ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

QUANTUM KEY DISTRIBUTION PROTOCOL-BASED IMAGE ENCRYPTION ALGORITHM

M.Sc. Thesis by

Omar Salhab

Department of Computer Engineering

July 2022

ANKARA

QUANTUM KEY DISTRIBUATION PROTOCOL-BASED IMAGE ENCRYPTION ALGORITHM

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Department of Computer Engineering

by

Omar Salhab

July, 2022

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"QUANTUM KEY DISTRIBUTION PROTOCOL-BASED IMAGE ENCRYPTION ALGORITHM"** completed by **OMAR SALHAB** under the supervision of **PROF. DR. FATIH V. ÇELEBİ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fatih V. ÇELEBİ

Supervisor

Prof. Dr. Remzi YILDIRIM

Jury Member

Doç. Dr. Hasan Hüseyin SAYAN

Jury Member

Prof. Dr. Sadettin ORHAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 19/07/2022 **Signature: ……………………………………….**

Name & Surname: Omar Salhab

ACKNOWLEDGMENTS

Firstly, I would like to convey my sincere appreciation and gratitude to my supervisor, Prof. Dr. Fatih V. ÇELEBİ for his continuous support and care during my study. With his help I was able to finish writing this thesis. Secondly, I would like to thank my family who continued to offer their help and support remotely from Palestine.

2022, 19 July **Omar Salhab**

QUANTUM KEY DISTRIBUTION PROTOCOL-BASED IMAGE ENCRYPTION ALGORITHM

ABSTRACT

In this thesis, a simulation of BB84 QKD protocol will be implemented to generate a secret key which is used for the encryption and decryption of digital images. Moreover, a method for combining the original image with the generated secret key will be proposed. In the proposed method, both the original image and the key are split into their RGB components, and a combination of XOR operations and modular arithmetic is applied to combine them. The QKD-based encryption method is tested on various standard test images of different dimensions. To ensure the robustness, efficiency, and performance of this QKD-based encryption scheme, several tests and analysis will be run on the resulting data including standard attack checks, correlation coefficient analysis, entropy analysis, histogram analysis. Finally, the results will be compared with the E91 QKD protocol. As the future of the secure-deemed public-key cryptography is in jeopardy, Quantum Key Distribution (QKD), which is part of Quantum Cryptography (QD), addresses these security concerns by providing a robust and secure solution. QKD utilizes the principles of quantum mechanics and the laws of physics to establish a secure secret key between two parties with the help of Heisenberg uncertainty principle and the no-cloning theorem. The key security feature of QKD protocols is the ability to detect whether an eavesdropper is present since an attempt to eavesdrop would cause a disturbance in the system and alert the original parties. There has been remarkable development regarding the design of QKD protocols recently, and many in-field prototypes are being implemented.

Keywords: Quantum key distribution, image encryption, quantum cryptography.

KUANTUM ANAHTAR DAĞITIM PROTOKOLÜ TABANLI GÖRÜNTÜ ŞİFRELEME ALGORİTMASI

ÖZET

Bu tezde, öncelikle dijital görüntülerin şifrelenmesi için BB84 QKD protokolü kullanılarak yapılmıştır. Sonrada, orijinal görüntünün oluşturulan gizli anahtarla birleştirilmesi için bir yöntem önerilecektir. Önerilen yöntemde hem orijinal görüntü hem de anahtar RGB bileşenlerine bölünür ve bunları birleştirmek için XOR işlemleri ve modüler aritmetik kombinasyonu uygulanmaktadır. QKD tabanlı şifreleme yöntemi, farklı boyutlardaki çeşitli standart test görüntüleri üzerinde test edilmektedir. Sonuçlar E91 ile karşılaştırılır. Güvenli kabul edilen açık anahtar şifrelemenin geleceği tehlikede olduğundan, Kuantum Şifrelemenin (QD) bir parçası olan Kuantum Anahtar Dağıtımı (QKD), sağlam ve güvenli bir çözüm sağlayarak bu güvenlik endişelerini gidermektedir. QKD, Heisenberg belirsizlik ilkesi ve klonlama teoremi yardımıyla iki taraf arasında güvenli gizli anahtar oluşturmak için kuantum mekaniği ilkelerini ve fizik yasalarını kullanmaktadır. QKD protokollerinin temel güvenlik özelliği, bir gizli dinleme girişiminin sistemde bozulmaya neden olacağı ve orijinal tarafları uyaracağı için bir dinleyicinin mevcut olup olmadığını tespit etme yeteneğidir. Son zamanlarda QKD protokollerinin tasarımı ile ilgili dikkate değer bir gelişme oldu ve birçok saha içi prototip uygulanmaktadır. Bu QKD tabanlı şifreleme şemasının sağlamlığını, verimliliğini ve performansını sağlamak için, standart saldırı testleri, korelasyon katsayısı analizi, entropi analizi ve histogram analizi dahil olmak üzere elde edilen veriler üzerinde çeşitli testler ve analizler gerçekleştirilecektir.

Anahtar Kelimeler: Kuantum anahtar dağıtımı, görüntü şifreleme, kuantum şifreleme.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Cryptography is the science in which a secure communication is established and facilitated between the sender and the intended recipient that allows them to view and understand the contents of the transmitted message. Other unauthorized people present in the medium will not be able to understand the contents of the message. The concept of cryptography and encryption using a private key emerged alongside the idea of writing. As the human civilization evolved, people got divided into diverse groups and tribes which helped the development of the notion of power, politics, and wars. The need to send a secret message in which a third party could not read it was paramount. The success of transmitting such secret messages often led to either life or death, especially in wars. Moreover, it was one of the leading factors to the rise and fall of kingdoms and empires throughout history [1,2].

Today, cryptography has many applications that we use in our everyday lives. Examples include doing online shopping from the comfort of our homes during this pandemic period, sending and receiving instant messages on social media networks that utilize end-to-end encryption technologies, and making cryptocurrencies that are rising in popularity possible [3]. The uses of cryptography are not limited to hiding text messages, but they include other types of media like images and audio. A popular phrase says “A picture is worth a thousand words” due to how much information and ideas a picture can convey. Since digital images have applications in many fields like medical and military, it is important to be able to hide or encrypt information on a digital image from unauthorized access.

