as discussed above.

The importance of such a communication medium is not limited to just cryptography but is famous for other reasons as well. Albert Einstein refutes the possibility of transporting matter and energy from one place to another without it travelling by an intermediate locations [5], [6]. But, this is not true for teleportation in quantum states of EPR particles. This is done when only the structure is teleported and the matter stays with the sender while the information be already present at the receivers end. Einstein called this the "Spooky Action at a Distance".

J. Hoffstein et al., in their book *An Introduction to Mathematical Cryptography* [7], says that "We stand today on the brink of a revolution in cryptography." He goes on to defend his claim by showing the past relevance of cryptography by bringing up the employment statistics of National Security Agency. Which are that NSA is reputed to be the world's largest single employer of Ph.D.s in mathematics. However, post 1970s, there are now far more cryptographers employed in academia. He also mentions the United States government's treatment of cryptographic algorithms as munitions, to an extant where export of any such information was considered a treason.

With research focused towards the use of quantum computers to solve hard problems, it is equally legible to ask what are some of the hard quantum mechanics problems that could be used as a spine of a new cryptosystem whose class of difficulty meets the competition of the abilities of a theoretical

quantum computer. Such a system comes with it's details and intricacies. For this paper we I will analyse the problems of noise and the error caused by it, in a quantum network of communication. I will combine the naive empirical mathematics with machine learning's statistical models to demonstrate the relationship between desired accuracy, quantum bits necessary and present noise. This paper may help fill in some of the details in a usual Quantum Key Exchange protocol. August 26, 2015

II. Related Work

Where there are current cryptosystem secure against a quantum computer such as NTRU cryptosystem, Goldreich–Goldwasser–Halevi (GGH) lattice-based cryptosystem and Merkle–Hellman knapsack cryptosystem. They are all strictly reliant on the same principle of a hard mathematical problem. We now look at a cryptosystem based on quantum mechanics that uses the entanglement property of EPR pairs.

Charles H. Bennett et al., wrote one of the first papers describing how to use the quantum teleportation properties of EPR pairs in order to communicate information [8]. The paper called *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels* published on 29 March 1993, lays down the groundwork for all the research done afterwords on quantum communication. He shows the teleportation using correlated EPR pairs. The main idea is that the correlations assists in the "teleportation" of an intact quantum state from the sender to the receiver.

To talk about the simplest way of sending information about the quantum state of a particle, we'll call the sender "Alice", and "Bob" will be the receiver. Suppose Alice has a quantum system (a photon or spin-state particle), in state $|\phi\rangle$ unknown to her. The goal here is to communicate this state to Bob so he can make a copy of it. The knowledge of the the state vector of $|\phi\rangle$ will be sufficient. Although normally we don't know that ahead of time. If Alice knew the state vector $|\phi\rangle$ belongs to a given orthonormal set beforehand, then she can produce a copy of $|\phi\rangle$. The state vector with more than two (possible) nonorthogonal states can not be copied.

The simplest method for Alice to send Bob the state vector $|\phi\rangle$ would be to send the particle itself. Otherwise she can make it interact unitarily with another system, or "ancilla", whose initial state vector is known $|a_0\rangle$. The original particle is in a standard state $|\phi\rangle_0$ after the interaction. While the ancilla containing all the information about the state vector $|\phi\rangle$, in state $|a\rangle$. Alice can now send Bob the ancilla and Bob can reverse her manipulations to obtain a copy of $|\phi\rangle$. Park and James explain further about this "spin-exchange measurement" in [9]. They talk about how information can be swapped from one system to another, but it cannot be duplicated [10].

We saw that EPR can be used to exchange information over a distance and showed one way of doing it through spin-exchange measurement. Now we turn our attention to a secure communication, or in other-words a public key exchange. There are many papers that talk about a key exchange which are permutations of the usage of the general idea of EPR pairs' entanglement property. Few examples are BB84 protocol [11], Cabello protocol [12], B92 [13] and Einstein-Podolsky-Rosen scheme [14], [15].

F.G. Deng et al. [2], Long, Gui-Lu and Liu, Xiao-Shu [16] show a similar QKE scheme. Some notation would be helful before presenting the scheme, we can represent any EPR pair in one of the four Bell measurement states,

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad \text{State 00}$$
$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \qquad \text{State 01}$$
$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \qquad \text{State 10}$$
$$|\psi_4\rangle = \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle) \qquad \text{State 11}$$

As seen in the equations, Alice and Bob can decide on an encoding which in this case is, $(|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle, |\psi_4\rangle) = (00, 01, 10, 11)$. I will write an EPR pair as an ordered pair (x, y) for simplicity where x is the state of the first particle in EPR pair and y is the state of the second particle. For the sake of the key exchange, we will concern ourselves with an observation whose output will be binary. So, either particle in an EPR pair will output either 0 or 1 in case of this observation. Therefore, $x \in \{0, 1\}, y \in \{0, 1\}$. Notice the entanglement property implies that if an EPR pair was originally entangled as $(x, y) = (1, 0)$ and we observe y then y get's randomly reassigned to 1 or 0 where x get's negated as 0. Now the result of the observation and state x are same, namely 0. Similarly if the original entanglement was of the same states $(x, y) = (0, 0)$ then after observation on y, the other partner particle should be exactly the opposite. The key exchange happens as follows:

1) Alice produces a set of n EPR pairs originally in opposite states (or in the same states, the choice can be made either way but this effects how to conduct a check for an eavesdropper later). Let's denote this set as P and write,

$$P = \{n \in \mathbb{N} : (x, y)_0, \ldots, (x, y)_{n-1}\}$$

She then splits the pairs into two sets where one, P_x contains only x states and the other, P_y contains only y states.

$$P_x = \{n \in \mathbb{N} : x_0, \ldots, x_{n-1}\}$$
$$P_y = \{n \in \mathbb{N} : y_0, \ldots, x_{n-1}\}$$

Notice we originally entangled them so that,

$$P_x = \neg P_y$$

2) Alice sends P_y to Bob and Bob picks a sufficiently large subset of P_y and performs the binary measurement we discussed before on it. He records the result as a binary vector and shares it with Alice through a classical channel, say a telephone.

3) Alice now compares the result of Bob's measurements on some subset of P_y with the corresponding subset in P_x.
4) For some given error and how it correlates with the size of the original subset chosen in P_y, we can now tell if someone, say Eve, looked at P_y while it was sent to Bob [2]. If yes, then Alice never sends the set P_x or if the error is under the expected probability then she sends over the remaining P_x. This is the first eavesdropper check in the system.
5) After this we can make other checks to make the system secure against the man in the middle attacks as explained by F.G. Deng in [2] and G. L. Long in [16].

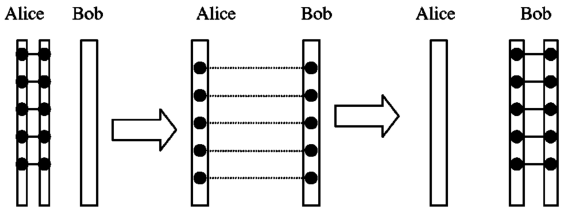

Fig. 1. QKE

In this scheme both [2] and [16] prove the security of the system and difficulties of actual implementation.

Riedmatten et al., discusses a similar issue with respect to the distance a qubit has to travel between Alice and Bob. Noisy detectors and lossy fibres decrease the signal to noise ratio (fidelity) with distance, and the maximal distance for a given fidelity is limited. Riedmatten's quantum communication is done through man in middle which they call Charlie. Alice's qubit is teleported to Bob by Charlie. Charlie measures the joint Bell state (BSM) of Alice's photon and one-half of the EPR pair, which projects Bob's photon into the state of Alice's photon. This way, the total distance travelled by the logical qubit is l from Alice to Bob but the effective distance covered by the photon to be detected by Bob is reduced to around $\frac{l}{3}$ [17]; maximising the distance for a given fidelitity. Riedmatten further shows the different fidelity plots for different quantum communication relays.

