Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Turanov Dilimbek

Development of the model and methods of quantum encryption on MatLab

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Computing Systems and Software
(degree code: 6M070400)
Department of Computer Sciences
Faculty of Engineering and Natural Sciences

Supervisor: **Baimuratov Olimzhon**

Kaskelen, 2019

Abstract

Quantum cryptography is one of the areas that has been studied very deeply lately. The whole new world is very closely connected with network resources. Due to the imminent era of quantum computers or very powerful computing systems, the entire existing cryptosystem was under threat. And many see a way out of this situation in the use of quantum systems. At the moment, it is very difficult to implement all aspects of quantum systems, due to the limitations of existing technologies. Despite this, some aspects can all be put into practice, but most aspects are only mathematically realized. In this master's thesis, an attempt is made to implement these aspects on the MatLab platform.

Аңдатпа

Кванттық криптография - соңғы уақытта өте терең зерттелген салалардың бірі. Жаңа әлем желілік ресурстармен тығыз байланысты. Кванттық компьютерлердің немесе өте қуатты есептеу жүйелерінің пайда болу кезеңіне байланысты. барлық қолданыстағы криптожүйе қатерге ұшырады. Көптеген адамдар кванттық жүйелерді пайдалануда бұл жағдайды көреді. Қазіргі кезде қолданыстағы технологиялардың шектеулеріне байланысты кванттық жүйелердің барлық аспектілерін іске асыру өте қиын. Осыған қарамастан. кейбір аспектілерді қолдануға болады. бірақ көптеген аспектілер тек математикалық түрде жүзеге асады. Бұл магистрлік диссертациялық жұмыста MatLab платформасында осы аспектілерді іске асыруға әрекет жасалды.

Аннотация

Квантовая криптография одно из направлений которое очень глубоко изучается в последнее время. Весь новый мир очень тесно контактирует с сетевыми ресурсами. Из-за грядущей в скором времени эпохи квантовых компьютеров или очень мощных вычислительных систем, вся существующая криптосистема оказалось под угрозой. И многие видят выход из данной ситуаций в использований квантовых систем. В данный момент очень затруднительно реализовать все аспекты квантовых систем, из-за ограниченности существующих технологий. Несмотря на это какие-то аспекты все ж удается реализовать на практике, но большинство аспекты лишь реализованы математический. В данной магистерской диссертаций делается попытка реализовать эти аспекты на платформе MatLab.

Contents

1. Introduction

1.1 Motivation

Quantum cryptography is a very dynamic branch of modern cryptographic science, promising many new perspectives in traditional areas of application - diplomatic communications, military, business and other areas that require the transfer of secret information. Experimental and theoretical work on quantum cryptography, carried out until today, considered a variety of various exchange schemes and protocols, as well as the stability of these schemes and protocols in relation to various methods of unauthorized access. Among the already considered techniques, there are many quite complex ones that are practically unrealizable in the framework of the technology of the foreseeable future.

1.2 Aims and Objectives

- Analyze quantum systems to encrypt and development of the models with quantum states for use in the classical models.
- analysis tools and platforms for develop quantum systems
- analysis of protocols BB84, B92...
- development of a model for key generation based on the BB84 protocol in Matlab
- implementation of quantum encryption methods for token generation

2. Analysis of models and methods quantum system with encryption

Since ancient times, people have sought ways of communication that would ensure the preservation of the transmitted information in secret from third parties, which was important for the needs of diplomacy, trade, military Affairs and love correspondence. Various types of information coding were used for this purpose. All of them provided the secrecy of the transmitted information in one way or another, but none of them gave absolute protection. In 1918, vernam Invented the cipher, for which later, at the end of the 40-ies,, was held proof of absolute secrecy. The conditions of this secrecy are, in fact, the main drawback of this cipher: a completely random key of the same length as the transmitted message is required, and this key should be used only once. Therefore, before you can send a secret message, you must first pass through a channel that is highly secure from unauthorized access, the same length of the message containing the secret key. Such a system is cumbersome, inconvenient to use and expensive, which is why it is rarely used. In the 70s, the so-called public key cryptography system was invented, in which there are two keys: one for encrypting messages, publicly disclosed, and the other for decrypting, kept secret. This system is now used almost everywhere, although its secrecy has not been strictly proved by anyone (as, indeed, the opposite has not been proven). This system is based on a special kind of functions, the calculation of which in one direction is not difficult, and in the opposite direction - very difficult. In particular, the problem of calculating the secret key in the presence of a public key is reduced to the problem of factorization of large numbers, which is considered difficult to solve until now.

However, due to the expected appearance of quantum computers for which fast factorization algorithms have already been developed, public key systems may lose their effectiveness. Therefore, there was a need for cryptographic systems based on other principles. The work "Conjugate coding", which was written by Stephen Wiesner from Columbia University, at first few people noticed and not even published, marked the beginning of a new direction in cryptographic science - quantum cryptography. In it, thanks to the laws of quantum mechanics, it became possible to distribute between two or more subscribers a secret key that meets all the requirements of the Vernam cipher, which means the absolute secrecy of the transmitted information. In 1984, Bennett and Brassard patented the first exchange Protocol for a quantum cryptographic system, known as BB84. Since then, interest in quantum cryptography in the world began to grow extremely quickly, and to date, a huge number of studies have been conducted, affecting various aspects of it. According to the authors of BB84, quantum cryptography is a method that allows two users who do not initially have any secret data common to them to agree on a random key that will be secret from a third party who exercises unauthorized access to their communications. In cryptographic science has developed its own traditional terminology, somewhat specific sounding at first glance, but very convenient in practice. Thus, legal users are traditionally called "Alice" and "Bob", while the person who performs unauthorized access is called "eve". We will not deviate from the canons and will keep this terminology in this paper. The main quantum mechanical principles that form the basis for quantum cryptography are: 1. The inability to distinguish between absolutely two non-orthogonal quantum States The arbitrary state of any two-level quantum mechanical system can be represented as a linear superposition of its eigenstates and with complex coefficients: . where. The laws of quantum mechanics do not allow to distinguish two quantum States absolutely reliably

and . if not met – 0, i.e. the States are orthogonal. 2. The prohibition of cloning theorem Due to the unitarity and linearity of quantum mechanics, it is impossible to create an exact copy of an unknown quantum state without affecting the initial state. For example, suppose Alice and Bob use two-level quantum systems to transmit information, encoding bits of data by the States of these systems. If eve intercepts the carrier of information sent by Alice, measures its state and sends further to Bob, then the state of this carrier will be different, than at

measurement. Thus. eavesdropping on a quantum channel induces transmission errors that can be detected by legitimate users. 3. Quantum entanglement Two quantum mechanical systems (even separated spatially) can be in a state of correlation. so that the measurement of the selected value carried out on one of the systems will determine the result of the measurement of this value on the other. This effect is called quantum entanglement. None of the entangled systems is in a certain state. so the entangled state can not be written as a direct product of the States of the subsystems. The singlet state of two particles with spin 1 2 can serve as an example of an entangled state:

The measurement performed on one of the two subsystems will give with equal probability or . and the state of the other subsystem will be opposite (i.e.. if the measurement result on the first system was . and Vice versa). 4. Causation and superposition Causality. which is not initially an ingredient of non-relativistic quantum mechanics. can nevertheless be used for quantum cryptography in conjunction with the principle of superposition: if two systems whose States form a certain superposition are separated in time. without being connected by causality. it is impossible to determine the superposition state by taking measurements on each of the systems sequentially. The communication process will be discussed in detail on the example of the BB84 Protocol. as historically the first and most popular at the moment. The remaining protocols will be described very briefly. As for the specific schemes of quantum cryptographic installations. here will be considered only those of them. eavesdropping which is the subject of this study using the exchange protocols BB84 and B92 on phase states.

3. Development of a model in Matlab for key generation based on the BB84,B92 Protocol

3.1 BB84 Protocol

The first exchange Protocol for quantum cryptography, called BB84 [4], was invented by Bennett and Brassard in 1984. It uses four quantum States of a two-level system to encode information, forming two conjugate bases (indicated here by letter indices A and B): (see Figure 3.1)

Here, the states $|0_A\rangle$ and $|1_A\rangle$ encode the values "0" and "1" in basis A, and $|0_B\rangle$ and $|1_B\rangle$ encode the same values in basis B. We can imagine them as polarization states of a particle with spin 1/2: $|0_A\rangle$ and $|1_A\rangle$ correspond to horizontal (0°) and vertical (90°) polarization directions, and $|0_B\rangle$ and $|1_B\rangle$ correspond to two diagonal directions, namely 45° and −45° (obtained from $|0_A\rangle$ and $|1_A\rangle$ by rotating the coordinate system by 45°). Two states belonging to the same basis are orthogonal, that is, they can be distinguished reliably provided that the measurements are carried out in the same basis. However, a measurement in the wrong basis (i.e., for example, an attempt to determine which of the two polarizations - 0° or 90° – has a particle that is actually polarized at an angle of 45°), will give an absolutely random result.

We first describe the Protocol under the assumption of no noise in the quantum channel, then modify the description to take noise into account. Information

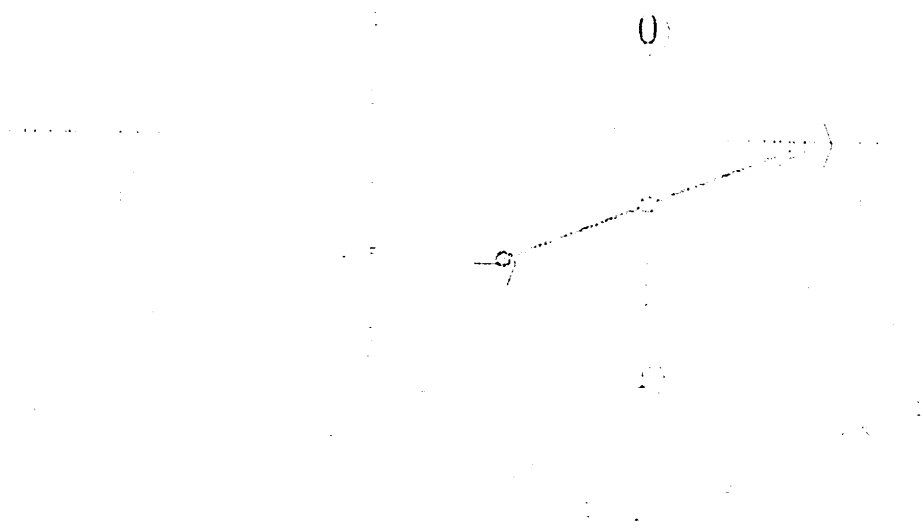


Figure 3.1: All bases position in Sphere Bloch. $|0\rangle$. $|1\rangle$. $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

exchange is carried out in two stages: first through the quantum channel, then through the usual channel open to eavesdropping (for example, via the Internet). In the first stage, Alice chooses randomly and with equal probability one of the four quantum States $|0_A\rangle$. $|1_A\rangle$. $|0_B\rangle$. $|1_B\rangle$, and sends it to Bob on the quantum channel, fixing in their records the value of the data bit and the basis in which it is encoded. Bob measures the transmitted state in one of two possible bases A or B, selected independently of Alice, randomly and with equal probability, also recording the measurement result and the selected basis. If the basis selected by Bob for the measurement matches the basis selected by Alice for transmission, the data bits of Alice and Bob will be identical; otherwise they will coincide with probability 1 2. Alice and Bob repeat this procedure N times, as a result of which each of them will have a string of bits of length N. Since the choice of bases was carried out by users randomly and independently, in about 50% of cases they will choose different bases for transmission and detection.

In the second stage, Alice and Bob communicate through an open channel, which, however, must have the property that eve can not change the messages transmitted between them. Alice and Bob tell each other the values of the bases they used in the transmission, and agree to exclude from their data those bits for which the transmission and detection bases did not coincide. The resulting string bit is

called the raw key.

Let us imagine that eve carries out eavesdropping the quantum channel. capturing the media information sent by Alice. by measuring their state and forwarding them next to Bob. This strategy is called "interception regeneration". We will consider here only those cases in which Alice and Bob have chosen the same bases (the remaining bits will be excluded from the final key in any case). Since Eva is forced to choose the bases for detection randomly and independently of Bob and Alice. then approximately 50% of the bases of eve and Bob will not coincide. The results of Bob's measurements will be random, but approximately 50% coinciding with Alice's data. Thus. Bob's measurements will give the correct result with probability 1 2 1 2 * 1 2 3 4. while in the absence of eve they would give the correct result always.

This means that in order to perform the Eva presence test. Alice and Bob must compare publicly some randomly selected subset of their data (of course. not using then the data bits from that subset). If there are errors. it means that Eva was eavesdropping: in this case. the data is discarded and the transfer process begins from the beginning. If there are no errors. the remaining bits form the final secret key.

There is. however. another way of eavesdropping. known as "beam splitting". The principal feature of it is that Alice and Bob are not able to determine the presence of this kind of eavesdropping in the channel. It is known that with the widely used method of obtaining single photons. namely. the attenuation of laser radiation to the average photon intensity $\mu \leq 1$ per pulse. a certain proportion of the output radiation will contain more than one photon per pulse (this is determined by the Poisson statistics. to which the laser radiation is subject). Thus. putting an ordinary divider in the path of photons. Eva can get some information about the key. and without making mistakes in the transmission. This possibility is taken into account by Alice and Bob in the process of obtaining the final secret key: they exclude from their data the number of bits corresponding to the amount of information that Eva can receive as a result of this attack.

The above for the second stage is valid only for an ideal. silent quantum channel. But in the real channel there is always noise. so some discrepancy in the data of Alice and Bob will always be. even in the absence of eavesdropping. Since Alice and Bob cannot distinguish between errors that cause eavesdropping and errors

caused by natural channel noise, they have to assume that all transmission errors are caused by eve's presence. At this stage of the exchange in an open channel is complicated.

First, Alice and Bob extract the raw key as described above, and, of course, those broken intervals are removed where Bob could not project the particle at all (for example, due to an imperfect detector, or because of the imperfection of the method used to generate single photons). It is necessary to tell that in real systems of such intervals the majority.