When it comes to cryptography, there are two main processes in which cryptography revolves around: encryption and decryption. Encryption is the process of converting the original text or media, also known as plaintext, into an unreadable format called ciphertext. This is achieved at the sender’s end, and it involves combining the plaintext with a secret key to produce the ciphertext. Decryption, which occurs at the receiver end, is the reverse of encryption, and it uses the key to decode the ciphertext and convert it to its original readable form. Perhaps one of the earliest and most simple examples of an encryption method is the *Caesar cipher* [4], which was named after the famous Roman general, Julius Caesar. At that time, secret messages were needed to be sent for military purposes. The idea behind this cipher is simple; each letter in the plaintext is substituted by another letter in the alphabet that follows the original letter by a fixed amount. In the case of the original Caesar cipher, the ciphertext is obtained by replacing each letter from the plaintext with a letter that follows the original letter by three positions [5]. Thus, each letter in the alphabet is shifted by three, as shown below:

Plain alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Cipher alphabet	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

To demonstrate, suppose we want to encrypt and send the message “meet at night”. By using Caesar cipher, this plaintext will be encrypted, and the resulting ciphertext is “PHHW DW QLJKW”. Decrypting the message is as easy and straightforward as the encryption process. Each letter is replaced by a letter that precedes it by three positions. However, the ciphertext should only be decrypted and understood by those who possess the key to the cipher. Does this mean that this encryption scheme is impenetrable, uncrackable, and secure against attacks? Absolutely not. Since there are only twenty-six letters in the English alphabet, the number of possibilities for what each letter is going to be replaced by is only twenty-five. Using brute force and trial-and-error methods, the key can be easily obtained (assuming we already know that a shift cipher is being used) [6]. Another method that can be used to crack the cipher is by making use of the frequency in which each letter is used in the English language [7]. For example, the letter “E” is the most used letter in English with a frequency of 11.6%. By analyzing the ciphertext, we can note which letters are mostly used, and replace them with the corresponding high-frequency letters from the English alphabet. Obviously, the bigger the ciphertext is, the better the results we obtain from trying to crack the cipher. As we can see, there are numerous methods one can use to decrypt the secret message and acquire the original text.

Other more sophisticated and complex ciphers were designed subsequently to be more secure than Caesar cipher. However, even with their enhanced implementations which involved randomly assigning a letter from the alphabet to be replaced with the original letter, as opposed to simply shifting the alphabet by a fixed number of positions, they were still vulnerable to certain attacks. While they might be secure against brute force attacks because of the vast number of possibilities, they are still vulnerable to the same frequency analysis which was discussed earlier. It was only until the Vigenère cipher was designed that people thought that a secure cipher was finally discovered. Vigenère cipher is a polyalphabetic cipher that used several Caesar ciphers which makes it much harder to crack. Even with its relatively advanced design at the time, it was finally cracked after three centuries [8]. One thing can be understood, and it's that there exists a big desire to come up with a cryptography scheme that is totally secure against attacks.

Cryptography techniques can be classified into two paradigms: classical cryptography, and quantum cryptography. They differ in what protocols, concepts, and theories are used. Classical cryptography schemes are used in many areas in our everyday lives as it is considered secure against attacks. Even the most powerful computer would need trillion of years to crack some of the classical cryptography algorithms [9,10]. However, the rise of quantum computers would render these algorithms useless as they have the power to crack algorithms in a matter of minutes [11]. Therefore, it is best to deal with quantum computers using actual quantum cryptography. In the following chapters, classical cryptography and quantum cryptography will be discussed.

1.2 The Aim of Study

This study has been conducted for two objectives. Firstly, the need to switch from classical cryptography to quantum cryptography will be emphasized by examining the weaknesses of classical cryptography algorithms that were introduced in the new era of quantum computers. Secondly, an application of a Quantum Key Distribution (QKD) protocol will be implemented to achieve secure image cryptography. The proposed method will use the key generated by the simulated BB84 QKD protocol and combine it with the digital image to perform image encryption and decryption. The performance will be evaluated by running tests that determine the efficiency and security of the proposed method.

1.3 Structure of The Study

The study will be structured as follow:

- **Chapter 1:** This chapter serves as an introduction to my study work. A general overview of how cryptography works is provided. The aims and objectives of the study are mentioned.
- **Chapter 2:** The second chapter will review some of the state-of-the-art research that was conducted in the field. Research that highlighted developments in the fields of both QKD and image encryption is discussed.
- **Chapter 3:** This chapter will go over the classical encryption methods from private-key encryption to public-key encryption. The security of these methods will be discussed in the era of classical computers.
- **Chapter 4:** This chapter will introduce quantum computers and show how quantum computers defeat classical encryption methods and can render them useless. The approaches that are being taken to address the weakness introduced in the era of quantum computers will be discussed. The two main approaches include post-quantum cryptography and quantum cryptography.
- **Chapter 5:** This chapter handles the implementation details of the simulated BB84 QKD protocol and shows the design of the encryption scheme. The results and the analysis are also discussed.
- **Chapter 6:** The final chapter serves as a conclusion of the study where it summarizes the study and shows the conclusions that were drawn.

CHAPTER 2

LITERATURE REVIEW

E Diamanti et al. presented their research which highlighted the challenges that quantum cryptography and QKD are facing. These challenges include the cost, key rate, distance, and practical security. Furthermore, some approaches to overcome these challenges were proposed [12].

H Pawar et al. in their research presented a theoretical comparison between classical and quantum cryptography for image encryption and decryption. They gave an overview of each method and explained the theory behind it. The strengths and weaknesses of both classical and quantum cryptography were highlighted. Finally, they came to the conclusion that quantum cryptography is indeed more secure than its classical counterpart when transferring multimedia data such as digital images. However, these are all theoretical comparisons as an implementation of a QKD protocol hasn't been made, and therefore, the study lacked practical tests to support the conclusions [13].