My paper will study that how does the size of the sufficient large subset of P_y look like. Also how can we come up with a secure size for a given error rate caused by noise in the network. Lastly we will try to implement various machine learning algorithms with features being error (only in certain subset of qubits) and noise rate that we know is present in our quantum network.

III. Experimental Results

We would like to know that what is the size of the sufficiently large subset for Bob to pick for a given probability of making an error. Let's take an example under the assumption of no noise. Let Alice pick for $n = 2$,

$$P = \{(0,1),(0,1)\}$$

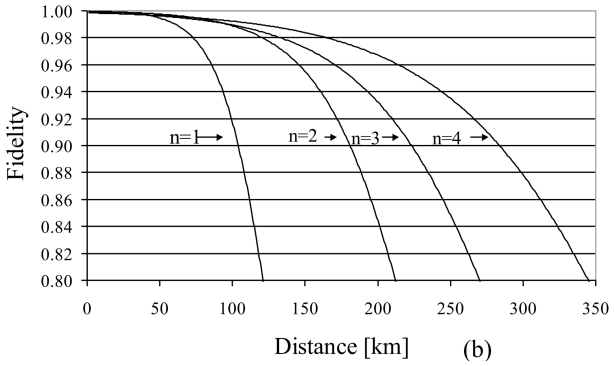

Fig. 2. " Fidelity of the transmitted quantum state as a function of the distance for different configurations. Direct transmission ($n = 1$), with an EPR source in the middle ($n = 2$), teleportation ($n = 3$), and entanglement swapping ($n = 4$). We assume that the fidelity is affected only by the detectors noise. The curves are plotted for a realistic dark count probability $D = 10^{-4}$ per ns and a fibre attenuation of 0.25 dB/km"

Then,

$$P_x = \{0,0\}$$
$$P_y = \{1,1\}$$

She sends Bob P_y and say in this case, he picks all of P_y as a subset. He observes, $[1,1]$ while turning $P_x = \{1,1\}$ from $\{0,0\}$. Now Bob communicates $[1,1]$ to Alice and she observes P_x and obtains $[1,1]$. She matches this with Bob's results and finds a perfect match. If we had a remaining subset of P_x at this point she would send it to Bob and he would have both P_x, P_y forming the whole P.

But say Eve looked at P_y while it was being sent to Bob, her observations would cause P_y to be reset randomly. Say she resets $P_y = \{0,0\}$ and saves for herself the original measurements $[1,1]$. This would also cause P_x to be negated into $(1,1)$. But when Bob observes $P_y = \{0,0\}$ again, P_x comes back to $(0,0)$. This is the scenario where Eve deceives the system. This happens exactly when Eve is able to negate all of P_y. But what is the probability of that happening for n bits? Each bit can be set randomly to a 0 or 1 so it is the same as flipping a coin n times and observing that exact sequence. Hence the probability of Eve negating all of P_y where $|P_y| = n$ is,

$$\left(\frac{1}{2}\right)^n$$

Thus we can write a function which maps from the size of subset of P_y or in other words the number of checks made, to corresponding error probability as,

$$f(x) = \left(\frac{1}{2}\right)^x$$

However we are more interested in the inverse of this function which maps from a given error probability to the number of checks. Therefore we need the inverse function of $f(x)$. Let's denote error probability by ε then observe that $f(x) = \varepsilon$ where,

$$f(x) = \left(\frac{1}{2}\right)^x \Rightarrow \log_{\frac{1}{2}}(f(x)) = x$$
$$-\log_2(f(x)) = x$$

Hence,

$$f^{-1}(\varepsilon) = -\log_2(\varepsilon)$$

We take the ceiling of this function to get whole numbers for the number of checks and write $f(x) = \varepsilon$,

$$f^{-1}(\varepsilon) = \lceil -\log_2(\varepsilon) \rceil$$

Now we can evaluate $f^{-1}(\varepsilon)$ for desired ε and obtain the number of checks given no noise.

We can show this experimentally, for a word size of 1000 and sampling randomly and increasing the sample size with each iteration until maximum sample size, in this case N, that the error dives under (or stays under) ε.

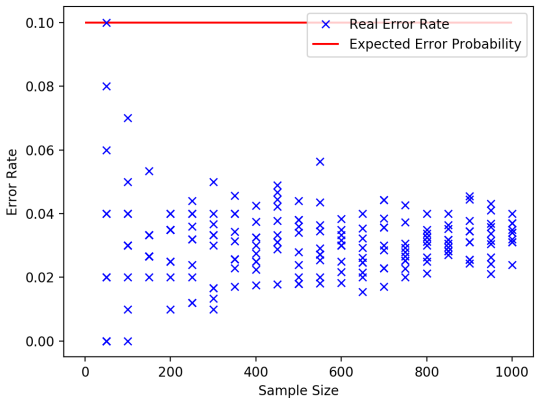

Fig. 3. $N = 1000, \varepsilon = 0.1, f^{-1}(0.1) = 4$

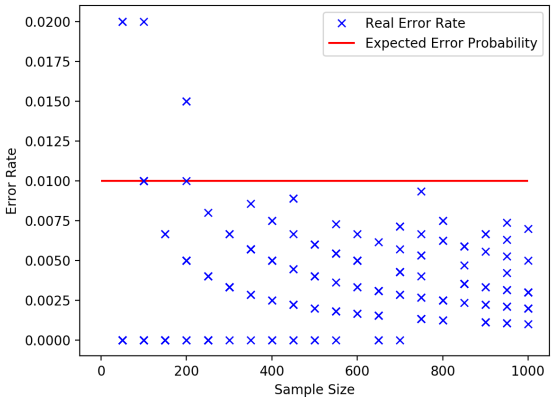

Fig. 4. $N = 1000, \varepsilon = 0.01, f^{-1}(0.01) = 7$

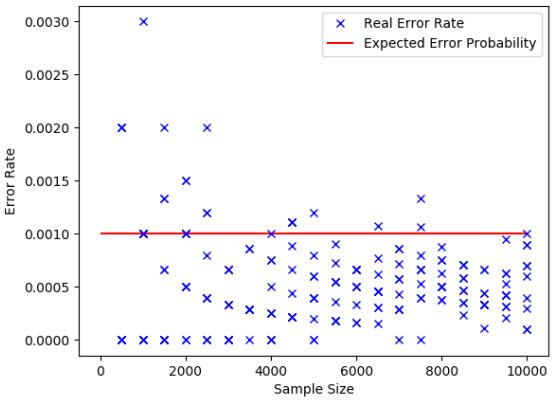

Fig. 5. $N = 10000, \varepsilon = 0.001, f^{-1}(0.001) = 10$

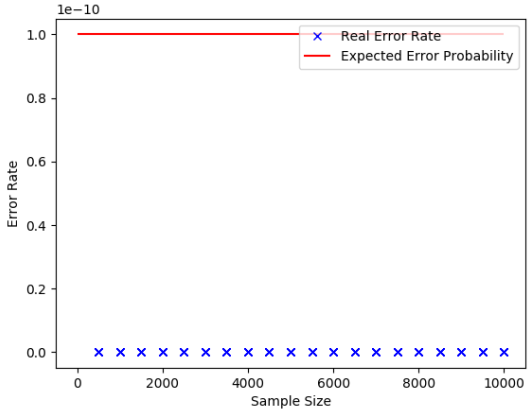

Fig. 6. $N = 10000, \varepsilon = 0.1 \times 10^{-9}, f^{-1}(0.1 \times 10^{-9}) = 34$