Next, Alice and Bob make an estimate of the percentage of errors in the raw key, publicly comparing their randomly selected subset of data, which, of course, will be excluded from further consideration. If the error rate exceeds some specified level, it will be impossible for Alice and Bob to come to the shared secret key. In this case, all data is discarded and the transfer process starts again. If this level is not exceeded, Alice and Bob move on to error correction.

The goal here is to remove all errors from the raw key and arrive at a common, error-free code sequence (which will, however, be only partially secret, due to the fact that some information will leak to eve during the correction process itself). To begin with, Alice and Bob make some random permutation of their data in order to randomize the error position. After that, the lines are divided into blocks of length l, and this length is chosen so that the probability of detecting more than one error in the same block is small enough (l is chosen based on the estimated percentage of errors in the raw key). For each of the blocks, parity is checked, and then the last bit of each of the compared blocks is excluded. If Alice and Bob do not have the same parity, a binary search for the location of the erroneous bit is performed inside the block, with the exception of the last bits of the compared subunits. The found erroneous bit is also deleted. The whole process (permutation, block partitioning and parity checking) is repeated the required number of times, after which the same actions are performed, but with parity checking in randomly selected subsets and excluding a randomly selected bit. Finally, if no errors are found during a number of consecutive iterations, Alice and Bob conclude that there is a very high probability that the remaining data contains no errors.

As mentioned above, the data remaining after error correction will be only partially secret. The following procedure is used to extract the final secret key from

this data. Based on the percentage of errors in the raw key, the maximum number of bits k known to eve from the total number of remaining bits n is determined. Let s also be a privacy parameter, the value of which is chosen by users arbitrarily. Alice and Bob publicly select n − k − s random subsets of their data. They do not reveal the parity of these subsets - these parity and make up the final secret key. It can be shown that the General information that eve may have about the final key is less than $2^{-s}/ln2$ bit.

3.2 B92 protocol

For most protocols, only the exchange process over the quantum channel will be described, since the second stage of communication is basically the same.

Introduced by Bennett in 1992. He showed that, in principle, any two non-orthogonal states can be used for quantum cryptography. Let $|v_0\rangle$ and $|v_1\rangle$ be two non-orthogonal quantum states encoding the "0" and "1" bits, respectively. Their multiplication is $0 < \|\langle v_0|v_1\rangle\|^2 < 1$. Alice sends Bob a randomly selected state, after which Bob randomly applies one of two incompatible design operators to him:

$$P_0 = 1 - |v_1\rangle\langle v_1| \tag{3.1}$$

$$P_1 = 1 - |v_0\rangle\langle v_0| \tag{3.2}$$

P_0 uniquely destroys $|v_1\rangle$, but gives a positive result with a probability of $1 - \|\langle v_0|v_1\rangle\|^2 > 0$ being applied to $|v_0\rangle$, and vice versa for P_0. Thus, the measurement result can be $|v_0\rangle$, $|v_1\rangle$ or ambiguous (zero can be obtained as a result of the impact of the i-th projector on the i-th state, or of any projector on the vacuum state - no photon, and all these cases cannot be distinguished). At the exchange stage through the open channel, Alice and Bob eliminate ambiguous results, and after that, in the absence of eavesdropping, approximately $(1 - \|\langle v_0|v_1\rangle\|^2)/2$ their data will be absolutely correlated.

3.3 Other Quantum protocols

3.3.1 Protocol BB84 (4+2)

This protocol brings together ideas from BB84 and B92. Bits "0" and "1" can be encoded in two bases, but the two states within one basis are not orthogonal.

3.3.2 Six-state protocol

Initially, this is the same protocol as BB84, but with another basis, namely:

$$|0_C\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \tag{3.3}$$

$$|1_C\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \tag{3.4}$$

In accordance with this, there are two more possible directions of polarization for the transmitted photon — the right- and left-circular ones.

EPR-Protocol

Eckert proposed a protocol based on quantum entanglement. First, N maximally entangled EPR pairs of photons are created, then one photon from each pair is sent to Alice, and the other to Bob. Three possible quantum states for these EPR pairs are:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|\frac{3\pi}{6}\rangle_B - |\frac{3\pi}{6}\rangle_A|0\rangle_B) \tag{3.5}$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|\frac{\pi}{6}\rangle_A|\frac{1\pi}{6}\rangle_B - |\frac{1\pi}{6}\rangle_A|\frac{\pi}{6}\rangle_B \tag{3.6}$$

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|\frac{2\pi}{6}\rangle_A|\frac{5\pi}{6}\rangle_B - |\frac{5\pi}{6}\rangle_A|\frac{2\pi}{6}\rangle_B) \tag{3.7}$$

what can be written in general terms as

$$|\psi_i\rangle = \frac{1}{\sqrt{2}}(|0_i\rangle_A|1_i\rangle_B - |1_i\rangle_A|0_i\rangle_B) \tag{3.8}$$

The last formula clearly shows that each of these three states encodes the bits "0" and "1" in a unique basis. Then Alice and Bob take measurements on their

parts of separated EPR pairs using the appropriate projectors.

$$P_1 = |0\rangle\langle 0|, P_2 = |\frac{\pi}{6}\rangle\langle\frac{\pi}{6}|, P_3 = |\frac{3\pi}{6}\rangle\langle\frac{3\pi}{6}| \tag{3.9}$$

Alice writes the measured bits, and Bob writes their additions to 1. The measurement results, in which users selected the same bases, form the raw key. For the rest of the results, Alice and Bob test the fulfillment of Bell's inequality as a test for the presence of Eve (Eve is interpreted here as a hidden parameter).

Goldenberg-Weidman Protocol Alice and Bob use two orthogonal states to communicate:

$$\psi_0 = \frac{1}{\sqrt{2}}(|a\rangle + |b\rangle) \tag{3.10}$$

$$\psi_1 = \frac{1}{\sqrt{2}}(|a\rangle - |b\rangle) \tag{3.11}$$

coding bits "0" and "1", respectively. Each of the two states $|\psi_0\rangle$ and $|\psi_1\rangle$ is a superposition of two localized normalized wave packets, $|a\rangle$ and $|b\rangle$, which Alice sends to Bob through two channels of different lengths, with the result that they end up with Bob at different times: wave package $|b\rangle$ leaves Alice only after wave packet $|a\rangle$ has already reached Bob. To do this, you can use an interferometer with different lengths of arms. Bob delays his measurement until both wave packets reach it. If the time of sending the package $|a\rangle$ is known to Eva, then she is able to intercept the information, sending Bob at the appropriate time a package identical to $|a\rangle$, then measuring the superposition state sent by Alice and then sending Bob the wave package $|b\rangle$ with the phase adjusted according to the result of its measurements. To prevent this attack, random parcel times are used.

3.3.3 Protocol Koash-Imoto [10]

This protocol is a modification of the previous one, but it eliminates random transmission times by asymmetrizing the interferometer, i.e. dividing the light in unequal proportions between short and long arms. In addition, the phase difference between the two arms of the interferometer is π. Thus, the two states

encoding bits "0" and "1" are

$$|\psi_0\rangle = -i\sqrt{R}|a\rangle + \sqrt{T}|b\rangle \tag{3.12}$$

$$|\psi_1\rangle = \sqrt{R}|a\rangle - i\sqrt{T}|b\rangle \tag{3.13}$$

where R and T are the reflectivity and transmittance of the input beam splitter, respectively. In the case of an asymmetric scheme, when the amplitude of the probability of finding a photon in one or another arm of the interferometer depends on the value of the transmitted bit, the compensation due to the phase does not work completely, and when Eve applies the above tactics, there is a non-zero probability of detection error.

3.4 Threshold of reliability ratios when using quantum technologies.

The approaches of classical and quantum systems are quite distinguishable. And the result of the process is considered to be the randomly generated key x, which was obtained using quantum states. All that a third party can do is intercept the state of the generated key x. And you can describe it using the density matrix

$$\rho_{XE} = \sum P_X(x)|x\rangle\langle x| \otimes \rho_E^x \tag{3.14}$$

In our case, x is the status register for the similarity of the classic, but with the key: $|x\rangle = |x_1\rangle \otimes |x_2\rangle ... |x_k\rangle$, $|X| = 2^k$, $x \in X = \{0,1\}^k$ ρ_E^x - the part of the matrix in which there is a relative connection with the generated key. After unauthorized access by a third party to the key, all further received data is obviously with him ρ_E^x.

QKD strives to minimize the chances of predicting the state of the generated key if a third party gets unauthorized access. And this can be achieved only by absolutely random key generation. Which will lead to a discrepancy in the results of errors, and the proof is the density matrix ρ_{XE}. Recall that in addition to its impeccable protection, the system is able to implement on the threshold of ideality of the environment and the device parameters are at the level of the minimum

allowable threshold. which means non-correlated density matrix $\rho_U \otimes \rho_E$

The trace distance is the distance between the situations:

$$\|\rho_{XE} - \rho_U \otimes \rho_E\|_1 < \varepsilon_1 \tag{3.15}$$

By definition. it is connected by a trace metric:

$$\|\rho\|_1 = \frac{1}{2} Tr\{|\rho|\} = \frac{1}{2} Tr\{\sqrt{\rho^2}\} \tag{3.16}$$

$$\rho_{XE} = \sum_{x=X} P_X(x)|x\rangle\langle x| \otimes \rho_E^x \tag{3.17}$$

$$|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \tag{3.18}$$

$$\rho_U = \frac{1}{N} \sum_{x \subset X} |x\rangle\langle x| \tag{3.19}$$

$$\rho_E = Tr_X\{\rho_{XE}\} = \sum_x^N P_x(x)\rho_E^x \tag{3.20}$$

$$\rho_X = Tr_E\{\rho_{XE}\} = \sum_x^N P_x(x)|x\rangle\langle x| \tag{3.21}$$

ρ_U – density matrix (in terms of homogeneity and ideality)

All third-party actions are just attempts to acquire the $y \in Y = \{0,1\}^n$ key states, and not the sender's x key itself, which consists of a bit string. y is the value of the connection of the third party system and the key x. which is a bit string and which depends on the analysis of the third party system ρ_E to the system of authenticated process users.. The analysis from the third party can be written down by the decomposition of the unit: $I_E = \sum_{y \in Y} M_y$. $y \in Y = \{0,1\}^n$. M_y- a positive operator-valued measure in the third-party states.

$P_{XY}(x,y)$ - joint probability distribution x. y.

$P_X(x)$ and $P_Y(y)$ - marginal probability distribution.

$P_{X|y}(X = x \vee y)$ - probability of occurrence of y. if x occurred.

$P_{X|Y}(x \vee Y = y)$ - probability of x occurring if y occurred.

Formula Bayes:

$$P_{XY}(x,y) = P_X(x)P_{X,Y}(X = x \vee y) \tag{3.22}$$

$$P_{XY}(x,y) = P_{X|Y}(x \vee Y = y)P_Y(y) \tag{3.23}$$

$$\sum_{x \in X} P_{X,Y}(x \vee Y = y) = \sum_{y \in Y} P_{X,Y}(X = x \vee y) = 1 \tag{3.24}$$

$$\sum_{x \in X} P_{XY}(x,y) = P_Y(y) \tag{3.25}$$

$$\sum_{y \in Y} P_{XY}(x,y) = P_X(x) \tag{3.26}$$

The situation in which it is conditionally probable that the authenticated users have the key x generated among themselves, at the same time its copy is in the third party after the unauthorized analysis by the third party is equal to y:

$$P_{X,Y}(X = x|y) = \{M_y \rho_E^x\} \tag{3.27}$$

According to which we reproduce the summation:

$$\sum_{y \in Y} P_{X|Y}(X = x|y) = 1 \tag{3.28}$$

Probability of a positive result for a third party:

$$P_{X|Y}(X = x|y) = \{M_x \rho_E^x\}, x = y \tag{3.29}$$

According to the laws of probability theories, we can safely say that the probability of a positive outcome for a third party in quantum cryptography does not exceed

$$P_{Guess}(X|Y) = \max_M \sum_{x \in X} P_X(x) Tr\{\rho_E^x M_x\} =$$
$$\sum_{x \in X} P_X(x) P_{X|Y}(X = x|x) = \sum_{x \in X} P_{XY}(x,x) \tag{3.30}$$

$P_{XY}(x,y)$ - joint probability distribution of authenticated users (x) and third parties (y), which does not exceed:

$$P_{Guess}(X|E) = \sum_{x \in X} P_{XY}(x,x) \leq \frac{1}{N} + \|\rho_{XE} - \rho_{XE} \otimes \rho_E\|_1 < \frac{1}{N} + \varepsilon, N = 2^n \tag{3.31}$$

ε - is the privacy setting of authenticated users, which is obtained by clearing the generated state.

3.5 No-cloning theorem

Is it possible to make a copy of an unknown quantum state? Surprisingly, it turns out that the answer to this question is no. In this box we describe an elementary proof of this fact that captures the essential reason this is not possible.

Suppose we have a quantum machine with two slots labeled A and B. Slot A, the *data slot*, starts out in an unknown but pure quantum state, $|\psi\rangle$. This is the state which is to be copied into slot B, the target slot. We assume that the *target slot* starts out in some standard pure state, $|s\rangle$. Thus the initial state of the copying machine is

$$|\psi\rangle \otimes |s\rangle \tag{3.32}$$

Some unitary evolution U now effects the copying procedure, ideally,

$$|\psi\rangle \otimes |s\rangle \rightarrow U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{3.33}$$

Suppose this copying procedure works for two particular pure states, $|\psi\rangle$ and $|\phi\rangle$. Then we have

$$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{3.34}$$

$$U(|\phi\rangle \otimes |s\rangle) = |\phi\rangle \otimes |\phi\rangle \tag{3.35}$$

Taking the inner product of these two equations gives

$$\langle e|o\rangle = (\langle e|o\rangle)^2 \tag{3.36}$$

But $x = x^2$ has only two solutions, x 0 and x 1, so either $|o\rangle = |e\rangle$ or $|e\rangle$ and $|o\rangle$ are orthogonal. Thus a cloning device can only clone states which are orthogonal to one another, and therefore a general quantum cloning device is impossible. A potential quantum cloner cannot, for example, clone the qubit states $|e\rangle = |0\rangle$ and $|o\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, since these states are not orthogonal. What we have shown is that it is impossible to perfectly clone an unknown quantum state using unitary evolution. The short summary of this work is that even if one allows non-unitary cloning devices, the cloning of non-orthogonal pure states remains impossible unless one is willing to tolerate a finite loss of fidelity in the copied states. Similar conclusions hold also for mixed states, although a somewhat more sophisticated approach is necessary to even define what is meant by the notion of cloning a mixed state.