Q Zhang et al. conducted a study that covers two of the main challenges when attempting to transition from theoretical QKD to practical QKD implementation. They also discussed the efforts that were made to overcome and the implementation obstacles. One of the challenges is having ideal devices for that act as photon sources and detectors. In the real world, no ideal devices exist, and device imperfection is an inevitable problem. As a result, the security of QKD will suffer as attackers can exploit the device imperfections to their advantage. The proposed solution is to come up with QKD protocols that don't get affected by device imperfections. Two protocols that overcome this problem are decoy-state QKD and measurement-device-independent (MDI QKD). The other major challenge to globalize QKD protocols and deploying them on large-scale networks. Currently, QKD are working on small to medium-sized networks like local and metropolitan networks. It is a challenge to switch to large-scale quantum networks since this will cause the performance to suffer. New protocols are being proposed that extends the distance in which quantum networks operate [14].

S Li et al. proposed a new QKD scheme that is based on one decoy-state QKD protocols. The proposed method was proven to have a better key rate than its predecessors as well as an enhanced transmission distance. L Hu et al. proposed a method to improve parameters of cascade in the post-processing stage, which leads to better error detection and correction. However, the performance enhancement in these schemes comes with the cost of higher complexity and computation costs [15].

A Nurhadi et al. conducted a survey paper on QKD protocols. The paper reviewed nine of the key QKD protocols in the field. These protocols can be categorized under two main types: prepare-and-measure-based protocols and entanglement-based protocols. In addition to the survey, a simulation of three protocols was made, two of which are prepare-and-measure-based protocol while the third is an entanglement-based protocol. The experiment resulted in three parameters: number of keys (Nk), number of errors (Ne), and the ratio of the number of errors to the number of keys (Np) [16].

CHAPTER 3

CLASSICAL CRYPTOGRAPHY

Classical cryptography centers around the design of complex and sophisticated mathematical equations and the complications of factorizing enormous numbers into primes. It relies on the complexity of these mathematical equations to ensure that the transferred data stays safe from unauthorized access [17]. The harder the equations are to crack, the more secure the encryption scheme is. In classical cryptography, the plaintext is encrypted at the sender's end by means of a secret key that only the sender and receiver know of. The ciphertext is consequently sent to the receiver via an unsecure communications channel. At the receiver's end the key is reused to decode the ciphertext and convert it back to its original form. It is worth noting that an unsecure communications channel is typically used because the ciphertext that is being sent is unreadable for any party that does not have the key. Without the key, the sent ciphertext is just scrambles that is unreadable and does not make sense.

Classical cryptography can be categorized into three types: symmetrical cryptography and asymmetrical cryptography, and hashing.

3.1 Symmetric Cryptography

In symmetrical cryptography, also known as private-key cryptography, the same key that is used at the sender's end to encrypt the message is also used at the receiver's end to decrypt the sent message. The sender and the intended recipient should both agree on a secret key that is shared between them and will be used throughout the entirety of the communication process. If an unauthorized third party was able to intercept the key and figure out what it is, he can simply use it to view the sent messages and even modify them [18]. Thus, if any of the sender or receiver discover that the key has been compromised, the old key must be discarded, and a new key must be created immediately. Figure 3.1 shows an illustration for private-key encryption.

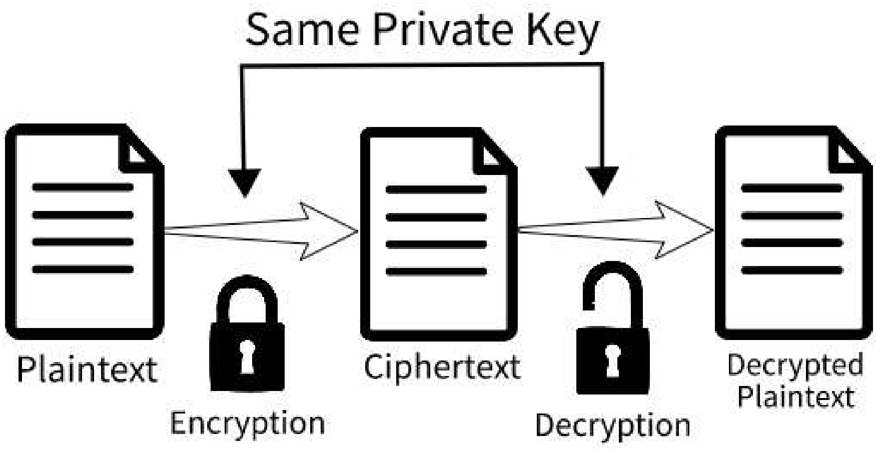


Figure 3.1 Private-key encryption

The advantage of private-key cryptography is that it is a simple type of cryptography, and it does not require much overhead. The computational simplicity of this method enables faster encryption and decryption speeds that could come in handy when needing to decode a huge amount of data. The drawbacks of this encryption scheme are key management and distribution. It is quite difficult to exchange keys over an unsecure network without a third-party gaining access to it [19].

3.2 Asymmetric Cryptography

Instead of there being only one secret key that is used for both encryption and decryption, asymmetric cryptography utilizes a key pair to achieve data secrecy: public key and private key. At the sender's end the public key can be used to encrypt the original data, while the corresponding private key is used to decrypt the data at the receiver's end [20]. Similarly, if the two parties wish to exchange messages back and forth, then two pairs of keys would be required. Each party should have the public key of the other party in their possession to be able to encrypt and send messages to them. This method overcame the security issues of the previous symmetric-key encryption because the private key will stay safe with the intended parties without the need of it being transmitted through an unsecure network [21]. Figure 3.2 shows an illustration for public-key encryption.