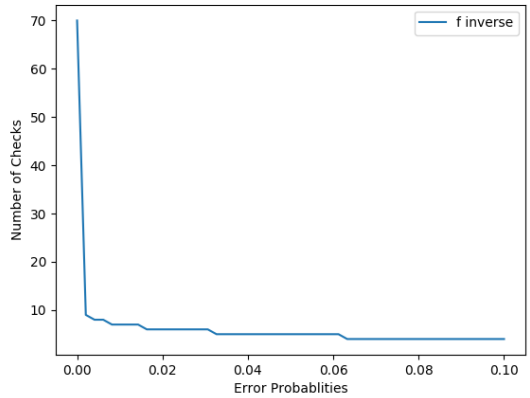

Fig. 7. $f^{-1}(\varepsilon), \varepsilon \in [0.1, 0.1 \times 10^{-20}]$

Now we face the problem where there is noise in the network like in all real world scenarios. But this time we have some headway or insight about at least how many checks we need to make? Let's define a function g that maps from error probability ε, known noise δ and word size n to the number

of checks x. Then we know,

$$\forall \delta > 0, \quad (g(\varepsilon, \delta, n) > f^{-1}(\varepsilon))$$

This simply means that if noise exists then we need to make more checks than given by our normal f^{-1}. Let's define $g(\varepsilon, \delta, n)$ such that,

$$g(\varepsilon, \delta, n) = \lceil \varphi(\varepsilon) \times \xi(\delta) \times n \rceil$$

Here we can just use the f^{-1} for φ,

$$\varphi(\varepsilon) = f^{-1}(\varepsilon) = \lceil -\log_2(\varepsilon) \rceil$$

Where ξ is what we need to find out. Naively we can just say, $\xi(\delta) = \delta$. But, we know that there exists some constant that makes this function work for a reasonable noise level δ with a given error probability ε. So let,

$$\xi(\delta) = c\delta$$

We can look for smallest c which drives the error under ε for each $\delta \leq 0.3$ upon making checks outputted by $g(\varepsilon, \delta, n)$. Following are the plots for $\delta \in \{0.3, 0.2, 0.1, 0.01, 0.001, 1 \times 10^{-4}, 1 \times 10^{-6}, 1 \times 10^{-7}, 1 \times 10^{-8}\}$,