3.6 Threat for QKD with a transfer of a modified condition

The quantum world is complex from the point of view of the laws of ordinary mechanics, since these laws are limited in the expanses of quantum physics. One of the main limitations of these laws is the immeasurability of quantum states. The limitations of the laws and is the main fundamental link, which makes quantum cryptography one of the most secure systems. There are many ways to take advantage of quantum physics in systems. One of the most realized uses of quantum technologies is the creation of a randomly generated secret key for encryption. There is a vulnerability of the system, due to the imperfection of the devices used. For example, one of the fundamental conditions for the application of quantum technologies in cryptography is the creation of a pure single-photon source. This difficulty will have to be solved in the future when more advanced devices appear. Today, multi-photon devices are used, which, as the study shows, leave unauthorized access.

The quantum key distribution protocol usually means authentication, pre-process

preparation, transfer, examination of states, handling errors (eliminating or enhancing secrecy using the compression method), analyzing and verifying the obtained keys. The data transmitted by the system must be absolutely protected from third parties and, if necessary, disclose the audition. When unauthorized access to the transmitted data occurs, the system should detect this attempt using the ratios of valid and resulting error. In other situations, the system is considered unprotected when unauthorized access by a third party goes unnoticed and the key is known to it. For several years, research has been conducted in which the capabilities of quantum cryptography have been demonstrated[17, 16].

Currently, in quantum systems, attenuated laser radiation is used as a source of a single-photon state (which is not purely single-photon), single-photon avalanche photo detectors (which has dark noise), efficiency (whose properties are not rare), communication channels (fiber optic and open space), which are subject to loss and noise. This in turn leads to the emergence of the possibility of a photon-splitting system attack (PNS, Photon Number Splitter attack[3]) and an attack with measurements with a certain outcome (Unambiguous Measurements [7, 1]). We included these factors in the analysis of the security of protocols [15]. Unauthorized access to a key using photo-detector blinding [12] is one of the new threats to quantum key distribution. In this threat, a third party uses the possibility of affecting a communication channel in which it sends a modified (falsified) state. This allows a third party to control the counts or their absence at the receiving side, and impose their own counting, which does not lead to an error at the recipient (receiving side). As a result, the third party remains unnoticed and informed about the key. This new method of threat significantly reduces the secrecy of the system. Most of the protocols, with no additional parameters, were potentially not resistant to this threat (BB84[5], SARG04, Six-State QKD[6], DPS (Differential Phase Shift)[8, 10] , COW (Coherent One Way)[8], E91[2], Decoy State QKD[9].

One way to solve this problem is to create a strictly single-photon source, which at the present time has proved difficult to implement. Until this difficulty is resolved, the threat will always exist. To date, the search for solutions to these problems. From the basic solutions, it is proposed to complicate and add additional parameters to existing protocols. The proposed solution does not solve the problem, but only complicates access by a third party. The methods of the third

party in turn become cleverer. The best way to create an over-secure system is to create a protocol in which security is achieved through internally-structured methods, rather than an improvement and addition in technical terms[11].

3.6.1 Threat with photon modification

Below we provide information about this new threat on the host device, while not trying to look at all this in terms of technical vision, since all this is in the articles[12, 11, 13]. The more devices are not perfect, the some ways to use these imperfections. One of these ways to influence the receiving side: 1. On avalanche photo-detectors, there is a difference in the sensitivity of temporal dependencies. This allows a third party to modify (falsify) the state and blind the devices of the receiving party. 2. Using the state of the lower threshold of intensity, in order to include only one device, in situations where the bases of the third party and the recipient will be the same. Otherwise, everything happens without counting. Basically, the possibility of third-party unauthorized access arises from the work of semiconductors (the so-called avalanche photo-detectors InGaAs:P). A blocking voltage is applied to the avalanche photo-diode. At the moment of arrival of the photon, a gate voltage pulse (typical duration of the order of several nanoseconds and an amplitude of several volts) is applied, which opens the photo-diode. The absorption of a photon leads to the formation of an avalanche of carriers and a voltage pulse, which is recorded. 1. The first blinding attack is based on the following property. If the radiation intensity is increased slightly above the quasi-single-photon mode, this will lead to an increase in the current flowing through the photo-diode at the time of registration. Due to the fact that a photo-diode without illumination is not active (locked), the dynamic increase in current above a certain value will lead to an effective decrease in the bias voltage during the action of the strobe and to lock the diode and, accordingly, decrease, up to its absence, the avalanche current (" no click "). Thus, the photo-diode is effectively blinded. 2. The second attack is connected with the transfer of the photo-diode from the counting mode to the linear classical photo-detection mode, when the current is a function of the radiation intensity (usually proportional to it). If we further increase the radiation intensity, then after blinding the photo-diode (see paragraph 1 above) and the absence of the registration current, starting with a certain threshold intensity value, a signal will appear again on the photo-diode

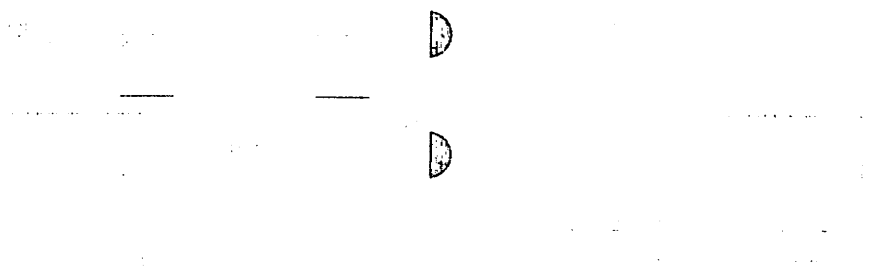


Figure 3.2: a) Schematic diagram of the receiving part of the fiber optic quantum cryptography system implementing the BB84 protocol. QC - quantum canal, Pm - phase-modulator. b) Photo-count statistics in two detectors for different information states.

through the photodiode a current proportional to the radiation intensity will flow[4].

3.6.2 The use of the difference in the sensitivity

The first protocol in quantum cryptography BB84[5] is in any way well suited to explain the main points of opposition to the threats to which some protocols are exposed and their strong and weak aspects[5, 12, 14, 13]. States in basis $-$ from quantum channel:

$$|0_+\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle) \tag{3.37}$$

$$|1_-\rangle = \frac{1}{\sqrt{2}}(|1\rangle - |2\rangle) \tag{3.38}$$

and in the basis of $\times$.

$$|0_\times\rangle = \frac{1}{\sqrt{2}}(|1\rangle + i|0\rangle) \tag{3.39}$$

$$|1_\times\rangle = \frac{1}{\sqrt{2}}(|1\rangle - i|2\rangle) \tag{3.40}$$

where $|1\rangle$ and $|2\rangle$states localized in the time windows 1 and 2. The distance between the localized states is equal to the difference in travel along the upper and lower arms of the unbalanced interferometer (Fig.3.2) In the process of entering

the receiving device. taking into account the selected phase. the quantum state in basis – have the form

$$|0_{-}\rangle \rightarrow \frac{1}{\sqrt{8}}(|1\rangle + (1 + e^{i\varphi B(+)})|2\rangle + |3\rangle) \tag{3.41}$$

D_0 :

$$|1_{-}\rangle \rightarrow \frac{1}{\sqrt{8}}(|1\rangle + (-1 + e^{i\varphi B(+)})|2\rangle + |3\rangle) \tag{3.42}$$

$$|1_{+}\rangle \rightarrow \frac{1}{\sqrt{8}}(-|1\rangle + (1 - e^{i\varphi B(-)})|2\rangle + |3\rangle) \tag{3.43}$$

D_1 :

$$|0_{+}\rangle \rightarrow \frac{1}{\sqrt{8}}(-|1\rangle + (1 - e^{i\varphi B(-)})|2\rangle + |3\rangle) \tag{3.44}$$

in basis of $\times$ (value $= \varphi_B(\times) = \pi/2$) -

$$|0_{\times}\rangle \rightarrow \frac{1}{\sqrt{8}}(|1\rangle + (i + e^{i\varphi B(\times)})|2\rangle + |3\rangle) \tag{3.45}$$

D_0 :

$$|1_{\times}\rangle \rightarrow \frac{1}{\sqrt{8}}(|1\rangle + (-i + e^{i\varphi B(\times)})|2\rangle + |3\rangle) \tag{3.46}$$

$$|0_{\times}\rangle \rightarrow \frac{1}{\sqrt{8}}(-|1\rangle + (-i - e^{i\varphi B(-)})|2\rangle + |3\rangle) \tag{3.47}$$

D_1 :

$$|1_{\times}\rangle \rightarrow \frac{1}{\sqrt{8}}(-|1\rangle + (i - e^{i\varphi B(+)})|2\rangle + |3\rangle) \tag{3.48}$$

If the bases on the receiving and transmitting sides of the state coincide.

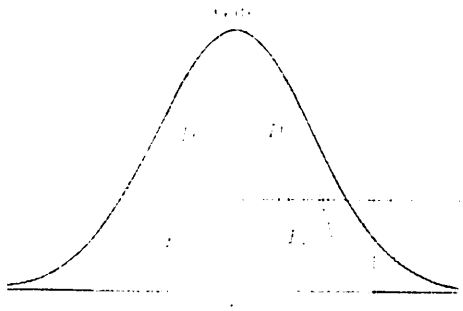


Figure 3.3: An example of temporal dependences of the sensitivity of avalanche photo-detectors (D_1. D_0). gating pulse (Gate) and falsified quantum states (F_0.F_1).

corresponding to 0 in both bases, they will give readings in time window 2 (Fig. 3.2) only in the D_0 detector. Accordingly, the states corresponding to 1 in both bases will give counts only in the D1 detector[16].

However, the avalanche photodiodes D_0 and D_1 have different temporal characteristics of sensitivity (Fig.3.3). The threat comes down to the following. The third party breaks the quantum communication channel (optical fiber) and carries out measurements similar to those on the receiving side in a randomly chosen basis. There are two options. Immediately, we note that the third party sends its modified state in the basis opposite to the one in which it carried out its measurements. 1) With the coincidence of the bases of the third party and the authenticated participants. (for example. "−"). Assuming that after the analysis, the third party received a "0", i.e. the correct result, but the third party does not know about it until the end of the process between the sender and the recipient, and therefore she prepares the falsified state "1" in the opposite basis "×", but more localized (narrow) and slightly shifted in time so that it does not fell into the sensitivity curve of the photo-detector D_1 (Fig. 3.3). And the third party re-sends:

$$|1_{F_0}\times\rangle = \frac{1}{\sqrt{2}}(|1_{F_0}\rangle - i|2_{F_0}\rangle) \tag{3.49}$$

After passing through the interferometer and phase modulator with the phase in the basis + (F), the states in front of the entrance to the photo-detectors are:

$$D_0: \qquad \frac{1}{\sqrt{8}}(|1_{F_0}\rangle + (i+1)|2_{F_0}\rangle + 3_{F_0}\rangle) \tag{3.50}$$

$$D_1: \qquad \frac{1}{\sqrt{8}}(-|1_{F_0}\rangle + (i-1)|2_{F_0}\rangle + 3_{F_0}\rangle) \tag{3.51}$$

The falsified state does not fall in time in the sensitivity curve of the photodetector D_0. and will not give an erroneous reading of the detector D_1 in the central time window 2 (Figure 3.2.3.3). As a result. it turns out an error-free reading in the D_0 detector. It does not matter that the number of counts in the detector D_0 and the ratio of the false state and the correct one decrease. I would like to mention that the error ratio is influenced not only by device perfection and the appearance of unauthorized access. but also due to the loss in the channel itself, which no longer depends on the technological component of the system. 2) If there is no coincidence of the bases of the third party and the authenticated participants. In this case. which would not be the outcome. the probability of a chance of getting the correct result for a third party is 50%. Given that if a third party does not apply the correct basis for the analysis, it will spoil not only the result. but also the state itself. 2a) suppose that the third party guessed the basis of the analysis. Then the third party re-sends the modified state in the wrong state. which remains unnoticed in time in the sensitivity curve of the receiving device D_0 (formula 6). This will not result in a readout in the D_1 host device. And the ratio of errors will be in the normal range. 2b) suppose that the third party did not guess the basis. The same as in the first case, everything repeats. except that the action remains noticed and will be shifted in time, but does not involve the rest of D_0:

$$|0_{F_1}\times\rangle = \frac{1}{\sqrt{2}}(|1_{F_1}\rangle + i|2_{F_1}\rangle) \tag{3.52}$$

The correct state transforms the count after the D_0 sensitivity of the detector falls into the curve. but this does not happen because the modified state is shifted in time. And this leads to the fact that on D_0. produces a count and errors in D_1. All this is caused by the interference of the state. which constructively extends over the different arms of the interferometer for D_0. and is extinguished on the D_1 state. along the upper and lower arms. And this whole process not only allows the third party to remain unnoticed in relation to the recipient (reducing the error to a valid one). but also in the end to intercept real information about the registered keys. which the recipient will not suspect. In the real world. of course.

it is difficult to carry out this threat, but it is quite provable and feasible. When it comes to analyzing the strength limit of any cryptographic system, it is taken into account that the third party will have the most advanced equipment and will have an ideal condition for implementing unauthorized interception and access to an encrypted and secure channel.

3.6.3 Threat to key with transfer from one mode to another

This threat consists of stages in which a third party re-sends the state of the sender (intercepted in the quantum channel). In this case, the third party after analyzing in a random basis and increases the intensity so as to transfer the detector to the classic mode. This thin line between the modes is retained by the method of insufficient intensity, which is used so as not to lose the blinding mode.[12, 14, 13])

There are moments:

1. With the coincidence of third party bases and authenticated participants. This leads to a complete constructive interference, the reason for this is to capture the detector full intensity, and that leads to the count in time.