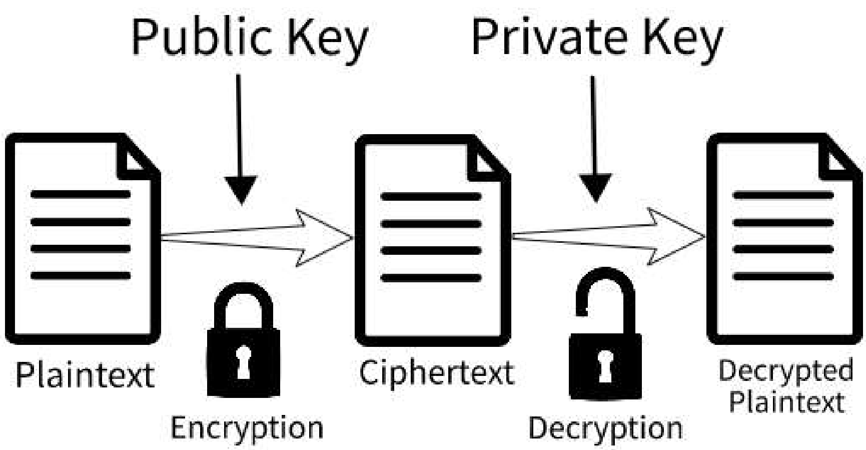


Figure 3.2 Public-key encryption

It is worth noting that both the private key and the public key are independent of each other. In other words, you cannot derive the private key with the public key. Some of the most popular asymmetric include Rivest–Shamir–Adleman (RSA), Elliptic curve cryptography (ECC), and Digital Signature Algorithm (DSA). Many internet-security protocols such as Transcript Layer Security (TLS) and Secure Sockets Layer (SSL) rely on asymmetric-key cryptography [22]. Thanks to these protocols, HTTPS and secure online communications are made possible.

A mix of both symmetric and asymmetric cryptography can also be used to establish secure communications [23]. For example, a symmetric-key encryption can be used to encrypt the data. This key is consequently encrypted using an asymmetric-key encryption and sent to the intended recipient via an unsecure communications channel. If an eavesdropper attempts to intercept the key, he would not be able view the key contents as it has been encrypted with a private key that only the receiver has. Once the recipient receives the encrypted key, he can simply decrypt it with his private key and use the decrypted symmetric key to establish symmetric cryptography.

3.3 Hashing

This type of cryptography utilizes a function called hash function that takes the data as an input, and outputs a unique fixed-sized string. Even a single letter change in the input would result in an output that is entirely different. The output of a hash function is generally called hash value or digest [24]. After the hash value is acquired, the original data is sent along with the hash value to the receiver. The receiver will then run the hash function once again on the data and compare the resulting hash value with the hash value he received. If they are identical, then the sent file hasn't been touched or modified. Hash functions are used for digital signatures and to maintain data integrity [25]. The most popular hashing algorithms include MD5 and SHA-1.

Encryption schemes that are based on factorizing numbers into their prime factors can be broken in theory. After all, there are many algorithms that perform this specific task of factorizing numbers into their prime factors. However, the difficulty arises when the number in question is huge. For example, it would take over 300 trillion years to crack a 2048-bit RSA encryption key on a "normal" classical computer. Even on the world's most powerful computer these days, it would still take over hundreds of thousands of years. The fact that RSA can be cracked after a very long time doesn't regard it as being insecure because by then the cracked message is of no importance.

CHAPTER 4

QUANTUM CRYPTOGRAPHY

4.1 The Risks

With the rise of quantum computers, computer scientists are more worried than ever about the future of public-key cryptography. Quantum computers are computers that exploit quantum mechanics and properties to solve mathematical problems that classical computers cannot [26]. The basic unit of information in quantum computers is qubit (quantum bit). Unlike normal classical bits, qubits are in a linear combination of both 0 and 1 at the same time. It is only when we measure it that it collapses to either 0 or 1 with a probability. This principle is called superposition as the qubit is in superposition of both 0 and 1 as long as it is not measured [27,28]. The state of a qubit is represented by a unit vector $|\psi\rangle$ that belongs to a Hilbert space $|\psi\rangle \in \mathcal{H}$. The simplest form is of a qubit with two states, $|0\rangle$ and $|1\rangle$. As mentioned before, the quantum system can exist in a superposition of both states at the same time as shown in equation 4.1 and 4.2:

$$|\psi\rangle=\alpha|0\rangle+\beta|1\rangle \tag{4.1}$$

$$|\alpha|^2+|\beta|^2=1 \tag{4.2}$$

Where α and β are arbitrary complex numbers and they should satisfy the equality in (2). The vector representation of $|0\rangle$ and $|1\rangle$ can be shown as follows in equation 4.3:

$$|0\rangle=\begin{pmatrix}1\\0\end{pmatrix}, |1\rangle=\begin{pmatrix}0\\1\end{pmatrix} \tag{4.3}$$

Another important property in qubits is entanglement. Entanglement plays an essential role not only in cryptographic applications, but also in other quantum computing applications. Two or more systems that are entangled with each other will be correlated and will experience a special kind of bond. A measurement or change on one of the quantum systems will result in an instant collapse of the other entangled system. The state of one of the systems or qubits cannot be described independently of the other paired systems. Scientists found out that for two quantum systems to be entangled, they don't need to be close to each other. In fact, even if the two entangled systems are on different ends of the world, they would still exhibit this phenomenon.

Quantum computer tremendously speeds up computational processes and can solve complex problems in a very short. Symmetric-key encryption algorithms like RSA can be broken rather quickly by means of quantum computers. As of now there are still no quantum computers that are capable of doing that, but the possibility that these quantum computers will exist over the next decade poses a huge security risk [29].

4.2 Post-Quantum Cryptography

The rise of quantum computers and the fear that classical cryptography schemes are no longer safe have led computer scientists to come up with a new set of symmetric-key algorithms called Post-Quantum Cryptography (PQC). The term "Post-Quantum" refers to the era where quantum computers exist and are in use. Although a fully functional quantum computer that is big enough to crack asymmetric-key encryptions in a matter of seconds or minutes doesn't exist just yet, but scientists are already planning ahead. PQC algorithms, like classical encryption schemes, are also based on complex mathematical computations. The difference is that PQC algorithms are thought to be secure on both quantum computers and classical supercomputers. Since they aren't based on factorizing huge numbers into their prime factors, PQC algorithms are thought to be secure against attacks performed on quantum computers [30]. It is worth noting that PQC algorithms are not based on any quantum mechanics or laws of physics. They do not need any special equipment to run them. The National Institute of Standards and Technology (NIST) is currently hosting a contest called Post-Quantum Cryptography Standardization, in which contestants will propose PQC algorithms. Research and development are still under way, and it might take a few years until PQC is adopted.