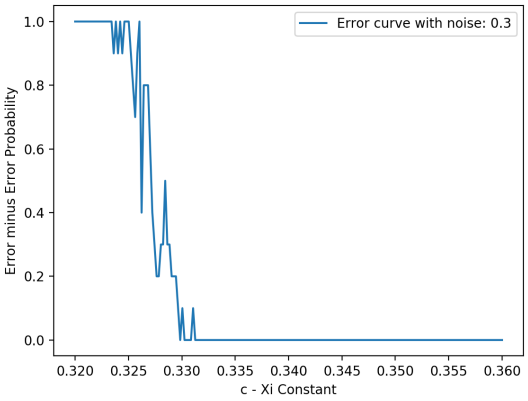

Fig. 8. $\delta = 0.3$

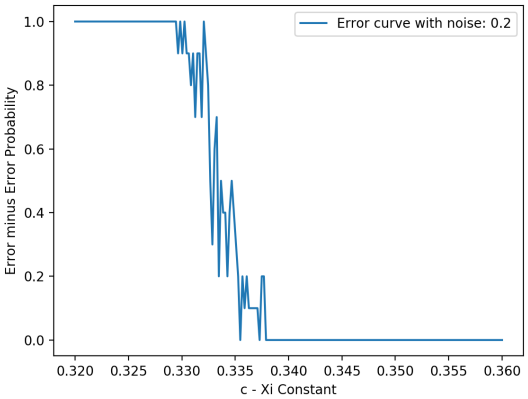

Fig. 9. $\delta = 0.2$

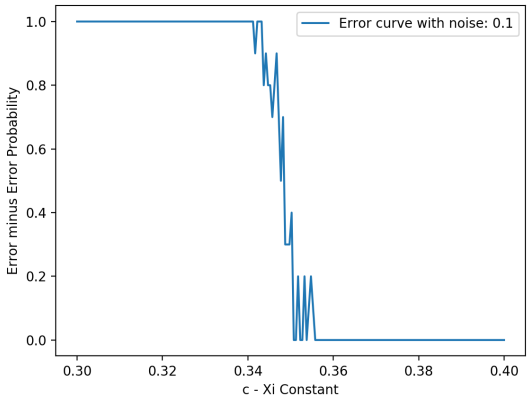

Fig. 10. $\delta = 0.1$

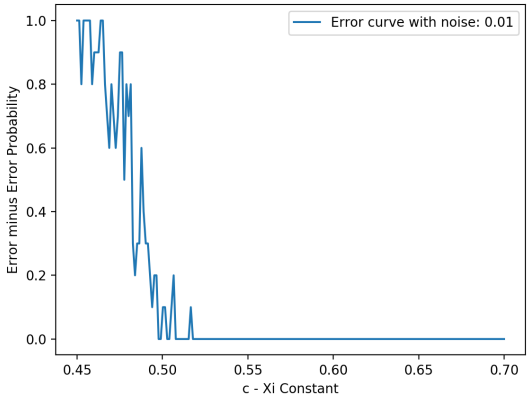

Fig. 11. $\delta = 0.01$

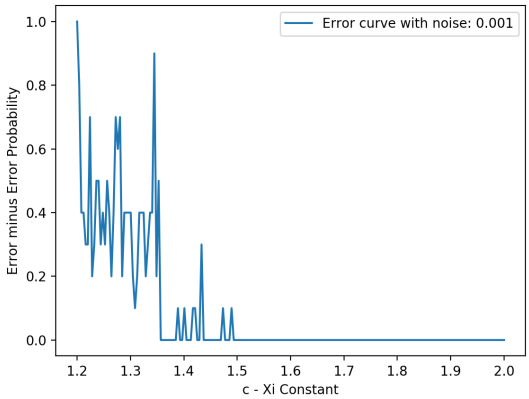

Fig. 12. $\delta = 0.001$

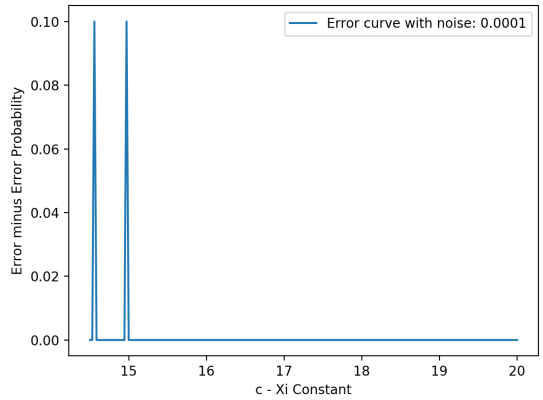

Fig. 13. $\delta = 0.0001$

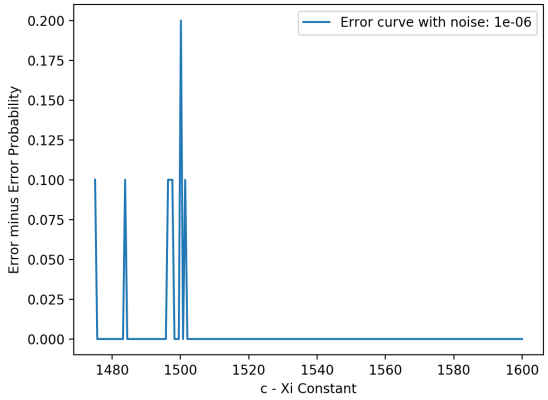

Fig. 14. $\delta = 0.000001$

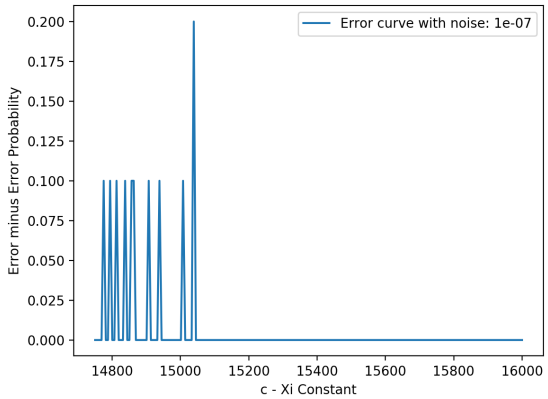

Fig. 15. $\delta = 0.0000001$

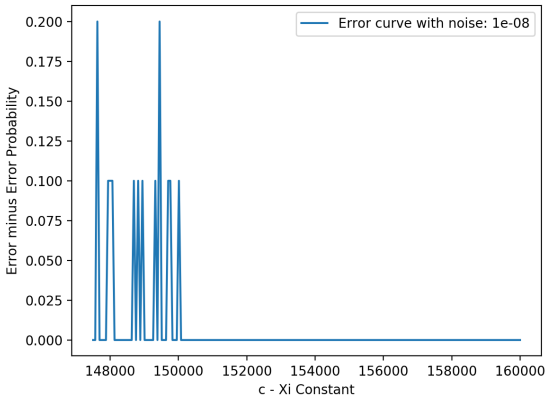

Fig. 16. $\delta = 0.00000001$

Looking at these plots we know which value of c behaves as desired. We are picking the values of c which make the function $\xi(\delta)$ drive the error bellow ε in our original function $g(\varepsilon, \delta, n)$,

δ	c
0.3	0.35
0.2	0.35
0.1	0.4
0.01	0.55
0.001	1.5
0.0001	15
0.00001	150
0.000001	1500
0.0000001	15000

Observe that after $\delta = 0.001$, there is a pattern, that for all $\delta \leq 0.001$, we have $\xi(\delta) = c\delta = 0.0015$. For the case where $0.001 < \delta \leq 0.03$, the points look logarithmic.

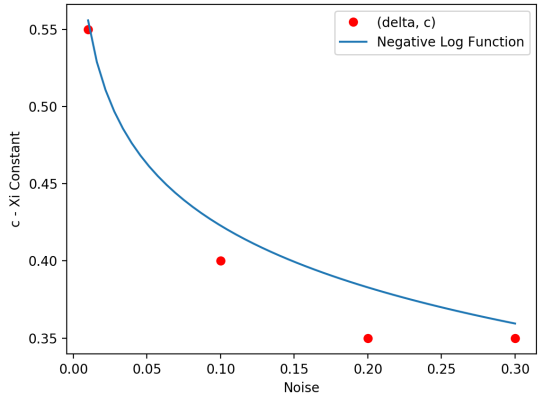

Fig. 17. $0.001 < \delta \leq 0.03, \xi(\delta) = c\delta$

Hence any negative base 2 log curve that passes above those points will work for $0.001 < \delta \leq 0.03$ in $\xi(\delta)$. Therefore, we

can finally write for $\delta \leq 0.3, \varepsilon \geq 0.01$,

$$\xi(\delta) = \begin{cases} 0.0015, & \delta > 0.001 \\ -\log_2(\delta^{0.04}) + 0.29 \end{cases}$$
$$\varphi(\varepsilon) = -\log_2(\varepsilon)$$
$$g(\varepsilon, \delta, n) = \lceil \varphi(\varepsilon) \times \xi(\delta) \times n \rceil$$

Which completes the function $g(\varepsilon, \delta, n)$. Now let's show this experimentally. Let $\varepsilon = 0.01, \delta \in \{0.3, 0.2, 0.01, 0.001\}, n = 1000$. Out max sample size N will be 1000.

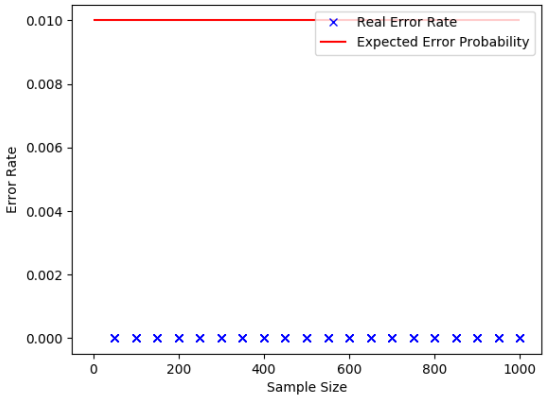

Fig. 18. $\varepsilon = 0.01, \delta = 0.2, 0.3, g(\varepsilon, \delta, 1000) = 509, 717$

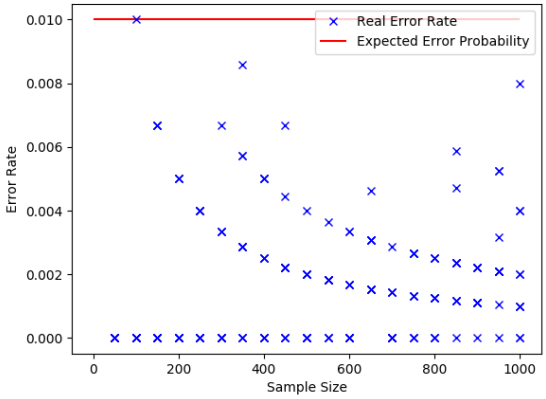

Fig. 19. $\varepsilon = 0.01, \delta = 0.01, , g(\varepsilon, \delta, 1000) = 37$

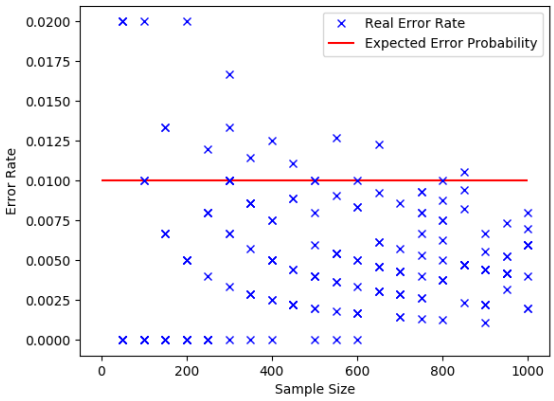

Fig. 20. $\varepsilon = 0.01, \delta = 0.001, g(\varepsilon, \delta, 1000) = 10$

Now let's fix the sample size $N = 1000$ and show the function $g(\varepsilon, \delta, n)$ for different word sizes.

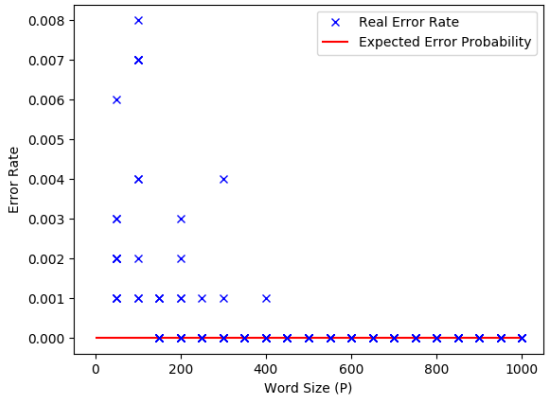

Fig. 21. $\varepsilon = 1 \times 10^{-15}, \delta = 0.01$

The following graph shows the number of checks returned by g,

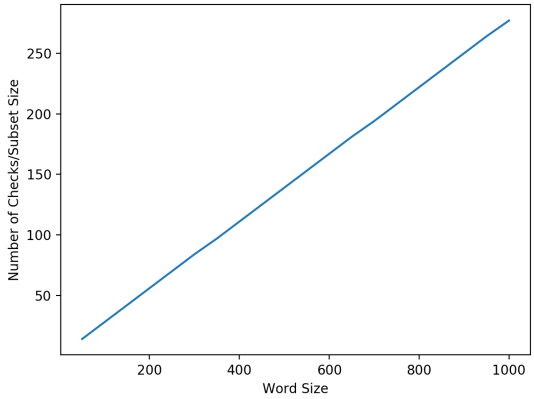

Fig. 22. Number of Checks (Subset Size) for each Word Size (P)