2. If there is no coincidence of the bases of the third party and the authenticated participants. In this case, the detector is transferred to the blinding mode, the cause of which is not enough intensity in time. The intensity with which the detector exits the blinding mode registers the signal as a linear device, and is equal to I_{th}, and the intensity with which the detector is blinded (lies in the "no click" zone) is equal to I_{bl}. ($I_{bl} < I_{th}$).

The intensity of the falsified state is equal to I_{facked}. The intensity in the side time windows (see Fig. 2) does not depend on the choice of third party bases and authenticated participants and is equal to $I_{facked}/8$ (see Fig.3.3 and formulas (3) - (7)). In this case, two situations are possible: a) The intensity of I_{facked} is such that in the side time windows, where the intensity does not depend on constructive or destructive interference, it is equal to $.I_{facked}/8$. At the same time, . $I_{facked}/8$ is obviously less than the threshold intensity I_{th}, at which the detector works as linear, but more than the intensity , which causes the blinding effect: $I_{th} < I_{facked} =< I_{th}$. There will be no counts in the side time windows (Fig. 3.2).

b) The intensity I_{facked} in the side time windows is equal to $I_{facked}/8$, less than the threshold intensity I_{th}, and the intensity I_{bl} causing the blinding effect:

$I_{facked}/8 < I_{bl}, I_{th}4$. In this case, counts in side time windows (Fig. 3.2) will take place. The detector is not blinded and works as a single-photon counting mode. The whole point of the threat from the third party comes down to what she is trying to guess, but at the same time minimizes the ratio of permissible and received errors as a result of the distribution process, if attempts to guess will be in vain, and she will reveal herself about her presence.

4. Development and implement quantum encryption methods for key generation

4.1 Make source code for MatLab

We use quantum protocols BB84 for create a key. Despite this. the key must comply with several laws of cryptographic reliability: 1) It must be absolutely random. 2) It must not be applied twice, 3) And the key length must not be shorter than the message itself. For classical algorithms. there is a problem that is partially solved - this is the generation of an absolutely random number. How would not complicate the mathematical process, in technical terms, it remained not perfect. Only with the use of quantum technologies this became possible. This is one of the main advantages of the quantum world. We tried to combine quantum methods with classical ones. One of the sought-after classic areas that needs a symmetric key creation is the creation and exchange of tokens for authentications. The use of the token has recently reached a new level with the advent of cryptocurrency. The relevance and security of tokens is the fundamental direction of the development of cryptosystems. Quantum methods give a new opportunity to use symmetric encryption algorithms. By default. the length of our key should not be less than 256 bits to comply with the third law of Shannon. Therefore. we will generate the key with at least 576 qubits. We have yet to analyze in the future what time interval is optimal and maximum use of the key. after which we will have to restart the generation.

We will try to show step by step how we generate the key between Alice and Bob

Basis:

$$B^{+} = \{\epsilon_0^{-} = \begin{pmatrix}1\\0\end{pmatrix}, \epsilon_1^{-} = \begin{pmatrix}0\\1\end{pmatrix}\} \tag{4.1}$$

$$B^{\times} = \{\epsilon_0^{\times} = \frac{\epsilon_0^{+} + \epsilon_1^{-}}{\sqrt{2}}, \epsilon_1^{\times} = \frac{\epsilon_0^{-} - \epsilon_1^{+}}{\sqrt{2}}\} \tag{4.2}$$

Alice implementing operation $T : \{0.1\}^2 \to H$

$$T(x.y) = \begin{cases} \epsilon_0^{+} & \text{if}(x.y) = (0,0). \\ \epsilon_1^{+} & \text{if}(x.y) = (1,0). \\ \epsilon_0^{\times} & \text{if}(x.y) = (0,1). \\ \epsilon_1^{\times} & \text{if}(x.y) = (1,1). \end{cases} \tag{4.3}$$

$$\|T(x.y)\| = 1 \tag{4.4}$$

Alice generated bits and basis randomly for use this in key generation

Require: Random Generator (0.1). T, n

Ensure: Strings $a.\ b \in \{0,1\} nandsequence(\{\ v\})_{i=1.....n}$

We randomly generate bits of information $a_1, \ldots, a_n$

$a \leftarrow (a_1, \ldots, a_n) \in \{0.1\}_n$

Then randomly generate basis for encode bits $b_1, \ldots, b_n$

$b \leftarrow (b_1, \ldots, b_n)$

locally store our bits a and list of basis b

repeat

$|v_i\rangle \leftarrow T(a_i.b_i)$

transmit $|v_i\rangle$ to Bob via quantum channel

$i \leftarrow i + 1$

until $i > n$

Here: n - number of qubits a - bits of information b - basis that Alice using for encode bits $|v_i\rangle$ - encoded state of qubits

Bob received encoded state of qubits and try decode with help basis, which he choose randomly

In Bob side we require random generator ($\{\ 0.1\ \}$) for $M = |\epsilon_1\rangle\langle\epsilon_1|$. for $\in \{+, \times, \}$. $n.sequence|\ v_i\rangle$, for $i, \ldots, n$

After process Bob will have two strings of n bits a'. $b' \in \{ 0.1 \}_n$

Randomly generate list of basis $b'_1 \ldots . b'_n$

$b' \leftarrow (b'_1 \ldots . b'_n) \in \{0.1\}_n$

$n \leftarrow 1$

Repeat

if $b'_i = 0$ then

Ask whether M_+ takes value 1 in state $|\psi_i\rangle$

else

Ask whether $M_\times$ takes value 1 in state $|\psi_i\rangle$

end if

if counter triggered then

$a'_i \leftarrow 1$

else

$a'_i \leftarrow 0$

end if

$i \leftarrow i + 1$

until $i > n$

$a' \leftarrow (a'_1, \ldots, a'_n) \in \{0, 1\}_n$

Bob transmit string $b' \in \{0, 1\}_n$ to Alice via public classical channel All information and result Bob store locally $a'.b'$ M - list of basis, with the help of measure Alice's qubits b' - one basis of Bob a' - result of Bob, which he obtain after measure

After measure Bob sent list of basis to Alice and she compares bases, and discards all the mismatched

Alice have $b.b' \in \{ 0,1 \}$

In this situation we ensure that sequence $(k_i \ldots . k_L)$(with $L \leq n$) of positions of coinciding bits

$c \leftarrow b \otimes b_i$

$i \leftarrow 1$

$k \leftarrow 1$

Repeat

$k \leftarrow min\{j : k \leq j \leq n$ such that with $j=0 \}$

if $k \leq n$ then

$k_i \leftarrow k$

$i \leftarrow i+1$

end if

until $k > n$

$L \leftarrow i-1$

Alice transmit $(k_1, \ldots, k_L)$ to Bob via public classical channel

If no eavesdropping on quantum channel

$P(a'_{k_1}, \ldots, a'_{k_L}) = (a_{k_1}, \ldots, a_{k_L}) = 1$

a_i	b_i	ψ_i	b'_i	$\langle \psi_i \vert M_+ \psi_i \rangle$	a'_i	b'_i	$\langle \psi_i \vert M_\times \psi_i \rangle$	a'_i
0	0	ϵ_0^+	0	0	0	1	1/2	0 or 1
1	0	ϵ_1^+	0	1	1	1	1/2	0 or 1
0	1	$\epsilon_0^\times$	0	1/2	0 or 1	1	0	0
1	1	$\epsilon_1^\times$	0	1/2	0 or 1	1	1	1

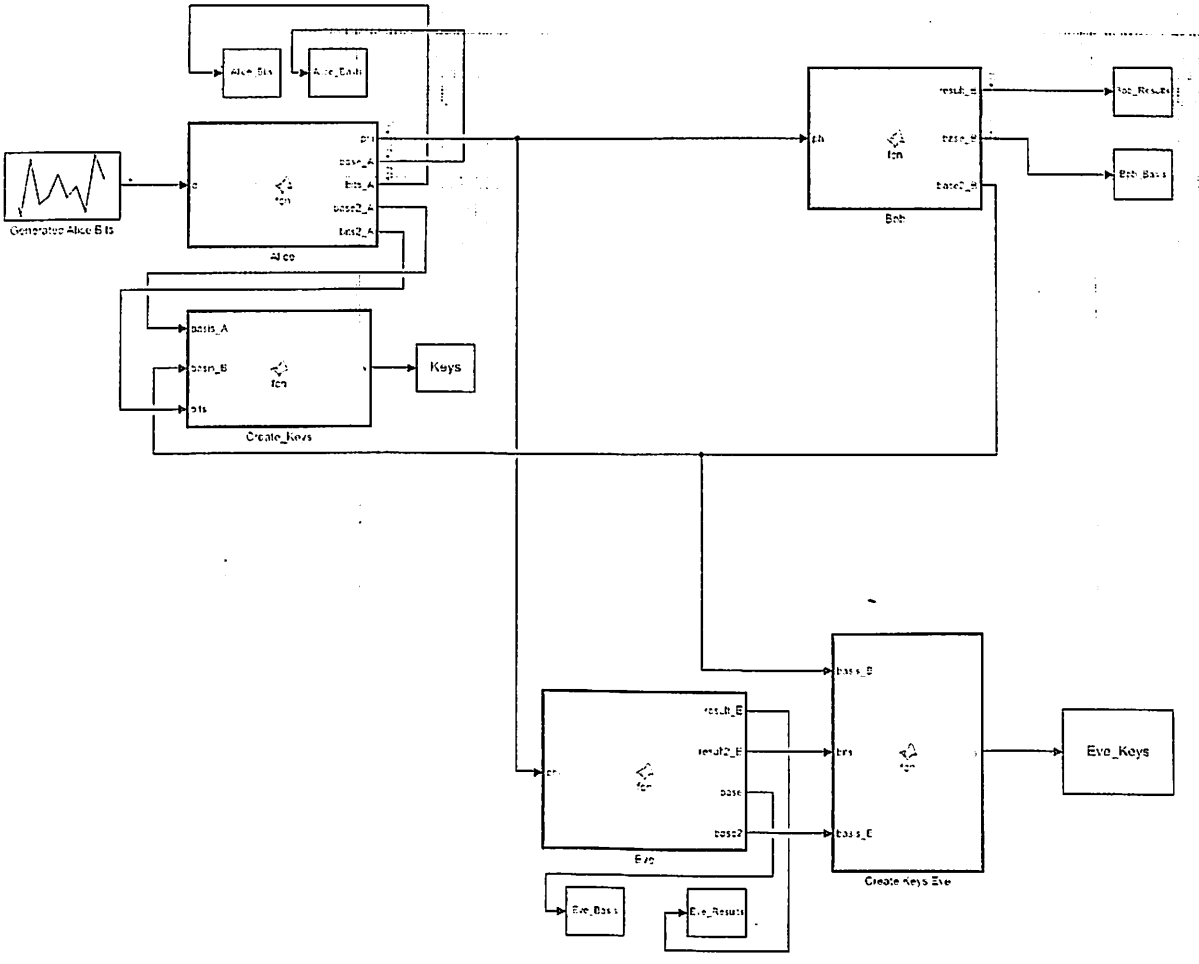
if $b'_i = b_i$ then $P(a'_i = a_i)$=1 Certainty on coincidences although a's never exchanged

If no intrusion, Alice and Bob will use a — sampled at places of coincidence — as key because

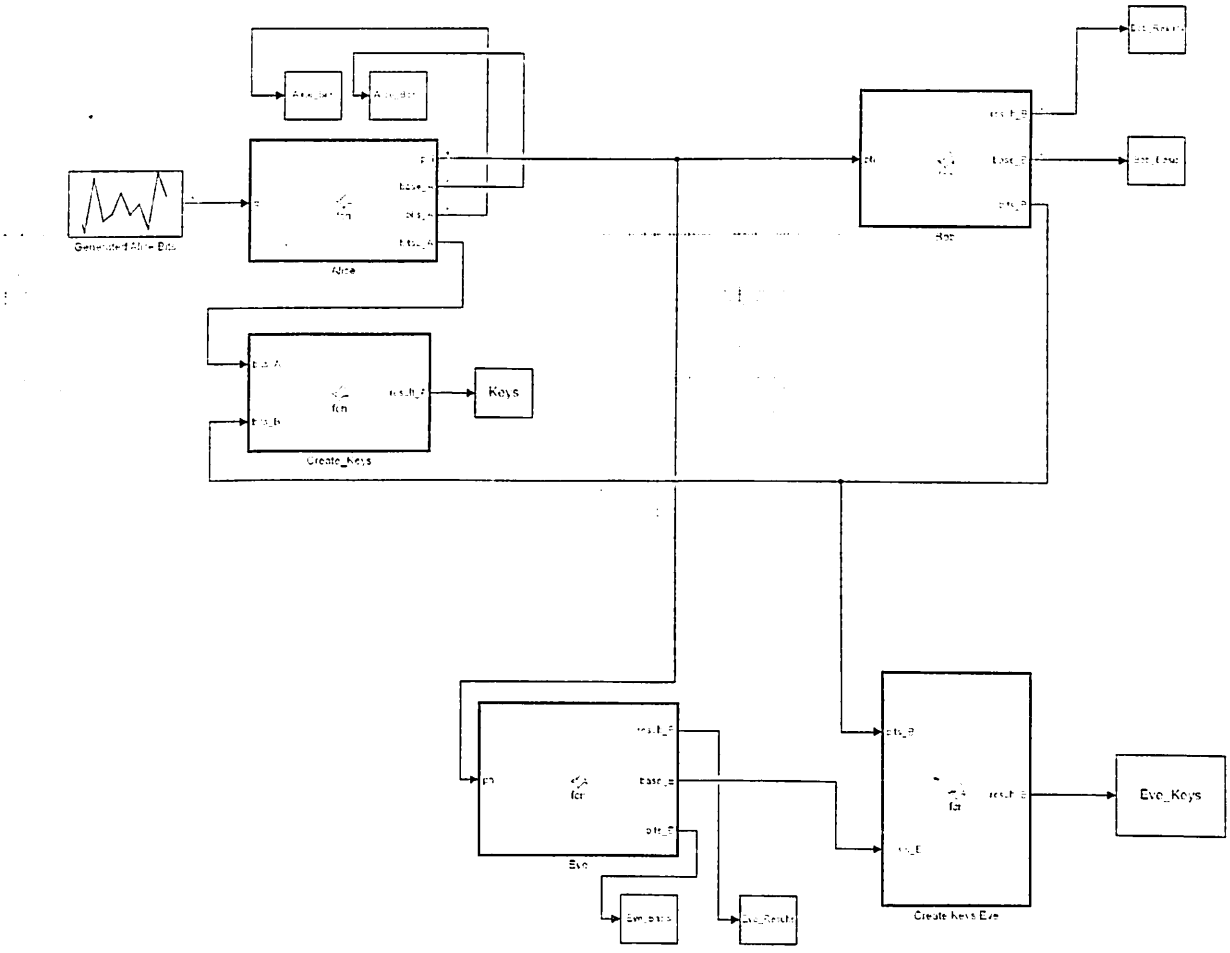
$$(a'_{k_1}, \ldots, a'_{k_L}) = (a_{k_1}, \ldots, a_{k_L})$$