4.3 Quantum Cryptography

Quantum cryptography harnesses the quantum mechanics and laws of physics to accomplish cryptography. Quantum cryptography utilizes the principles and fundamentals of quantum physics to perform secure data encryption and decryption. While PQC proposes a solution to the risk that quantum computers pose on classical cryptography techniques, some people argue it is better to fight the quantum computers with actual quantum cryptography. Unlike mathematical equations and expressions, laws of physics cannot be cracked, which makes quantum cryptography truly secure against attacks. Quantum cryptography is based on two key principles of quantum mechanics: Heisenberg uncertainty principle and no-cloning theorem.

4.3.1 Heisenberg Uncertainty Principle

This principle was proposed by the German physicist Werner Heisenberg. It states that a particle's exact position and exact momentum cannot be measured at the same time. The formula can be shown in equation 4.4:

$$\Delta x \Delta p \geq \frac{h}{4\pi} \tag{4.4}$$

Where Δx donates the change or uncertainty in position, Δp donates the uncertainty in momentum, h is Planck constant which is equal to $6.62607015 \times 10^{-34}$ joule second. As we decrease the uncertainty of x in order to get an exact position, the uncertainty in the momentum will increase to satisfy the formula, and vice versa. Measuring the position of an object will consequently affect the momentum and measuring the momentum will affect the position. In the context of quantum cryptography, since we are dealing with quantum particles, this phenomenon is applied. We cannot measure the state of the quantum particle without affecting other information that are encoded within the particle [31]. This principle plays a vital role in the detection of potential third parties who are trying to eavesdrop, as will be discussed later on.

4.3.2 No-Cloning Theorem

The no-cloning theorem, which was proposed in 1982, states it is impossible to copy or clone the unknown state of a quantum system. This is where the fundamental difference between quantum cryptography and classical cryptography takes place. A third-party cannot simply copy the states that were sent by the original sender to the intended receiver, and thus he cannot gain information about the secret message while being undetected. Attempting to copy an unknown state requires measurement of said state, and this could lead to altering the transmitted quantum data or destroying the quantum state [32].

4.4 Quantum Key Distribution

Quantum Key Distribution is the most popular application of quantum cryptography. It allows for a secure key transmission using the laws of physics and the principles of quantum mechanics. QKD relies on the Heisenberg uncertainty principle and the no-cloning theorem to detect whether an eavesdropper is present. A third-party eavesdropper would need to measure the quantum state of the qubits, and thus the original parties will be alerted. After the key has been securely transmitted, a classical encryption algorithm can be applied to combine the message with the generated key to produce the ciphertext. The ciphertext will then be sent over classical communications channel.

QKD generally consists of two stages: the quantum transmission stage which involves sending/measuring quantum states between two parties (conventionally called Alice and Bob), and the classical post-processing stage which involves transforming the generated qubits into a secure secret key. Over the years, there have been many QKD protocols that have been proposed. All the proposed protocols can be categorized under two types: prepare-and-measure-based QKD protocols, and entanglement-based QKD protocols, both of which differ in the first stage of quantum transmission. In prepare-and-measure QKD protocols, Alice sends quantum states to Bob who measures them at his end. Prepare-and-measure protocols rely on Heisenberg uncertainty principle and the no cloning theorem to detect whether a third-party intruder is present. In entanglement-based protocols, both Alice and Bob have entangled qubits that they use to generate the key [16]. This study will focus on the first type; prepare-and-measure QKD protocols, more specifically, the BB84 QKD protocol.

4.5 BB84

BB84 is the first QKD protocol which was proposed by Bennett and Brassard in 1984 and has been proven to be unconditionally secure. There are two channels required: the quantum channel which is used to send quantum states, and the authenticated classical channel which is used to send the messages encrypted using the generated secret key [33,34] as shown in figure 4.1 below.

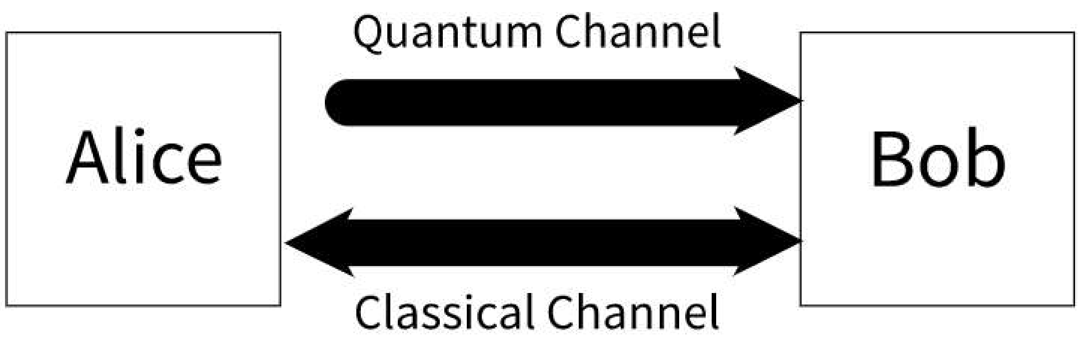


Figure 4.1 BB84 QKD protocol channels

Alice starts off by generating a random bit string and for each bit she randomly chooses one of the two conjugate polarization bases in which each bit will be encoded in. The two bases are the rectilinear (computational) basis for horizontal 0° and vertical 90° polarization which represent the bits 0 and 1 respectively, and the diagonal (Hadamard) basis for diagonal 45° and anti-diagonal 135° polarization which represent the bits 0 and 1 respectively. Alice then sends the polarized photons that represent the bit string that is polarized using one of the two bases to Bob without him knowing which basis is used for each bit.

Bob attempts to measure each photon in one of the two bases randomly. If Bob measures the photon using the same polarization basis that Alice used, then he will get a bit result that should match with Alice's bit. However, if Bob measured the photon in a different basis from what Alice used, then he will get a random result. For instance, if Alice encodes 0 using a photon that is in rectilinear basis (horizontal polarization) and Bob also uses rectilinear filter to measure the photon, then he ideally should also get 0. In the case that Bob measures the photon in the diagonal basis, then the probability of getting a zero is 50%. Likewise, if Alice encodes the bits in the diagonal basis and Bob measures them in the rectilinear basis, then he should also get a random result.

The next step is called the sifting stage. Alice announces publicly which polarization basis she used to encode each bit. Bob then compares the bases Alice used with the bases he used to measure the photons. In the case that Bob chose the same basis which Alice prepared the bit in, then they keep the decoded bit. If Bob chose a different basis from Alice's, then they discard the decoded bit as the result will be random. Since on average Bob chooses the correct basis 50% of the time, Alice and Bob end up with a key string that is approximately half of the original string.