Now we shall turn our attention to the Machine Learning algorithms. We'll observe how can machine learning be of help in detecting an eavesdropper by looking at the ratio of difference between a subset of P_x and P_y. Additionally, how small can these subsets (from which we measure the ratio of difference) be for a desired error probability? The first step in the generation of data would be to generate EPR pairs and call the set containing them P, as we previously did. The nature of the problem here is discriminative, we would like to classify the data according to whether or not the data was eavesdropped on during the transition. This makes the problem at hand a binary classification. The question for the machine learning problem is, what are the features and labels and how do we get them? Our labels are simply a binary array of boolean, where each label tells us if the data was affected by Eve. True meaning yes–it was and false meaning no–it was not.

$$L = \{l_i \in \{0, 1\} : i < N\}$$

We generate this boolean array with some probability of Eve, which is to say true entries. Then we generate N, P's which would be the sets of EPR pairs. The number of EPR pairs in each P, or qubits will be mentioned as the word-size n.

$$\mathcal{P} = \{P_i : i < N\}, \quad |P_i| = n$$

So now we have N messages or keys which are essentially P's where the length of each P is the n. These P's correspond to the labels and if the label $l_i \in L$ of a particular $P_i \in \mathcal{P}$ is true then it was affected by Eve. We have also treated each of these P with some random value of noise (with an upper bound) and stored that vector of length N as well. Now we pick a subset of sufficient size for a desired error portability and measure the difference ratios between only that subset of P_x and P_y. We do it for all $P \in \mathcal{P}$ and this gives us N ratios of differences. Combined that with the noise each P_i was affected with and we have our features. They are an $N \times 2$ matrix where the first column is difference ratios and the second column is the noise that effect that EPR pair.

The first machine learning algorithm that comes to mind for binary classification is *Logistic Regression*. We can use this model to judge how our initial accuracy rates look like, specially At this point I would like to point out that the cost of false negatives (saying that message wasn't stolen while it was) in our classification is far more costly than false positives (message was not stolen but we said it was). So we'd observe not just the accuracy rate but also the false false negative rates.

Let's train a logistic regression model with word-size $n = 1000$, 1000 training examples and 10000 testing examples. Observe the plots for the sufficient size of a subset of P with desired accuracy of $0.7, 0.8, 0.9, 0.999$. We will look at different noise levels for these accuracies along the x-axis and see what is the sufficient subset size,

In addition to logistic regression I will also implement simple *Linear Regression* before it is passed through a function like sigmoid. This is so I can pick a threshold for classifying a certain data point either way. This will help us drive the false negatives down to zero and we would be able to compare at what cost of false positives could that happen. We would test