Code in Appendix A.B

4.2 Create Simulink model based on quantum encryption



Protocol BB84



Protocol B92

4.3 Quantum encryption methods for token generation

When we have a secret key, is to choose an algorithm for encryption. There are two types of algorithms, symmetric and asymmetric. An asymmetric algorithm is an encryption method that uses two types of key, public and private. The public key is used to encrypt a message, and with a private key to decrypt it. At the same time, the public is accessible to everyone, but the private is not. There are several types of asymmetric ciphers: RSA, DSA, Elmagal, Diffie-Hellman, ECDSA, etc. A symmetric algorithm is a more intuitive encryption cycle. In this method there is only one key, with which encryption and decryption is performed. And the biggest problem in this algorithm was to implement a secure key exchange method between the two parties. The advantages of a symmetric algorithm over an asymmetric one are that it is simpler and it is allowed for the key length to

be less than the message itself. There are several types of symmetric algorithms: AES, DES, 3DES, PC2, PC5, BLOWFISH, TWOFISH, NUSH, IDEA and so on. In our case, to create a token, we will use an algorithm that is used by default, this is the HS256 algorithm.

Using the BB84 protocol, we generated a secret key between Alice (SECRET-KEY-ALICE) and Bob (SECRET-KEY-BOB). Now we can use this key to start the authentication process. Authentication implies the use of tokens, which contain all the necessary information for this, but all this information is encrypted using a secret key. The entire process of creating tokens we will carry out with the help of the Python libraries. In our case, the "PyJwt" library will be suitable for this.

After we have chosen the algorithm for encryption and have generated the secret key, it remains for us to fill in the payload with the information that we need to send in the encrypted form.

Listing 4.1: Python code for choose data and parameters

```
SECRET_KEY_ALICE = '10CF97807AA5C490EF4995682E1
--------------------994AEFEFFF9D497305BBB1AD8BE
--------------------08D570D559DA158C4A'
ALGORITHM = 'HS256'

EXP_SECONDS = 30
user_id = 7

payload = {'user_id':user_id,
        'exp':datetime.utcnow() + timedelta(seconds=EXP_SECONDS),
        'name':'John_Smith',
        'sub':'1234567890',
        'iat':'1516365222'
        }
```

Once we have filled out all the necessary information, with the help of the library we will begin the process of encryption:

Checking token after encode:

Results:

Listing 4.2: Python code for encode data

```
Token_encoded = jwt.encode(payload,
                           SECRET_KEY_ALICE,
                           algorithm=ALGORITHM)
```

```
print('Token:_', Token_encoded)

Token:  b'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
________eyJ1c2VyX2lkIjo3LCJleHAiOjE1NTM1NzEzMjUsIm
________5hbWUiOiJKb2huIFNtaXRoIiwic3ViIjoiMTIzNDU2
________Nzg5MCIsImlhdCI6IjE1MTYzNjUyMjIifQ.
________yLL93XgDgvvWwDq7a8aOnjXrLBWnlZvSzZohPHuBqik'
```

Bob, having received the token, starts the decryption process with his secret key, which he and Alice generated

Checking results after decode

Bob results:

Listing 4.3: Python code for decode data

```
SECRET_KEY_BOB = '10CF97807AA5C490EF49956S2E1
__________________994AEFEFFF9D497305BBB1AD8BE
__________________08D570D559DA158C4A'

Token_decoded = jwt.decode(Token_encoded,
                           SECRET_KEY_BOB,
                           algorithms=ALGORITHM)

print('Token_which_Bob_decode:_', Token_decoded)

Token which Bob decode:  {'user_id': 7, 'exp': 1553871325,
    'name': 'John_Smith',
    'sub': '1234567890',
    'iat': '1516365222'}
```

4.4 Analysis of the results in the formation of tokens

Two cases were modeled, in the first case the moment when the third party has the ability to copy the qubit is recreated, although this is impossible, and it is precisely this impossibility that the flawless protection of quantum cryptography lies. The opportunity arises from the fact that the equipment does not fully meet and respond and somewhere it is not yet possible to recreate all the requirements of quantum states. In simulations, qubits with a length of 20,30,40 were modeled and they were generated 20 times each, and as a result of this simulations, results

were obtained in which a third party with different indices was able to intercept an average of 50% of the generated key, as shown in appendixes D, and the graph is on average drawn relatively by a line, which is the same regardless of the length of the qubit.

Alice and Bob keys:
Eve results:

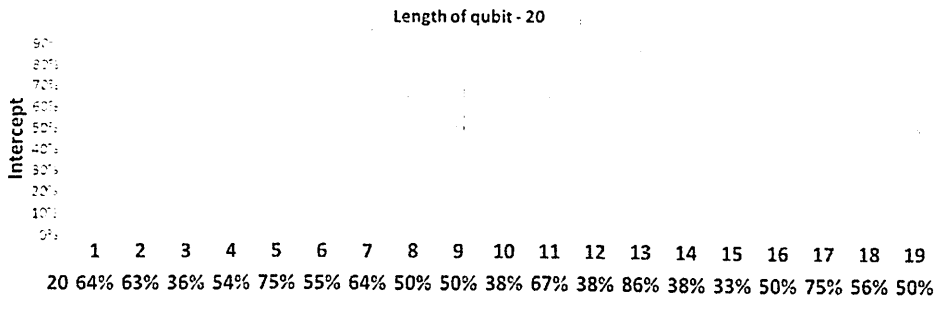


Figure 4.1: Case 1: Length of qubits 20

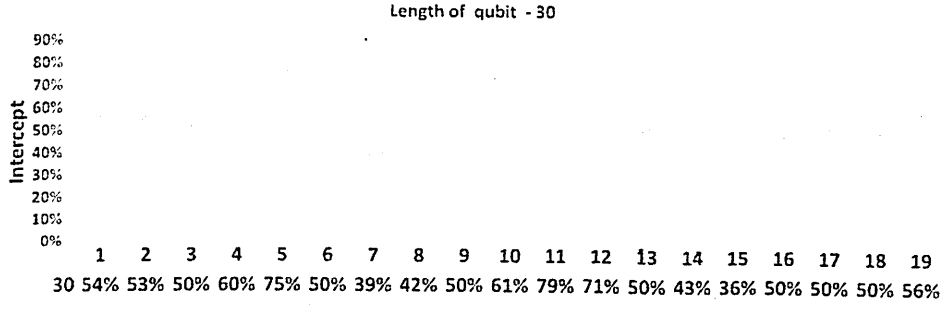


Figure 4.2: Case 1: Length of qubits 30

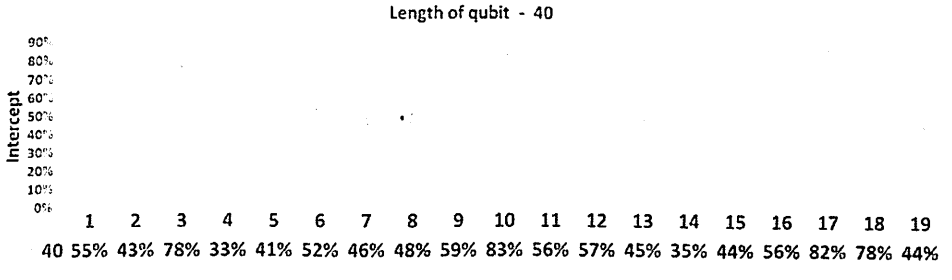


Figure 4.3: Case 1: Length of qubits 40

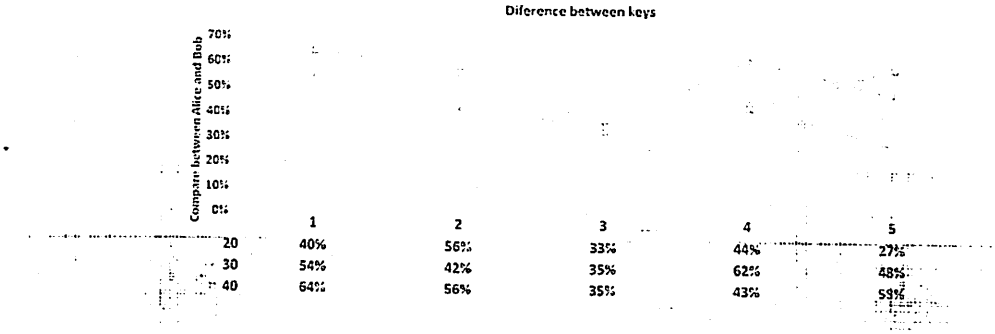


Figure 4.4: Graphics of Case 2

In the second case, a case is simulated that meets all the requirements of quantum conditions, and during simulations the results were obtained, demonstrating that with ideal equipment, quantum encryption using the quantum state makes it possible to detect the presence of a third party, as shown in the graph, and in this simulation qubits with a length of 20.30.40 and 20 times were also used.

5. Conclusion

Quantum encryption is one of the most popular trends in recent times, and the importance of these studies will grow as the appearance of quantum computers approaches.

In my theses there was an attempt to study and model the quantum state on Matlab. In simulations. two cases were simulated: In the first. a case was demonstrated when not perfect equipment was used to create and transmit qubits. which makes it possible to intercept information without violating the integrity of the encrypted information. At the same time, the third party manages to intercept only 50% of the generated key. but with different indices. In the second. the ideal case for quantum states was simulated. with the main analysis being introduced to the detection of a third party. The simulation revealed that a third party was identified with the errors it generated.

After modeling during the creation of tokens. results were obtained that demonstrate that the use of technological equipment. which does not fully meet the requirements of quantum mechanics and is not able to perfectly recreate the conditions and conditions of quantum physics. leaves vulnerabilities that attackers can use with very technological equipment. since not all the conditions of a quantum are met. But for quantum technologies. it develops very quickly and the main thing the world needs in these technologies, and the discrepancy in the technological aspect is a matter of time. which will be solved in the next decade.

References

[1] Chefles A. "Quantum state discrimination". In: *Journal Contemporary Physics* 6 (2010). pp. 401–424.

[2] Ekert A. "Quantum Cryptography Based on Bell's Theorem". In: *Physical Review Letters* 6 (1991). pp. 661–663.

[3] Mor T. Brassard Gilles Lutkenhaus N. "Limitations on practical quantum cryptography". In: *Physical Review Letters* 85 (2000). p. 1330.

[4] Bennett Charles. "Quantum cryptography using any two nonorthogonal states". In: *Physical Review Letters* 68 (1992). pp. 3121–3124.

[5] Bennett Charles and Brassard Gilles. "Quantum Cryptography: Public-Key Distribution and Tossing." In: *Proceedings of IEEE International Conference on Computers,Systems and Signal Processing, Bangalore, India, IEEE Press* (1984). pp. 175–179.

[6] Bruss D. "Optimal eavesdropping in quantum cryptography with six states." In: *arXiv:quant-ph/9805019* (May 7, 1998).

[7] Dieks D. "Overlap and distinguishability of quantum states". In: *Physical Review Letters* 126 (1988). p. 303.

[8] Tittel W. Gisin N. Ribordy G. "Quantum cryptography". In: *Physical Review Letters* 74 (2002). pp. 145–190.

[9] Won-Young H. "Observation of a Broad structure". In: *Physical Review Letters* 057901-1 (2003). p. 95.

[10] Yamamoto Y. Inoue K. Waks E. "Differential Phase shift quantum key distribution". In: *Physical Review Letters* 037902 (2002). p. 89.

[11] Kulik C. Kolokov A. Katamadze G. "On the passive probing of fiber optic quantum communication channels". In: *Journal of Experimental and Theoretical Physics* 137 (2010). p. 637.

[12] Wittmann C. Lydersen L. Wiechers C. "Hacking commercial quantum cryptography systems by tailored bright illumination". In: *Nature Photonics* 4 (2010), p. 686.

[13] Wittmann C. Lydersen L. Wiehers C. "Thermal blinding of gated detectors in quantum cryptography". In: *arXiv:quant-ph 1009.2663* (September 14. 2010).

[14] Sauge S. Makarov V. Anisimov A. "Controlling an actively quenched single photon detector with bright light". In: *arXiv:quant-ph 0809.3408* (September 19. 2008).

[15] Bechmann-Pasquinucci H. Scarani V. "The security of practical quantum key distribution". In: *Physical Review Letters* 6 (2009). p. 1301.

[16] Weisner Stephen. "Conjugate Coding". In: *ACM SIGACT News (New York)* (1983), pp. 78–88.

[17] Wootters William and Zurek Wojciech. "A Single quantum cannot be cloned". In: *Nature* 299 (1982), pp. 802–803.