Table 4.1 Procedure of BB84 protocol without the presence of Eve

Initial bit string	1	1	0	0	0	1	0	0
Alice's bases	×	×	+	×	+	+	×	×
Photon polarization sent	↖	↖	→	↗	→	↑	↗	↗
Bob's bases	+	×	+	+	×	+	+	×
Photon polarization measured	↑	↖	→	↑	↖	↑	→	↗
Bases comparison	✗	✓	✓	✗	✗	✓	✗	✓
Shared key		1	0			1		0

An example of a quantum transfer of states can be shown in table 4.1 [35]. Alice generates a random eight-bit string, with each bit prepared as a photon polarized in either the rectilinear basis + or the diagonal basis ×. Alice then sends the polarized photons to Bob who receives them without knowing in which basis Alice sent those photons. For each photon Bob chooses a random basis in which he measures the photon. Alice announces which bases she used to prepare each photon for Bob to compare. Out of 8 sent photons, Bob chooses the correct measurement basis four times which led him to obtain bit results that match with Alice's. However, in the other four times, Bob chooses the wrong measurement bases which led him to obtain totally random results. Bob discards the bits in which he chose the wrong measurement basis, and he ends up with a four-bit key.

This concludes the quantum transmission stage, and what follows is the classical post-processing stage. Alice and Bob use some of the remaining bits for parameter estimation. This stage is necessary to detect noise and possible eavesdropper presence. If no eavesdropper was detected, the other remaining bits will undergo error correction [36] and privacy amplification [37] in order to transform them into a secure pair of identical keys used for encryption and decryption.

It is normal to question how that scenario would play out if an eavesdropper (conventionally called Eve) is present, and how Alice and Bob will be notified to Eve's presence. Let us assume that Eve is the man-in-the-middle such that she intercepts the photons Alice sends before they reach Bob. Due to the no cloning theorem, Eve cannot copy the unknown states and send them to Bob. The only way to get information on the states would be to measure the photons. Just like Bob, she doesn't have any information as to which basis each photon is prepared in. Therefore, Eve chooses to measure the photons which Alice sends in bases which she chooses randomly. Eve statistically chooses the correct basis 50% of the time, and 50% of the time she chooses the wrong basis which results in loss of data. This is the key to determining the existence of an eavesdropper. Eve then prepares photons in the same bases she has measured in and sends the photons to Bob. Bob chooses to measure the photons in randomly chosen bases like before. Alice will subsequently announce publicly which polarization basis she used to encode each bit. Bob then compares Alice's bases which she used to prepare the photons with his own randomly chosen bases, and he discards the bits in which he chose different bases from Alice.

Alice and Bob will then compare some of the remaining bits with each other to determine whether an eavesdropper is present. If they do not match and there exist substantial inconsistencies in their corresponding bits in which the prepared/measured in the same bases, then it can be said that an eavesdropper is intercepting the key bits in the middle. The whole process will be aborted since they are certain that people are trying to interfere. An example of a quantum transfer of states with the existence of an eavesdropper is shown in table 4.2.

Table 4.2 Procedure of BB84 protocol with the presence of Eve

Initial bit string	1	1	0	0	0	1	0	0
Alice’s bases	×	×	+	×	+	+	×	×
Photon polarization sent	↖	↖	→	↗	→	↑	↗	↗
Eve’s bases	×	+	+	+	×	+	+	+
Photon polarization resent	↖	↑	→	→	↗	↑	→	↑
Bob’s bases	+	×	+	+	×	+	+	×
Photon polarization measured	↑	↗	→	→	↗	↑	→	↖
Bases comparison	✗	✓	✓	✗	✗	✓	✗	✓
Alice’s remaining bits		1	0			1		0
Bob’s remaining bits		0	0			1		1

In this example, both Alice and Bob chose matching basis four out of eight times. Therefore, they discard all the other bits in which their bases didn’t match and kept the four in which their bases were identical. They choose two out of four bits as parameter estimation in order to detect whether Eve is intervening. For the bit that corresponds to the first matching basis, both Alice and Bob chose to prepare and measure respectively in the diagonal basis. However, since Eve measured and retransmitted the bit to Bob using a photon that is polarized in the rectilinear basis, then Bob’s resulting bit doesn’t match with Alice’s. This, in turn, has notified the original two parties to the existence of an unauthorized party since the error rate when they compared the first two bits was 50%. Of course, in a practical scenario the bit length would be much larger. As we saw in this example, thanks to Heisenberg uncertainty principle and the no cloning theorem, Eve was detected and the whole key transmission process is aborted.

As for Eve, how much pieces of the raw key does she actually know? After Eve has chosen the bases in which each photon is measured in, she obtained the following 8-bit string:

1 1 0 0 0 1 0 1

Eve then has to discard the bits that she obtained by choosing the wrong bases to do the measurement after Alice announced the bases she used to prepare the photons. Therefore, Eve is left with the following three bits:

1 0 1

Eve also discards the bits where Bob measured in the wrong bases from what Alice prepared in. Eve is left with the following two bits:

0 1

Finally, Eve discards the bits that were chosen to be parameter estimation because Alice and Bob will not include them as part of the final key. Therefore, Eve is left with only one bit of information about the final key:

1

Out of this entire protocol, Eve was only able to learn one bit out of eight bits, and that bit wasn't even used in the final key since Alice and Bob decided to abort the protocol after they discovered the existence of an eavesdropper because of Heisenberg uncertainty principle and the no cloning theorem.

4.6 E91

E91 QKD protocol is an entanglement based QKD protocol that utilizes an entangled photon source to send a pair of entangled photons such that Alice gets one of the photons, and Bob receives the other paired photon. Any change in one of the entangled photons will result in a change to the other photon regardless of the distance between them as discussed earlier in the qubit properties.

CHAPTER 5

DESIGN AND RESULTS

This chapter will serve as a discussion of the design and implementation of the image encryption method that is based on the BB84 QKD protocol, as well as analyze the results that are obtained from running several tests that are meant to assess the efficiency and performance of the proposed method. The results are compared with E91 QKD protocol.