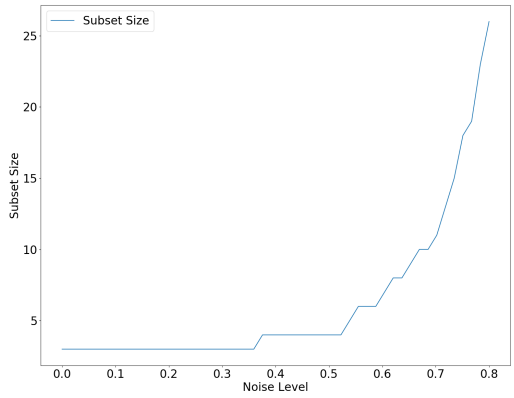

Fig. 23. Subset Size for Noises, Accuracy: 0.7

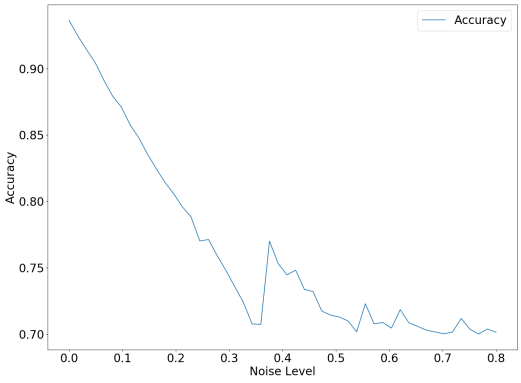

Fig. 24. Accuracy: 0.7

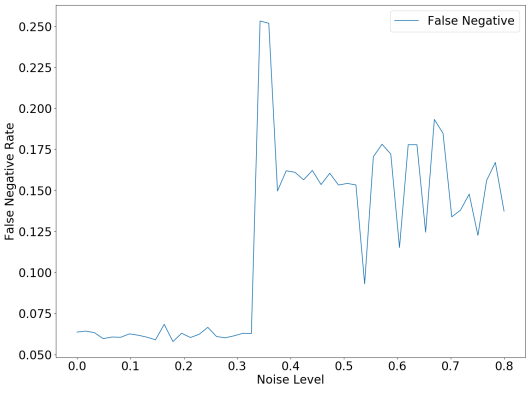

Fig. 25. False Negative Rate, Accuracy: 0.7

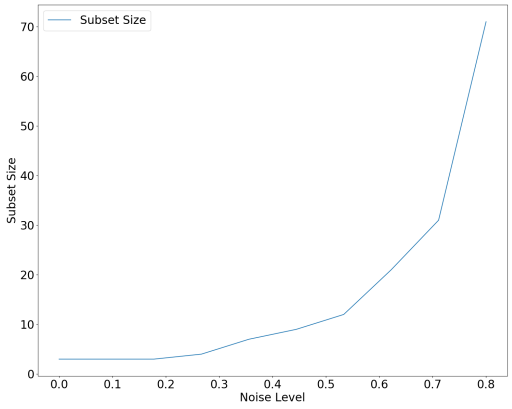

Fig. 26. Subset Size for Noises, Accuracy: 0.8

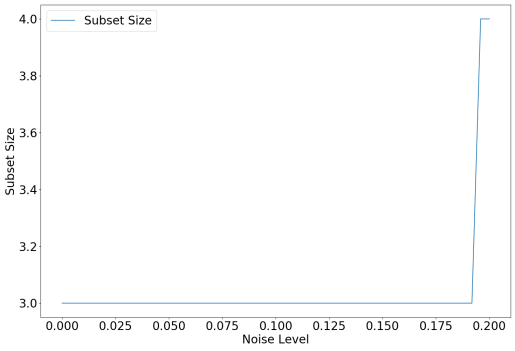

Fig. 27. Subset Size for Noises, Accuracy: 0.8

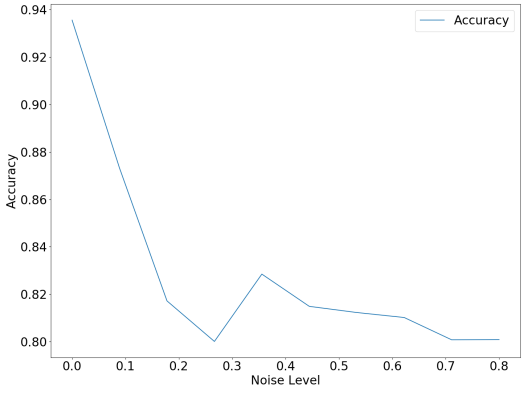

Fig. 28. Accuracy: 0.8

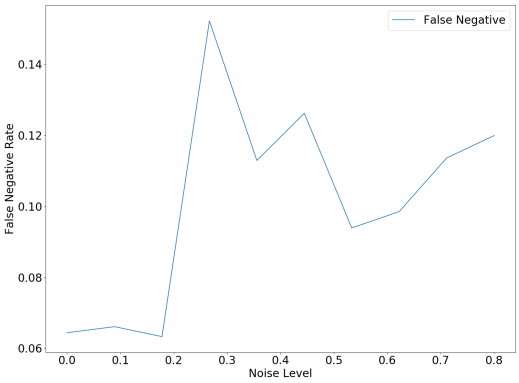

Fig. 29. False Negative Rate, Accuracy: 0.8

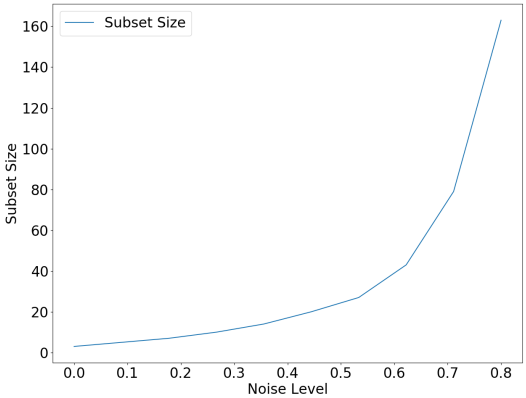

Fig. 30. Subset Size for Noises, Accuracy: 0.9

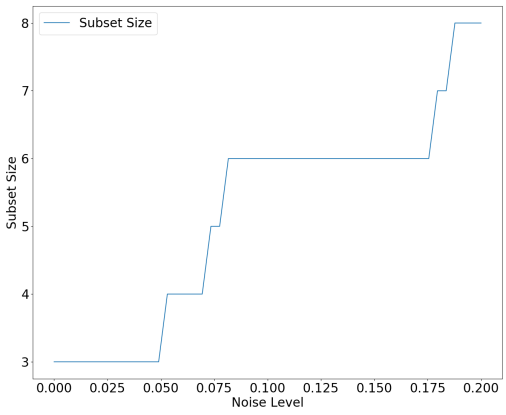

Fig. 31. Subset Size for Noises, Accuracy: 0.9

the effect of different thresholds on false negatives and false positives, to see how they correlate and if we can bring the false negative down to zero for some threshold value. Let's train a simple linear regression with word-size $n = 1000$, 10000 training examples and 1000 testing examples. We learned from our previous logistic regression that a subset size of 40 is good enough for any reasonable noise level

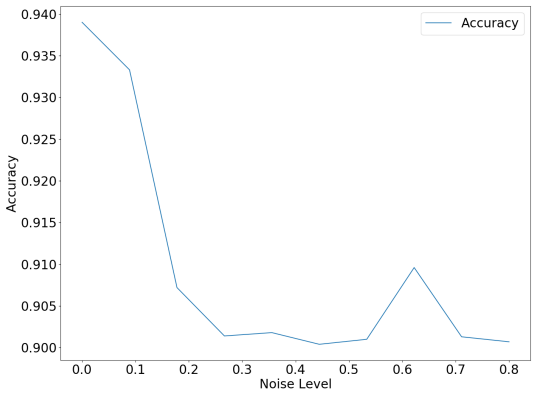

Fig. 32. Accuracy: 0.90

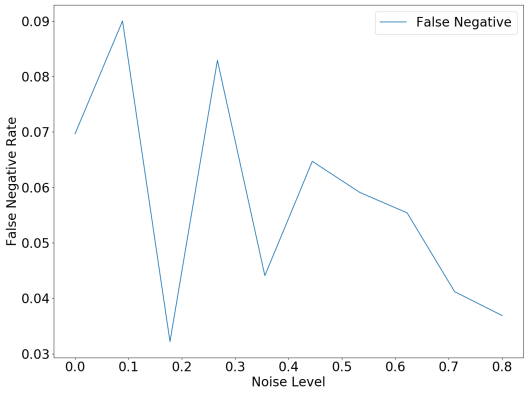

Fig. 33. False Negative Rate, Accuracy: 0.90

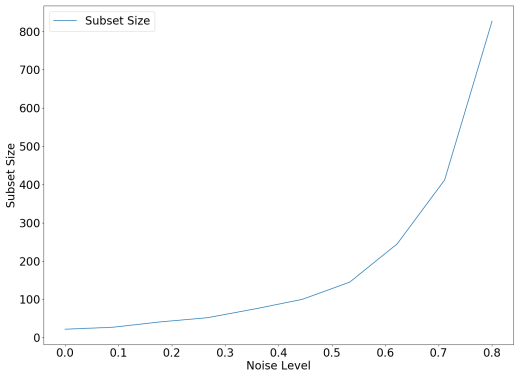

Fig. 34. Subset Size for Noises, Accuracy: 0.999

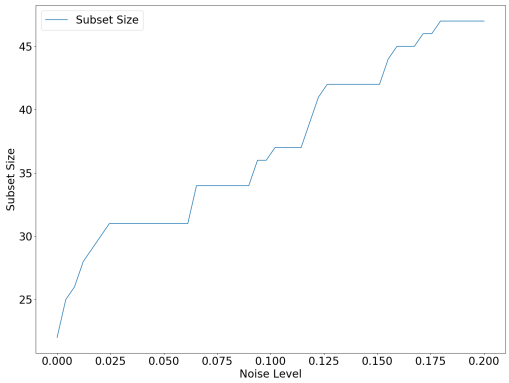

Fig. 35. Subset Size for Noises, Accuracy: 0.999

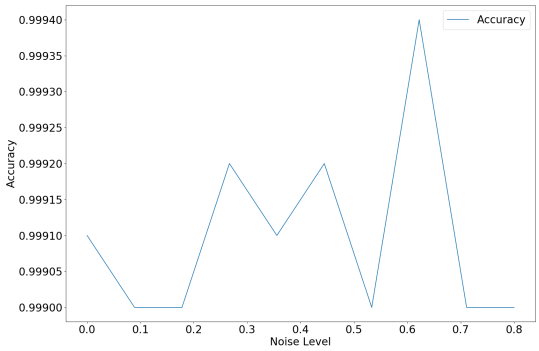

Fig. 36. Accuracy: 0.999

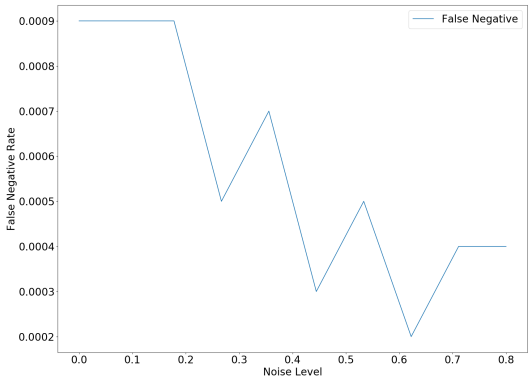

Fig. 37. False Negative Rate, Accuracy: 0.999

and accuracy, so we will use this information in our linear model. Furthermore, any output by the model which is bellow zero will be set to zero and above one to one respectively. To visualise our threshold we look at a testing set of 100 example (with increasingly noisy samples) and it's real labels versus predictions.

The plots show how the model becomes less sure of itself as the noise increases (becomes unreasonable). However, by just looking at the plots we can tell that if we pick our threshold