A. Appendix A

Listing A.1: Insert code directly in your document

```
% plot Bloch Sphere

[Xs, Yx, Zx] = sphere(25);
mySphere = surf(Xs, Yx, Zx);
axis equal
shading interp
mySphere.FaceAlpha = 0.25

line([-1 1], [0 0], [0 0],'LineStyle',':', ...
            'LineWidth', 1, 'Color', [0 0 0])
line([0 0], [-1 1], [0 0],'LineStyle',':', ...
            'LineWidth', 1, 'Color', [0 0 0])
line([0 0], [0 0], [-1 1],'LineStyle',':', ...
            'LineWidth', 1, 'Color', [0 0 0])

text(0, 0, 1.1, '$\left|␣0␣\right>$', 'Interpreter', 'latex', ...
    'FontSize', 20, 'HorizontalAlignment', 'Center')
text(1.1, 0, 0, '$\left|␣+␣\right>$', 'Interpreter', 'latex', ...
    'FontSize', 20, 'HorizontalAlignment', 'Center')
text(-1.1, 0, 0, '$\left|␣-␣\right>$', 'Interpreter', 'latex', ...
    'FontSize', 20, 'HorizontalAlignment', 'Center')
text(0, 0, -1.1, '$\left|␣1␣\right>$', 'Interpreter', 'latex', ...
    'FontSize', 20, 'HorizontalAlignment', 'Center')

% ------------------------------------------------------
```

Listing A.2: Insert code directly in your document

```
% Visualize Bloch Vector

X = [0 1; 1 0];
Z = [1 0; 0 -1];
Y = 1i * X * Z;
```

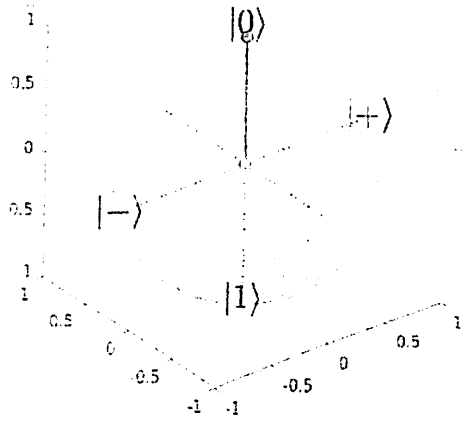


Figure A.1: $|0\rangle$

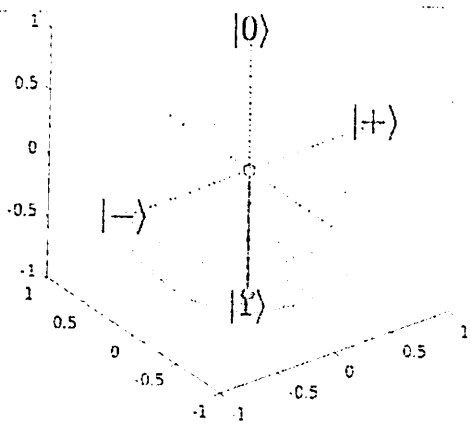


Figure A.2: $|1\rangle$

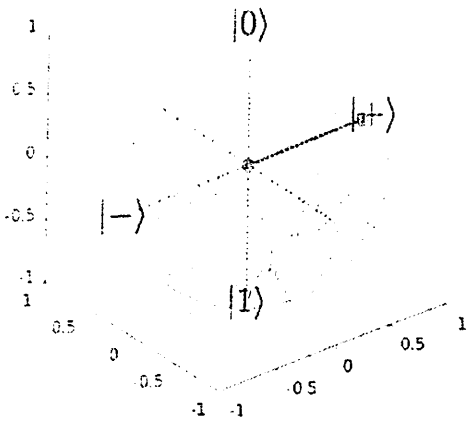


Figure A.3: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

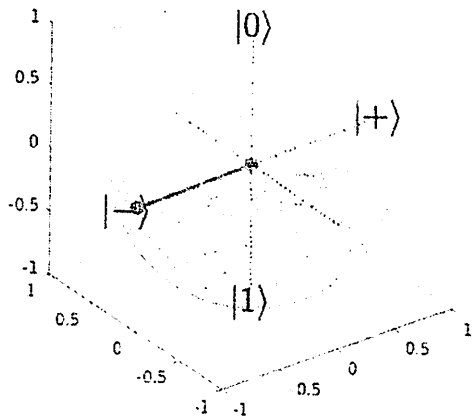


Figure A.4: $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

```
H = (1/sqrt(2)) * (X + Z);

E0_plus = bin2vec('0');
E1_plus = bin2vec('1');

E0_X = (E0_plus + E1_plus)/sqrt(2);
E1_X = (E0_plus - E1_plus)/sqrt(2);

plotBlochSphere

rho = ket2dm(ket0);
lambda0 = ket2bv(E0_plus);
lambda1 = ket2bv(E1_plus);
lambdaX0 = ket2bv(E0_X);
lambdaX1 = ket2bv(E1_X);

plotBlochVect(E0_plus,[1 0 0]);
plotBlochVect(E1_plus,[1 0 0]);
plotBlochVect(E0_X,[0 0 1]);
plotBlochVect(E1_X,[0 0 1]);

function rho = ket2dm(ket)
% Convert ket tp a density matrix rho

rho = ket * ket';
end

function lambda = ket2bv(ket)
    rho = ket2dm(ket);
    X = [0 1; 1 0];
    Z = [1 0; 0 -1];
    Y = 1i * X * Z;

    lambda = [trace(X*rho); trace(Y*rho); trace(Z*rho)];
end

function plotBlochVect(ket,color)
    lambda = ket2bv(ket);

    someBV = line([0 lambda(1)], [0 lambda(2)], [0 lambda(3)], ...
        'LineWidth', 2, 'Marker','0','Color',color)
end
```

B. Appendix B

Listing B.1: Insert code directly in your document

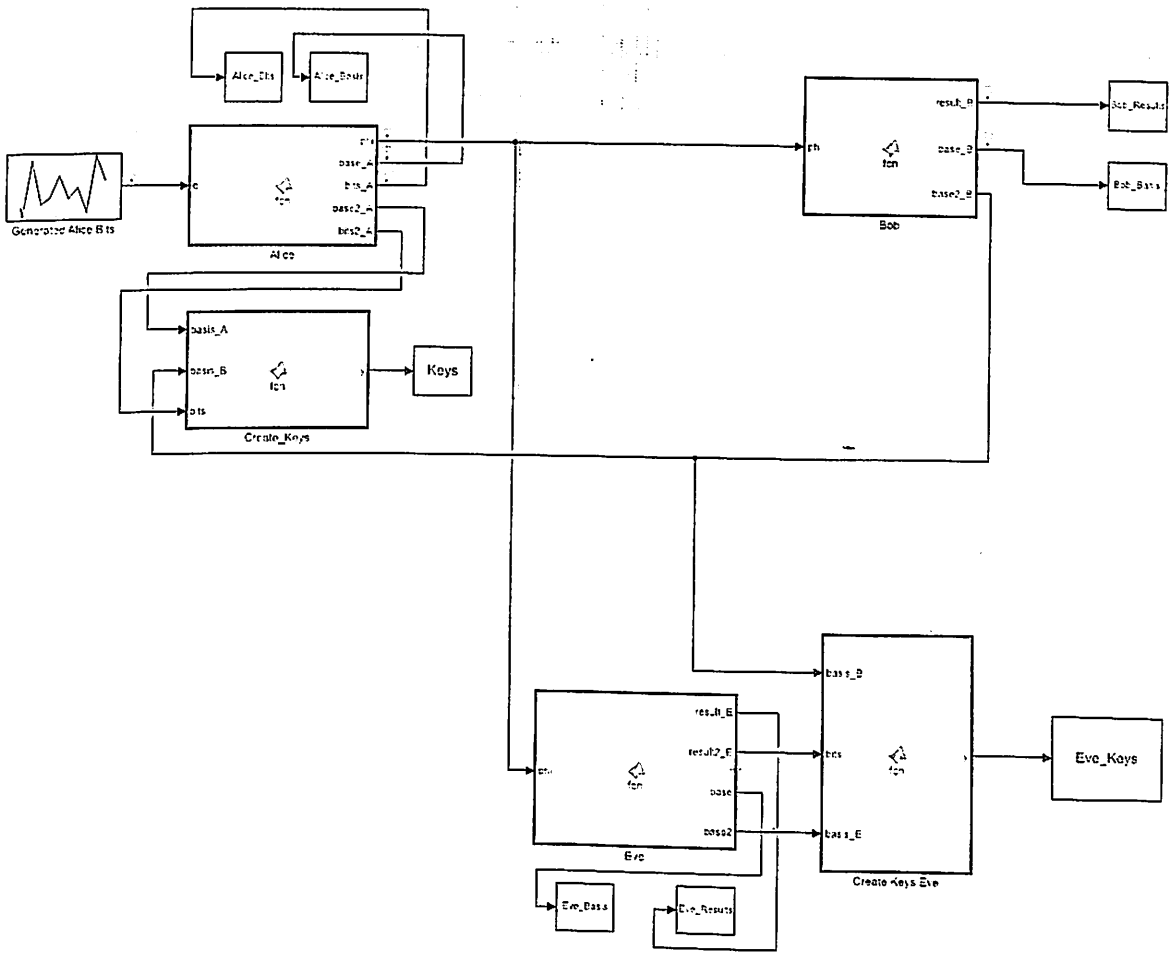
```
function [a,b,e] = bb84(n)

fprintf('\n\n===␣BB84␣protocol␣===\n\n');

list_qubits = cell(1,n);

H = hadamard(1);

%  ---------- Alice generates bits and basis -------
bits = rand(1,n) > 0.5;
basis_A = rand(1,n) > 0.5;

result_A = 1:n;

for a=1:n
    if bits(a)==0
        result_A(a)=0;
    else
        result_A(a)=1;
    end
end

E0_plus = bin2vec('0');
E1_plus = bin2vec('1');

E0_X = (E0_plus + E1_plus)/sqrt(2);
E1_X = (E0_plus - E1_plus)/sqrt(2);

% ------------- Alice encode qubits -----------------

for k = 1:n
    if basis_A(k)==0
        if bits(k)==0
```

```
            phi = E0_plus;
            list_qubits{k} = (H*phi);
        else
            phi = E1_plus;
            list_qubits{k} = (H*phi);
        end
    else
        if bits(k)==0
            phi = E0_X;
            list_qubits{k} = (H*phi);
        else
            phi = E1_X;
            list_qubits{k} = (H*phi);
        end
    end
end

% --------- Alice sent list qubits to Bob --------------

% --------- Eve eavesdrop qubits -----------------------

n_E = length(list_qubits);
basis_E = rand(1,n_E) > 0.5;

result_E = 1:n_E;

for i = 1:n_E
    if basis_E(i) == 0
        ppp = pretty(measure(H*list_qubits{i}));

        if ppp == pretty(measure(bin2vec('1')))
            result_E(i) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            result_E(i) = 0;
        else
            disp('Else1␣:');
            disp(pretty(measure(list_qubits{i})));
        end
    elseif basis_E(i) == 1
        ppp = pretty(measure(list_qubits{i}));
        if ppp == pretty(measure(bin2vec('1')))
            result_E(i) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            result_E(i) = 0;
        else
```

```
            disp('Else2 :');
            disp(pretty(measure(list_qubits{i})));
        end
    end
end

% ------------ Eve encode and sent list qubits ------------

list_qubits_E = cell(1,n_E);

for e = 1:n_E
    if basis_E(e)==0
        if bits(e)==0
            phi = E0_plus;
            list_qubits_E{e} = (H*phi);
        else
            phi = E1_plus;
            list_qubits_E{e} = (H*phi);
        end
    else
        if bits(e)==0
            phi = E0_X;
            list_qubits_E{e} = (H*phi);
        else
            phi = E1_X;
            list_qubits_E{e} = (H*phi);
        end
    end
end


% --------- Bob received list qubits ----------

% ----------- Bob generated basis ------------

n_B = length(list_qubits);

basis_B = rand(1,n_B) > 0.5;

result_B = 1:n_B;

for i = 1:n_B
    if basis_B(i) == 0
        ppp = pretty(measure(H*list_qubits{i}));
```

```
        if ppp == pretty(measure(bin2vec('1')))
            result_B(i) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            result_B(i) = 0;
        else
            disp('Else1 :');
            disp(pretty(measure(list_qubits{i})));
        end
    elseif basis_B(i) == 1
        ppp = pretty(measure(list_qubits{i}));
        if ppp == pretty(measure(bin2vec('1')))
            result_B(i) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            result_B(i) = 0;
        else
            disp('Else2 :');
            disp(pretty(measure(list_qubits{i})));
        end
    end
end

% ---------- Create keys --------------

keys = [];
count = 1;

for c=1:n
    if basis_A(c)==basis_B(c)
        keys(count) = result_B(c);
        count = count + 1;
    end
end

keys_E = [];
count_E = 1;
for c=1:n
    if basis_A(c)==basis_E(c)
        keys_E(count_E) = result_B(c);
        count_E = count_E + 1;
    end
end

a = result_A;
b = result_B;
e = result_E;
```

```
end
```


Alice_Bits
Alice_Basis
Generated Alice Bits
Alice
Bob
Bob_Results
Bob_Basis
basis_A
basis_B
bits
Create_Keys
Keys
Eve
Eve_Bass
Eve_Results
Create Keys Eve
Eve_Keys