5.1 Design

5.1.1 Key Generation

A simulation of the BB84 protocol is implemented using Python 3.10 in which it takes the number of qubits that Alice will generate as the input, and it outputs the final key. The application operates in two modes:

1) No eavesdropping mode: in this mode, Eve is not present. Therefore, whenever Alice and Bob choose the same basis for preparing/measuring, Bob should be able to acquire the same value of bits as Alice's. A final key is successfully generated, and it used in the next step of image encryption.
2) Eavesdropping mode: Eve is present in this mode, and her presence alerts both Alice and Bob due to mismatching bits, thus the entire key generation process is aborted.

The resulting key size is on average equal to half the size of the input since Bob chooses the correct basis to measure the qubits approximately 50% of the time. The required size of the key depends on the image size. Larger images require keys that are big enough to encrypt them with. In the next step, the generated key will be combined with the image using a proposed method to form the encrypted image that will be sent to Bob as shown in figure 5.1 [38]. E91 protocol's results are obtained using Evan Anderson's E-91 protocol simulation which is released under the GNU General Public License version 3 [39].

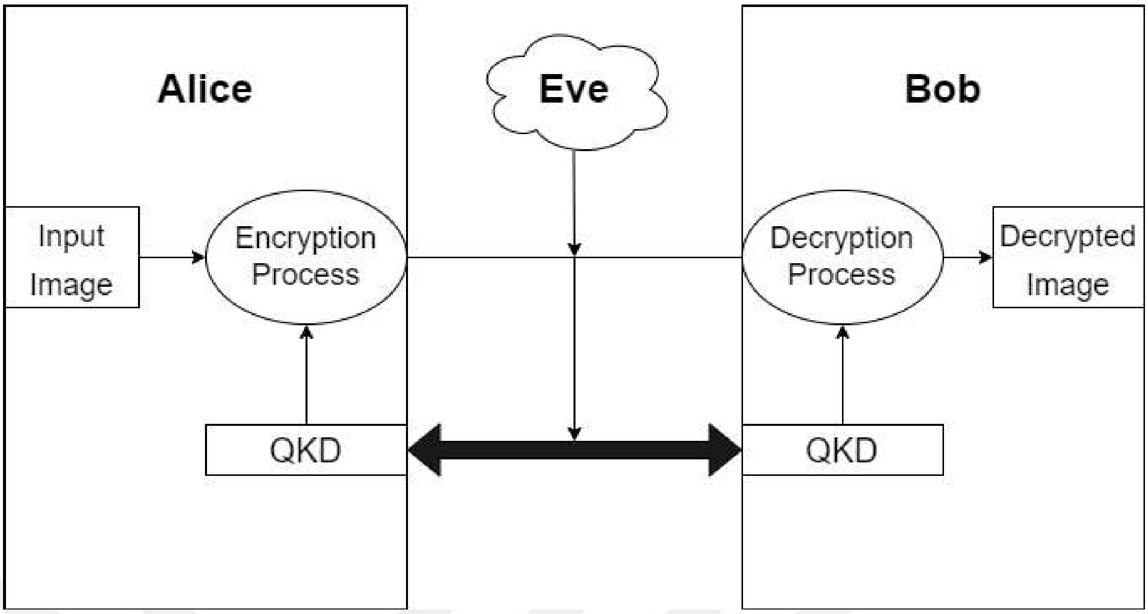


Figure 5.1 Schematic diagram for the QKD protocol-based encryption scheme

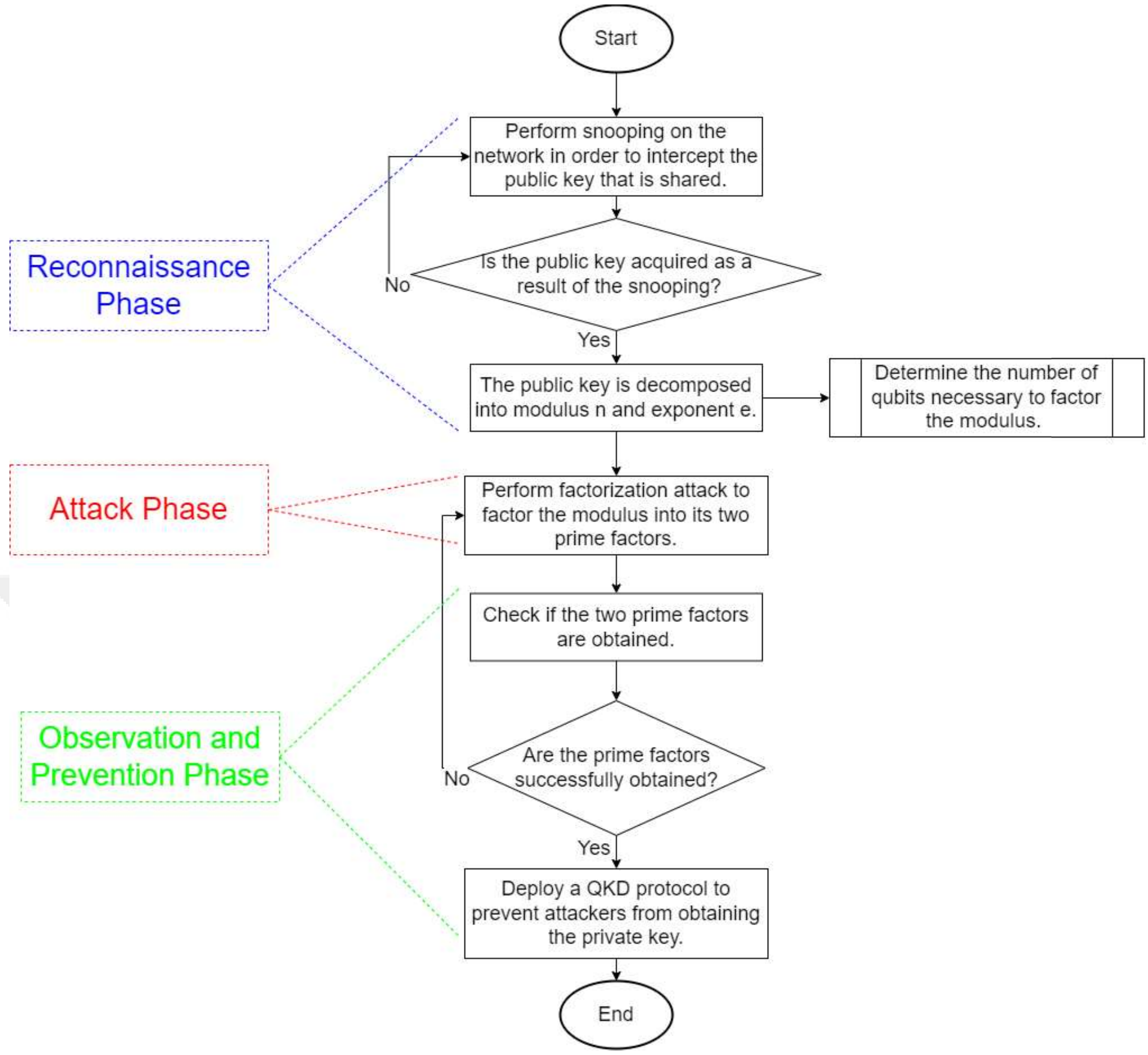


Figure 5.2 Analysis steps for a factorization attack and its prevention using QKD protocol