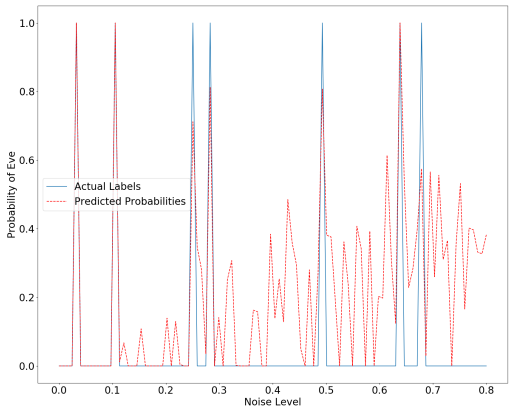

Fig. 38. Predictions of linear model across different noise levels

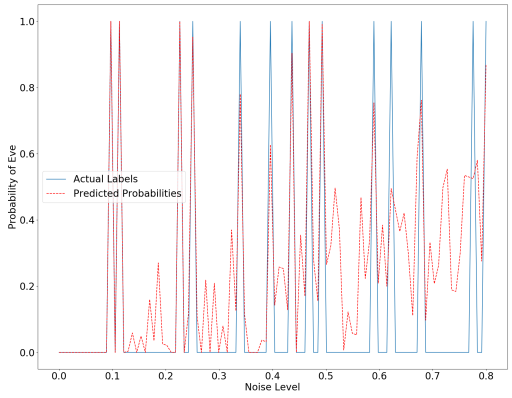

Fig. 39. Predictions of linear model across different noise levels

to be around 0.4, which means if the 0.4 probability of Eve is classified as a Yes/True then we will catch a few more false positives but almost no false negatives as desired. Consider the ROC plot and the plot of false negatives vs false positives given below. The "Error Rates versus the Threshold" figure agrees with with the value of 0.4 for about no false negatives and shows their relationship with false positives. We will pick threshold to be 0.25 and show confusion matrices with test sets where $50\%, 10\%, 1\%$ of the three test sets were corrupted by Eve.

```
Eve: 0.525000
Accuracy: 0.852000
Confusion Matrix:
[[0.328 0.147]
 [0.001 0.524]]

Eve: 0.085000
Accuracy: 0.735000
Confusion Matrix:
[[0.65  0.265]
 [0.    0.085]]

Eve: 0.010000
Accuracy: 0.715000
Confusion Matrix:
[[0.705 0.285]
 [0.    0.01 ]]
```

Here for the chosen threshold $= 0.25$, we get about 0.8-0.7 accuracy and for the test sets with the probability of Eve being less than 10%, we get no false negatives.

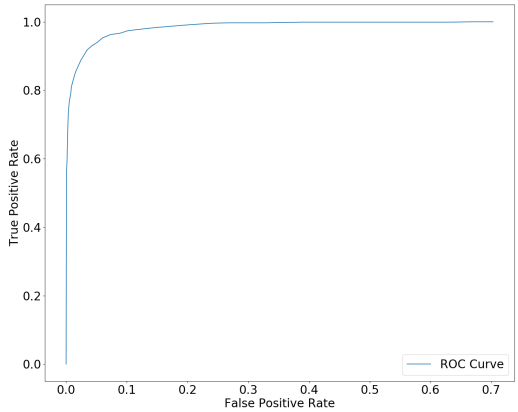

Fig. 40. ROC

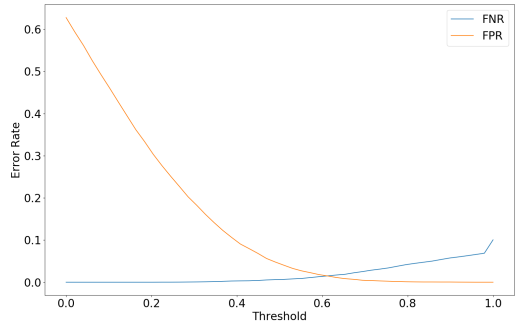

Fig. 41. Error Rates versus the Threshold

We generated synthetic data, now we can look at how will a prospective data set look like where some percentage of samples are affected by Eve and some are not. Is there a clear difference between them? The very first question that comes to mind is that if we can observe two clusters in our data where one is the cluster affected by eve, if so will all the data points which get eavesdropped on, lie close to that cluster? *K-means clustering* could be utilised in answering these questions. Let's first generate data with really low noise and probability of eve being zero and then plot the difference ratios against noise. We can see that the graph shows no difference between the subset of P_x and P_y which is expected due to almost no noise and no attacks. Now if we turn on the noise a reasonable amount

we see some difference ratios as shown in the figure bellow. In this case if there is an attack by Eve, it can be detected clearly and the clusters can be found easily by *K-means* as shown.

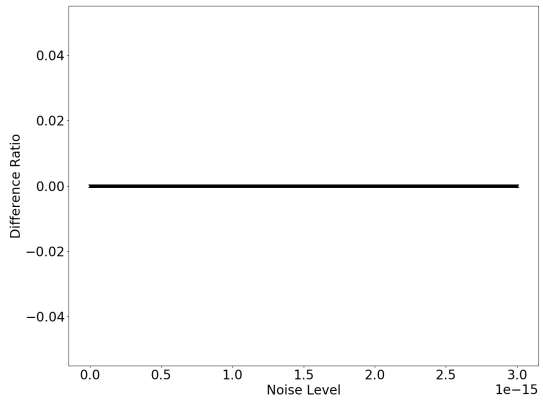

Fig. 42. No noise and absence of Eve

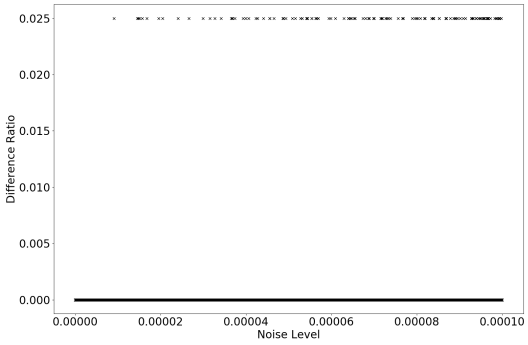

Fig. 43. Reasonable noise and absence of Eve

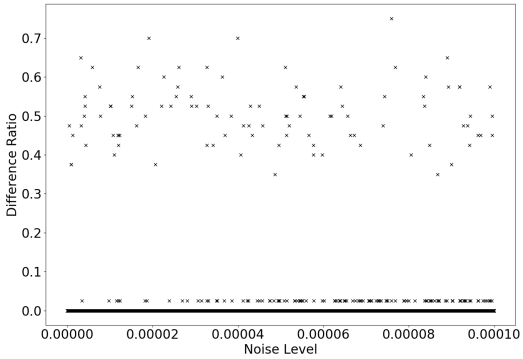

Fig. 44. Presence of Eve in reasonable noise

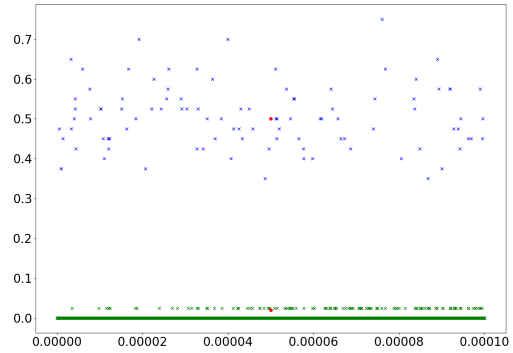

Fig. 45. Eve Detected

However observe what happens in absence of Eve when the noise approaches 0.5. Then observe how does the cluster of Eve affected points look like in presence of no noise. We can tell what would happen if these two occurred at the same time.

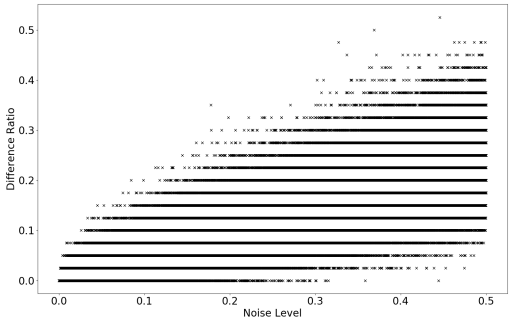

Fig. 46. Noise reaches 50% in absence of Eve

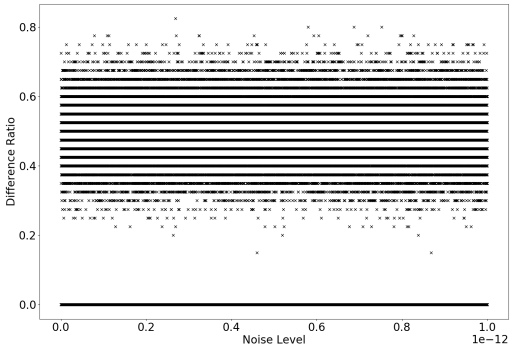

Fig. 47. Presence of Eve with 0% noise

K-means still does a pretty good job of clustering the two classes, the confusion matrix for this case is:

```
Confusion Matrix:
```