C. Appendix C

Listing C.1: Insert code directly in your document

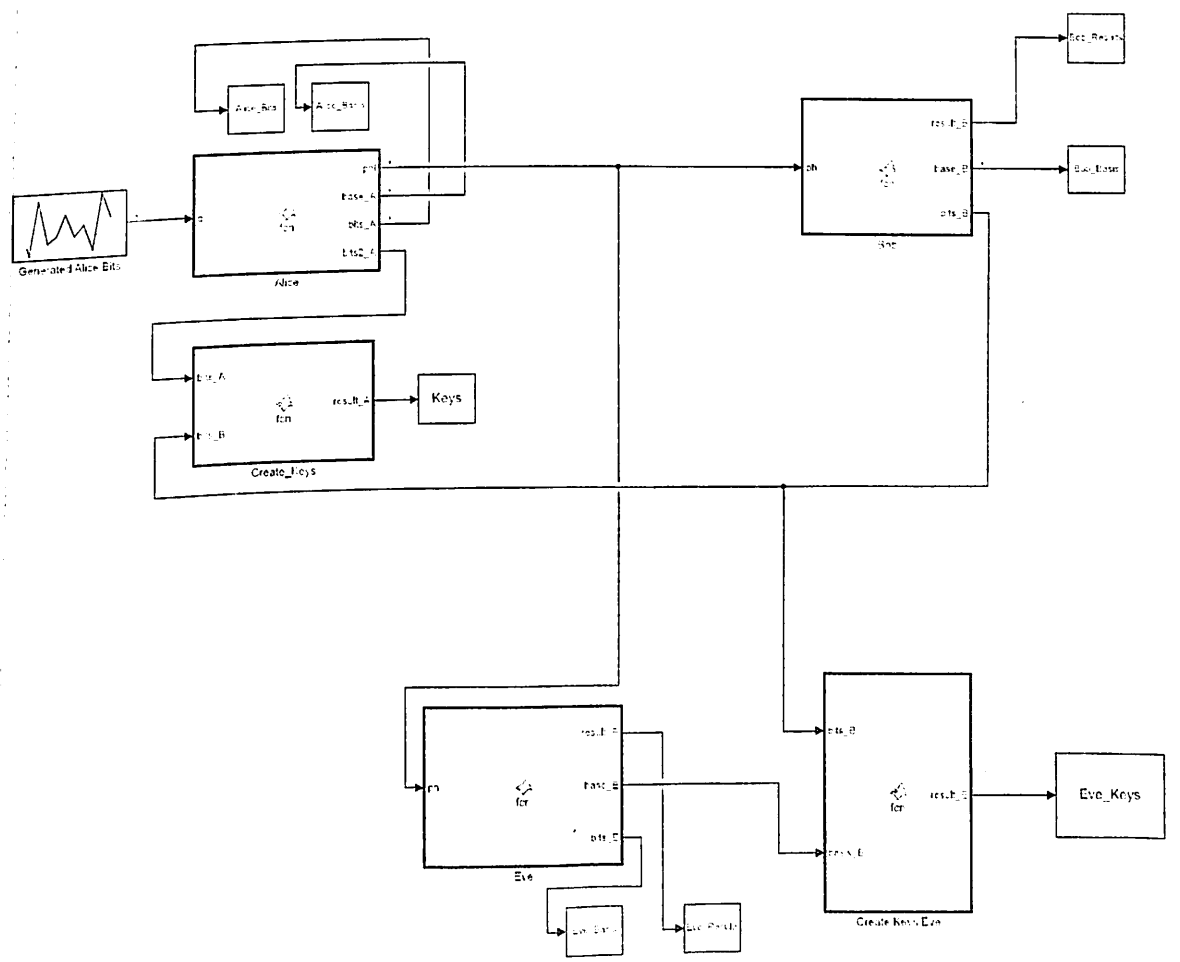
```
% ------------------------------------------------------
% --- Alice generated bits and sent list qubits ------
% ------------------------------------------------------

function y = bb92(n)


H = hadamard(1);
bits_A = rand(1,n) > 0.5;

list_qubits = cell(1,n);

E0_plus = bin2vec('0');
E1_plus = bin2vec('1');
E0_X = (E0_plus + E1_plus)/sqrt(2);

for i=1:n
    if bits_A(i) == 0
        list_qubits{i}= (H*E0_plus);
    elseif bits_A(i) == 1
        list_qubits{i} = (H*E0_X);
    end
end

% ------------------------------------------------------
% -------- Eve received list qubits -------------------
% ------------------------------------------------------


n_E = length(list_qubits);

basis_E = rand(1,n_E) > 0.5;
```

```
bits_E = [];

for e = 1:n_E
    if basis_E(e) == 0
        ppp = pretty(measure(H*list_qubits{e}));
        %disp(ppp);

        if ppp == pretty(measure(bin2vec('1')))
            bits_E(e) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            bits_E(e) = 0;
        else
            disp('Else1 :');
            disp(pretty(measure(list_qubits{e})));
        end
    elseif basis_E(e) == 1
        ppp = pretty(measure(list_qubits{e}));
        %disp(ppp)

        if ppp == pretty(measure(bin2vec('1')))
            bits_E(e) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            bits_E(e) = 0;
        else
            disp('Else1 :');
            disp(pretty(measure(list_qubits{e})));
        end
    end
end

result_E = [];
ccc_E = 1;

for e = 1:length(bits_E)
    if bits_E(e) == 1
        result_E(ccc_E) = basis_E(e);
        ccc_E = ccc_E + 1;
    end
end

% ------------------------------------------------------
% ---------- Bob received list qubits ------------------
% ------------------------------------------------------
```

```
n_B = length(list_qubits);

basis_B = rand(1,n_B) > 0.5;

bits_B = [];

for b = 1:n_B
    if basis_B(b) == 0
        ppp = pretty(measure(H*list_qubits{b}));
        %disp(ppp);

        if ppp == pretty(measure(bin2vec('1')))
            bits_B(b) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            bits_B(b) = 0;
        else
            disp('Else1 :');
            disp(pretty(measure(list_qubits{b})));
        end
    elseif basis_B(b) == 1
        ppp = pretty(measure(list_qubits{b}));
        %disp(ppp)

        if ppp == pretty(measure(bin2vec('1')))
            bits_B(b) = 1;
        elseif ppp == pretty(measure(bin2vec('0')))
            bits_B(b) = 0;
        else
            disp('Else1 :');
            disp(pretty(measure(list_qubits{b})));
        end
    end
end

result_B = [];
ccc_B = 1;

for b = 1:length(bits_B)
    if bits_B(b) == 1
        result_B(ccc_B) = basis_B(b);
        ccc_B = ccc_B + 1;
    end
end

% ------------------------------------------------
```

```
% -------- Bob sent bits to Alice -----------------
% -------------------------------------------------

result_A = [];
ccc = 1;

for a = 1:length(bits_A)
    if bits_B(a) == 1
        result_A(ccc) = bits_A(a);
        ccc = ccc + 1;
    end
end

% -------------------------------------------------
% ------------ Results --------------------------
% -------------------------------------------------

% ----------- Alice --------------------

rev_result_A = [];
for a = 1:length(result_A)
    if result_A(a) == 1
        rev_result_A(a) = 0;
    elseif result_A(a) == 0
        rev_result_A(a) = 1;
    end
end

%disp('Binary to Hexadecimal: ');
disp(binaryVectorToHex(result_A));
disp(binaryVectorToHex(rev_result_A));
disp('-----------------------------------------------------------------');

% ------------------------------ Bob ---------------------------------

rev_result_B = [];
for b = 1:length(result_B)
    if result_B(b) == 1
        rev_result_B(b) = 0;
    elseif result_B(b) == 0
        rev_result_B(b) = 1;
    end
end

%disp('Binary to Hexadecimal: ');
```

```
disp(binaryVectorToHex(result_B));
disp(binaryVectorToHex(rev_result_B));
% ------------ Eve --------------------

rev_result_E = [];
for e = 1:length(result_E)
    if result_E(e) == 1
        rev_result_E(e) = 0;
    elseif result_E(e) == 0
        rev_result_E(e) = 1;
    end
end

%disp('Binary to Hexadecimal: ');
disp(binaryVectorToHex(result_E));
disp(binaryVectorToHex(rev_result_E));
```



D. Appendix D

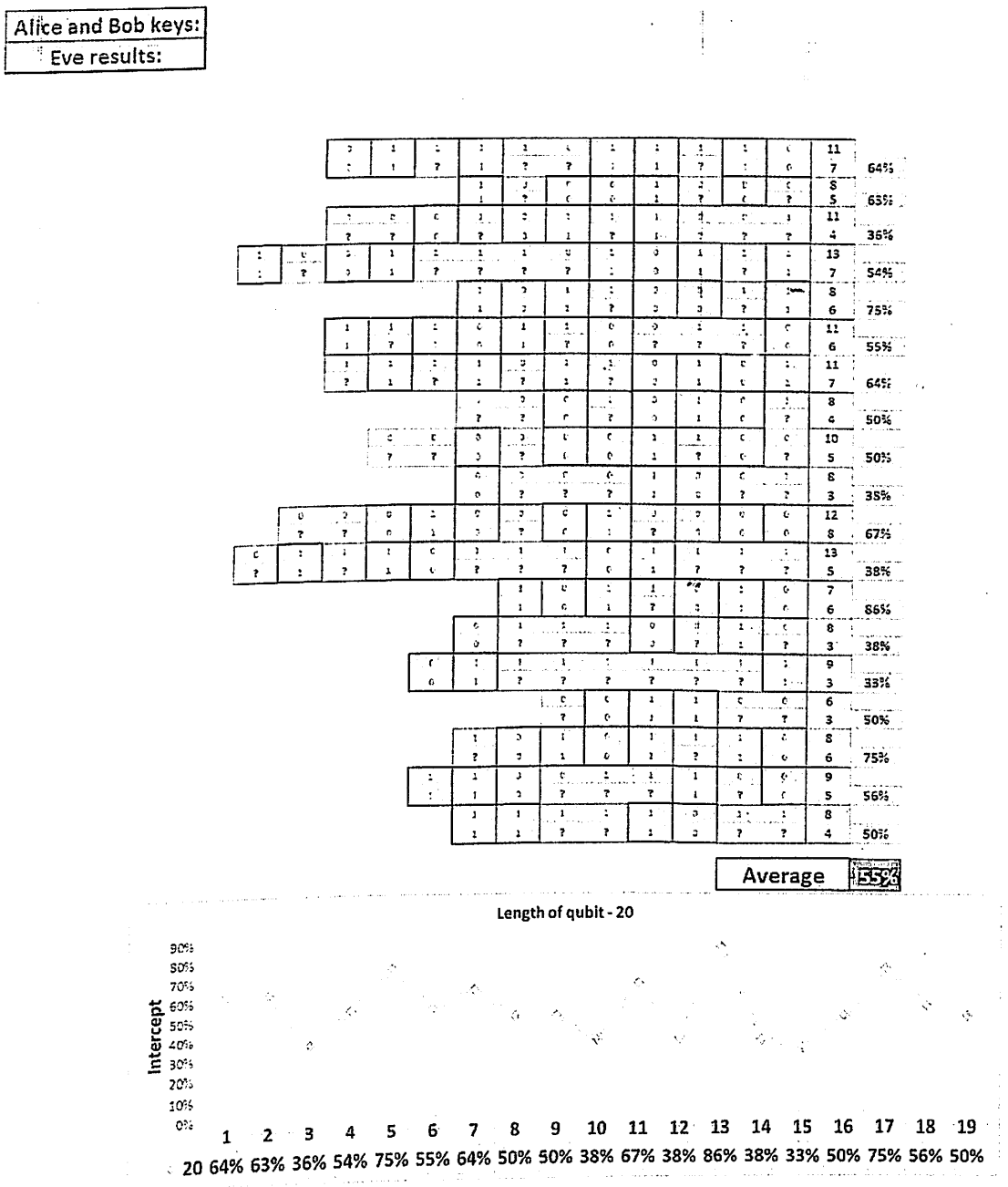


Figure D.1: Case 1: Length of qubits 20

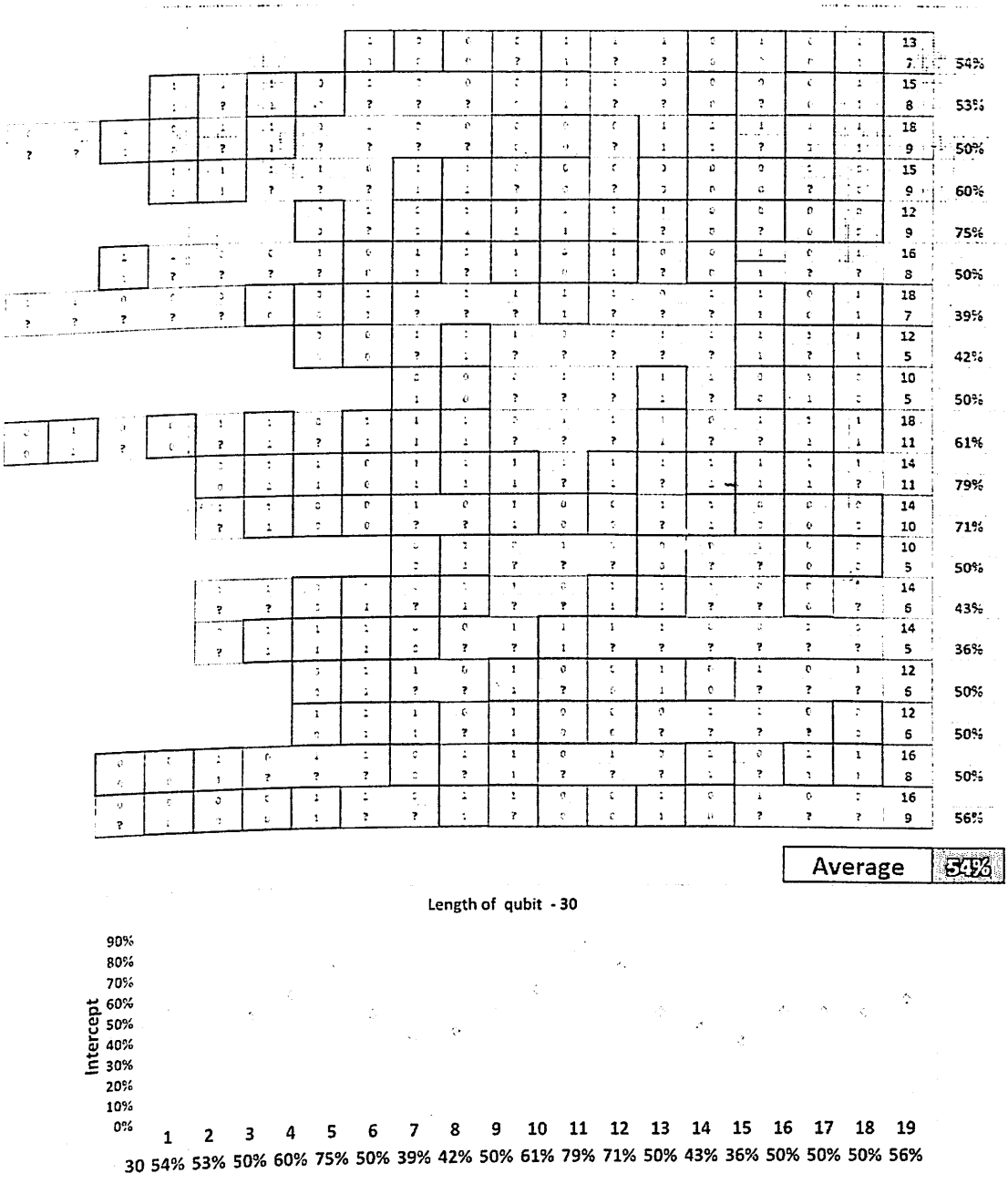


Figure D.2: Case 1: Length of qubits 30

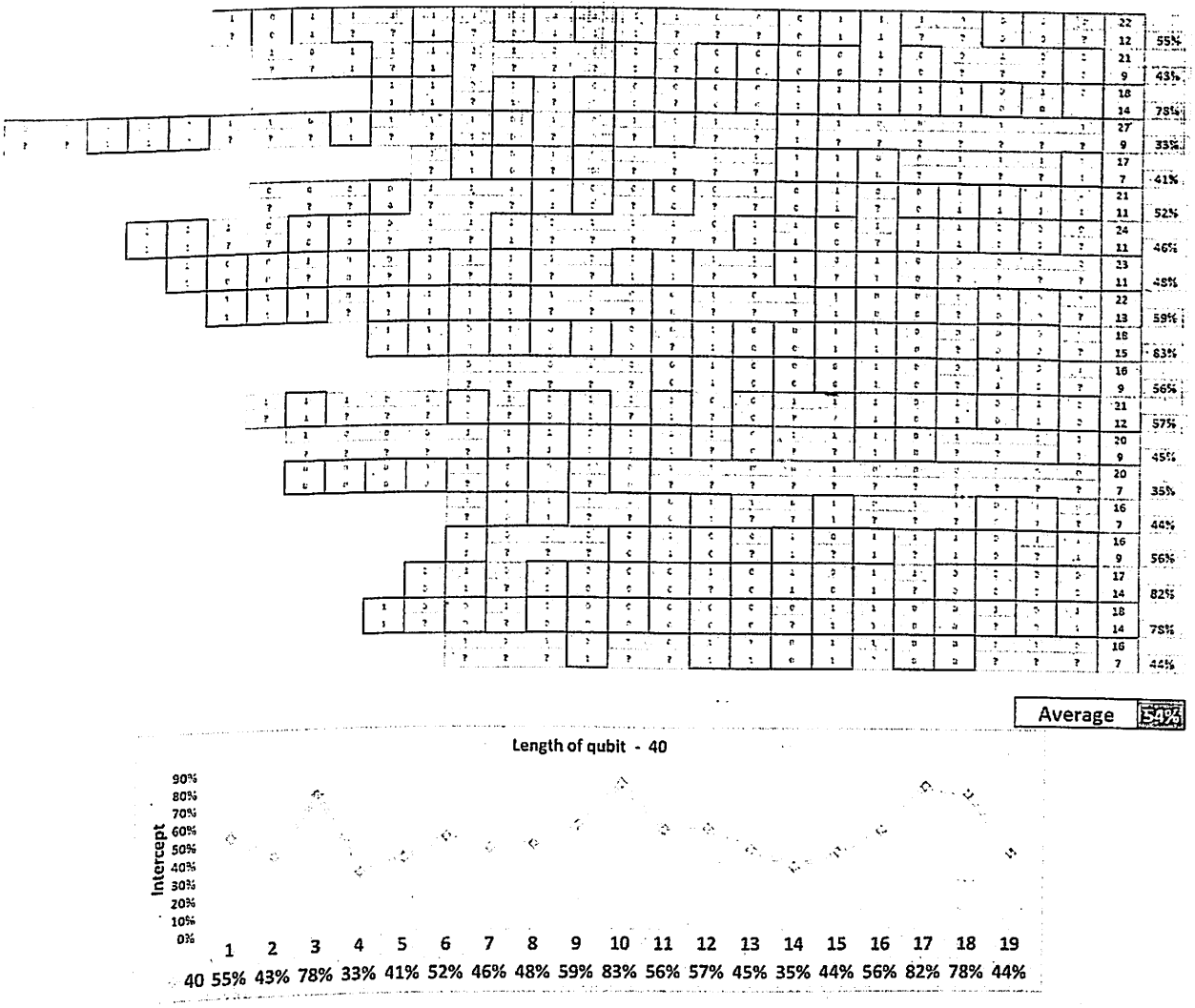


Figure D.3: Case 1: Length of qubits 40

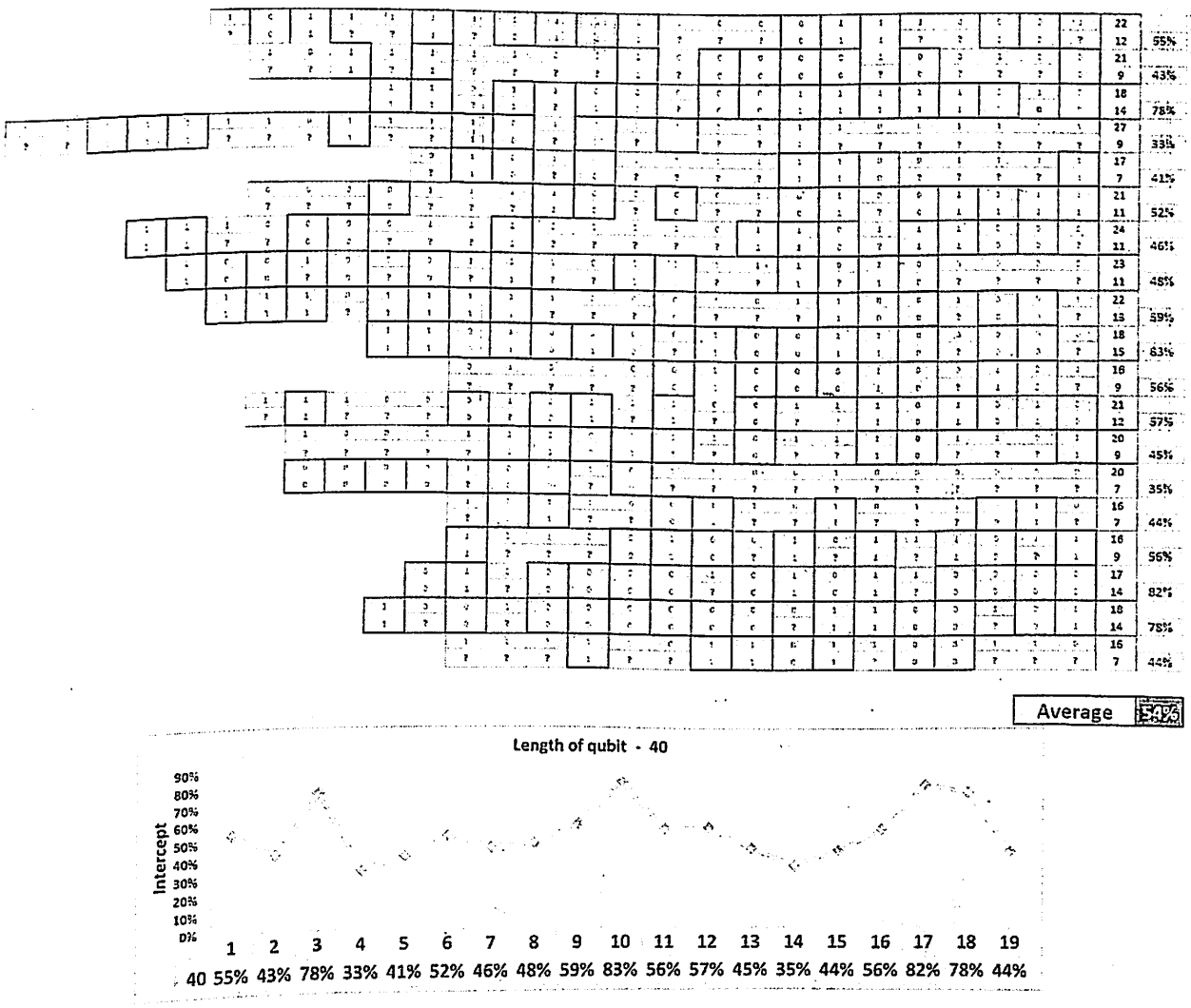


Figure D.3: Case 1: Length of qubits 40

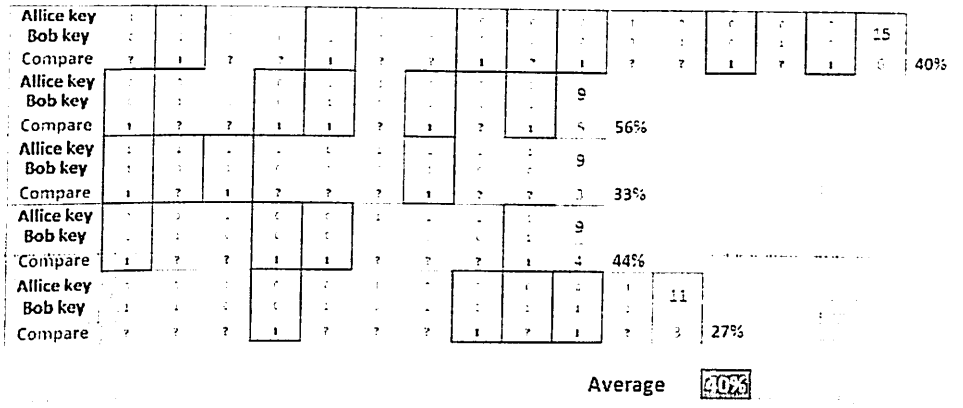


Figure D.4: Case 2: Length of qubits 20

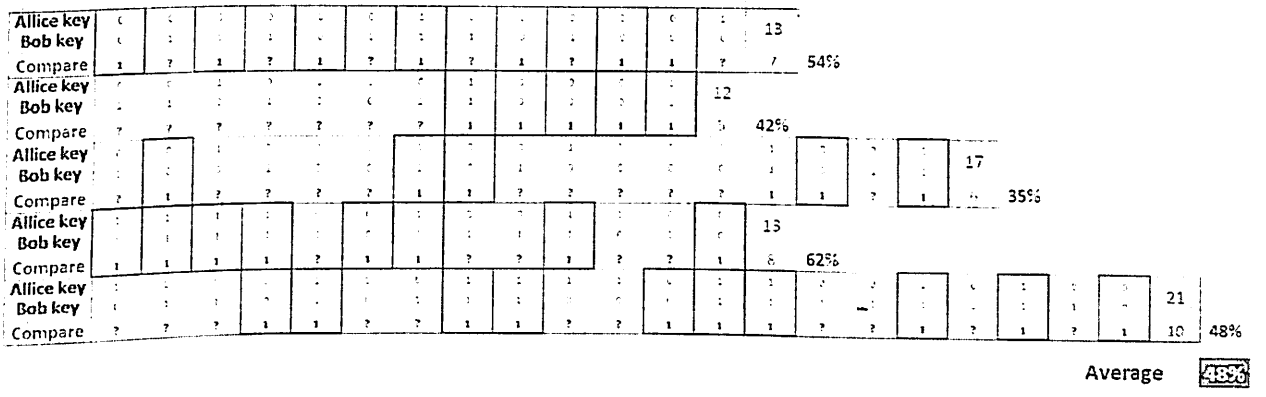


Figure D.5: Case 2: Length of qubits 30

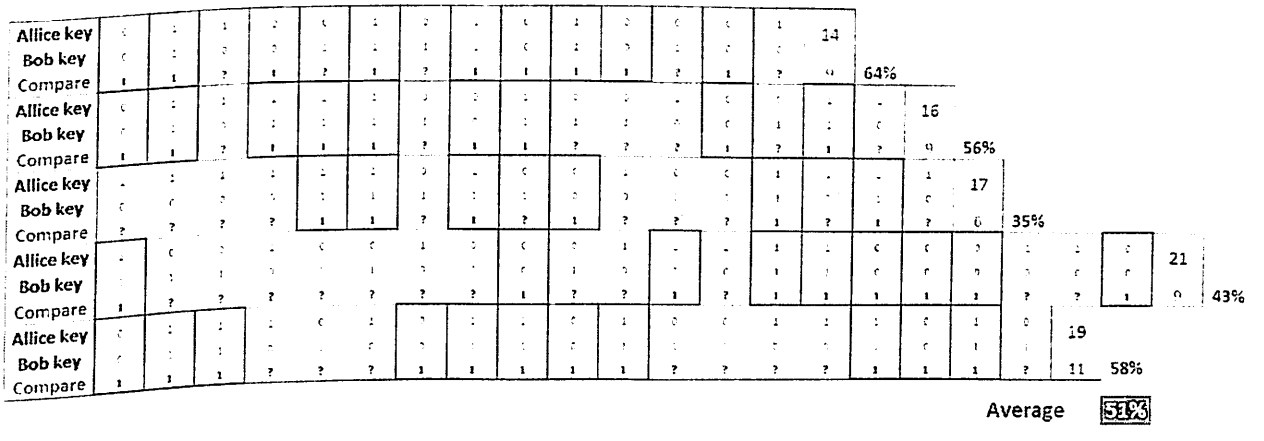


Figure D.6: Case 2: Length of qubits 40

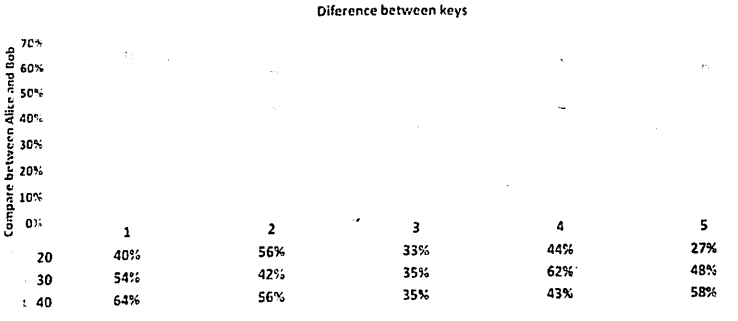


Figure D.7: Graphics of Case 2