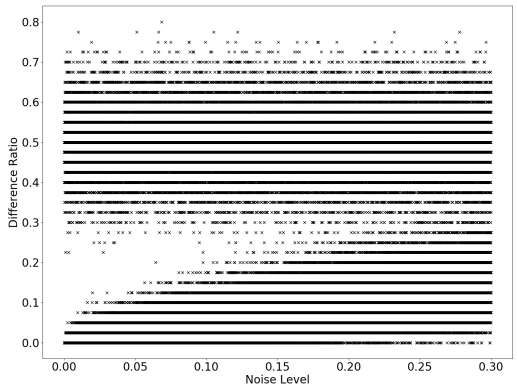

Fig. 48. Max 30% chances of Eve and noise level

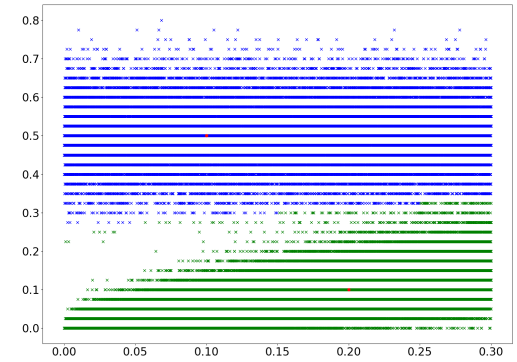

Fig. 49. Clusters Found

```
[[7.0118e-01 1.0000e-04]
 [1.9800e-03 2.9674e-01]]
```

Therefore we conclude that *K-means* is the best choice for reasonable noise and probability of an attack because it gives no false negatives and positives. Remember that we are still using subset's of size 40 to calculate difference ratios for the features even in *K-means*. We used logistic regression to observe this number and then implemented linear regression with observation of a threshold in case where we want do force the false negative down to zero even in bad noise.

Lastly, We attempt classification using a *Neural Network*. I implemented a neural network with three layers namely an input layer, a hidden layer and an output layer. It was trained on 10000 training examples which were again generated using a subset of P of size 40 and a word-size of 1000. With completely random assignment of initial weights I get about $80\% - 90\%$ accuracy. I iterated 10 times. However, starting off with random weights bearing a standard deviation of 0.001 and a seed of 0, (I used Tensorflow in Python), I was able to obtain accuracy of $92\% - 99\%$ with false negatives approaching 0 like the following:

```
Iteration: 0 Average Cost = 0.693
Iteration: 1 Average Cost = 0.693
Iteration: 2 Average Cost = 0.693
Iteration: 3 Average Cost = 0.688
Iteration: 4 Average Cost = 0.585
Iteration: 5 Average Cost = 0.474
Iteration: 6 Average Cost = 0.449
Iteration: 7 Average Cost = 0.441
Iteration: 8 Average Cost = 0.437
Iteration: 9 Average Cost = 0.434

Final Accuracy:
0.99

[[0.66  0.10]
 [0.00  0.33]]
```

IV. Conclusion

I iterated over methods of quantum communication and it's application in cryptography. We showed numerous papers concentrating on Quantum Key Exchange which depends on the quantum property of entanglement found in EPR pairs. The schemes relied upon the fact that under some noise and given error probability we could use a subset of the set of EPR pairs for security checks, the remaining set would be the key or the message. I formulated a theoretical function which outputs the size of the subset which we need to check for a desired error probability. The function is as follows,

$$\xi(\delta) = \begin{cases} 0.0015, & \delta > 0.001 \\ -\log_2(\delta^{0.04}) + 0.29 \end{cases}$$
$$\varphi(\varepsilon) = -\log_2(\varepsilon)$$
$$g(\varepsilon, \delta, n) = \lceil \varphi(\varepsilon) \times \xi(\delta) \times n \rceil$$

Here, $\delta \leq 0.3$, is the noise and $\varepsilon \geq 0.01$, is the error probability.

We then looked at machine learning algorithms considering our problem as a classification problem. The features were the difference ratios and noise. Logistic regression helped us find the appropriate size of the subset of P to calculate the difference ratios for the features. This subset correlated to a desired accuracy. In order to drive down the more costly false negatives, we implemented linear regression. Linear regression allowed for a threshold which enabled us to minimise false negatives for the best false positive rate we could get. Lastly K means clustering worked and found clusters perfectly in a reasonable amount of noise and we saw what could go wrong as the noise went up. A neural network showed the possibility of 99% accuracy with no false negatives when trained with shown initial weights and biases.

Further research could be done on the implications of allowing an error rate. For instance, is it possible for Eve to steal certain small enough parts of messages without causing error that is bigger than expected from a communication where same data is shared over and over again. A more theoretically compact noise function g can be looked for and proved with mathematical rigour. Machine Learning algorithms other than the ones i implemented could also be analysed.

ACKNOWLEDGEMENT

I'd like to thank Prof. Crick for helping and guiding me throughout the process of research and introducing me to the ways and methods it is done. I'd also like to thank Dr. Dai for attending this paper's presentation and Dr Fili for introducing me to Cryptography and helping me work through some of it's mathematics.

REFERENCES

[1] W. Diffie, "W. diffie and m. hellman, ieee trans. inf. theory 22, 644 (1976)." *IEEE Trans. Inf. Theory*, vol. 22, p. 644, 1976.

[2] F.-G. Deng, G. L. Long, and X.-S. Liu, "Two-step quantum direct communication protocol using the einstein-podolsky-rosen pair block," *Physical Review A*, vol. 68, no. 4, p. 042317, 2003.

[3] D. Bohm, "Quantum theory," *Prentice Hall, Englewood Cliffs, NJ*, 1951.

[4] I. Marcikic, H. De Riedmatten, W. Tittel, H. Zbinden, and N. Gisin, "Long-distance teleportation of qubits at telecommunication wavelengths," *Nature*, vol. 421, no. 6922, p. 509, 2003. [Online]. Available: https://www.nature.com/articles/nature01376

[5] A. Shimony, "Proceedings of the international symposium on foundations of quantum theory," *Physical Society of Japan, Tokyo*, 1984.

[6] P. A. Schilpp, "Albert einstein; philosopher, scientist," 1951.

[7] J. Hoffstein, J. C. Pipher, J. H. Silverman, and J. H. Silverman, *An introduction to mathematical cryptography*. Springer, 2008, vol. 1.

[8] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels," *Physical review letters*, vol. 70, no. 13, p. 1895, 1993. [Online]. Available: https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.70.1895

[9] J. L. Park, "The concept of transition in quantum mechanics," *Foundations of Physics*, vol. 1, no. 1, pp. 23–33, 1970.

[10] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[11] C. H. Bennett, "Quantum crytography," in *Proc. IEEE Int. Conf. Computers, Systems, and Signal Processing, Bangalore, India, 1984*, 1984, pp. 175–179.

[12] A. Cabello, "Quantum key distribution in the holevo limit," *Physical Review Letters*, vol. 85, no. 26, p. 5635, 2000.

[13] C. H. Bennett, "Ch bennett, phys. rev. lett. 68, 3121 (1992)." *Phys. Rev. Lett.*, vol. 68, p. 3121, 1992.

[14] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical review letters*, vol. 67, no. 6, p. 661, 1991.

[15] C. Bennett, "Ch bennett, g. brassard, and nd mermin, phys. rev. lett. 68, 557 (1992)." *Phys. Rev. Lett.*, vol. 68, p. 557, 1992.

[16] G.-L. Long and X.-S. Liu, "Theoretically efficient high-capacity quantum-key-distribution scheme," *Physical Review A*, vol. 65, no. 3, p. 032302, 2002.

[17] H. de Riedmatten, I. Marcikic, W. Tittel, H. Zbinden, D. Collins, and N. Gisin, "Long distance quantum teleportation in a quantum relay configuration," *Phys. Rev. Lett.*, vol. 92, p. 047904, Jan 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.92.047904