Figure 5.2 shows an attack carried out to obtain the key using Shor's algorithm on a quantum computer, and how a QKD protocol can be implemented to ensure secure key transmission.

Figure 5.3 shows the flowchart for the program that generates the key used in encryption.

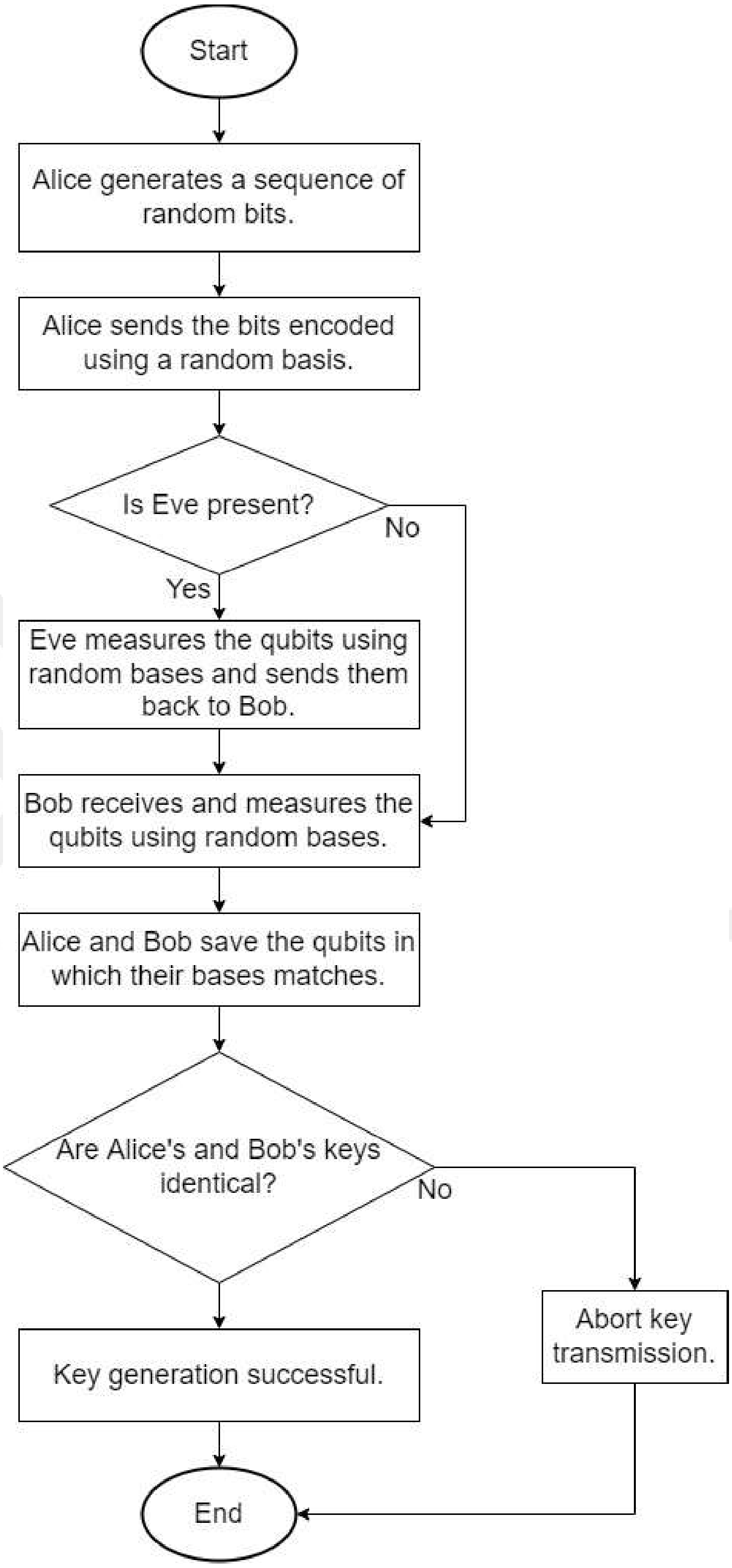


Figure 5.3 Flowchart of the program

5.1.2 Encrypting the Image with the Generated Key

The smallest unit in an image is called a pixel. A grid of pixels is what makes an image. Binary images (often referred to as bitmap images) consist of pixels that only carry one of two values: either 0 or 1. Since each pixel is 1 bit in size, it cannot have any more colors, shades, or intensities than one of the exact two values. For example, the two colors that can be represented by a binary image are black and white. Greyscale images are images that consist of 8-bit pixels. While binary images can only have one of the two values (black or white), greyscale images have shades between those two colors. They range from the black (which has the weakest intensity), to some grey levels in between, and finally to white (which has the strongest intensity). RGB images are colored images that have a color scheme made of red, green, and blue. RGB images are made of three layers or channels, such that each channel is made of 8-bit pixels that represent the intensity values of one of the three colors respectively, as shown in figure 5.4. Therefore, a RGB image pixel is of size 24 bits as it represents the three colors with 8 bits each.

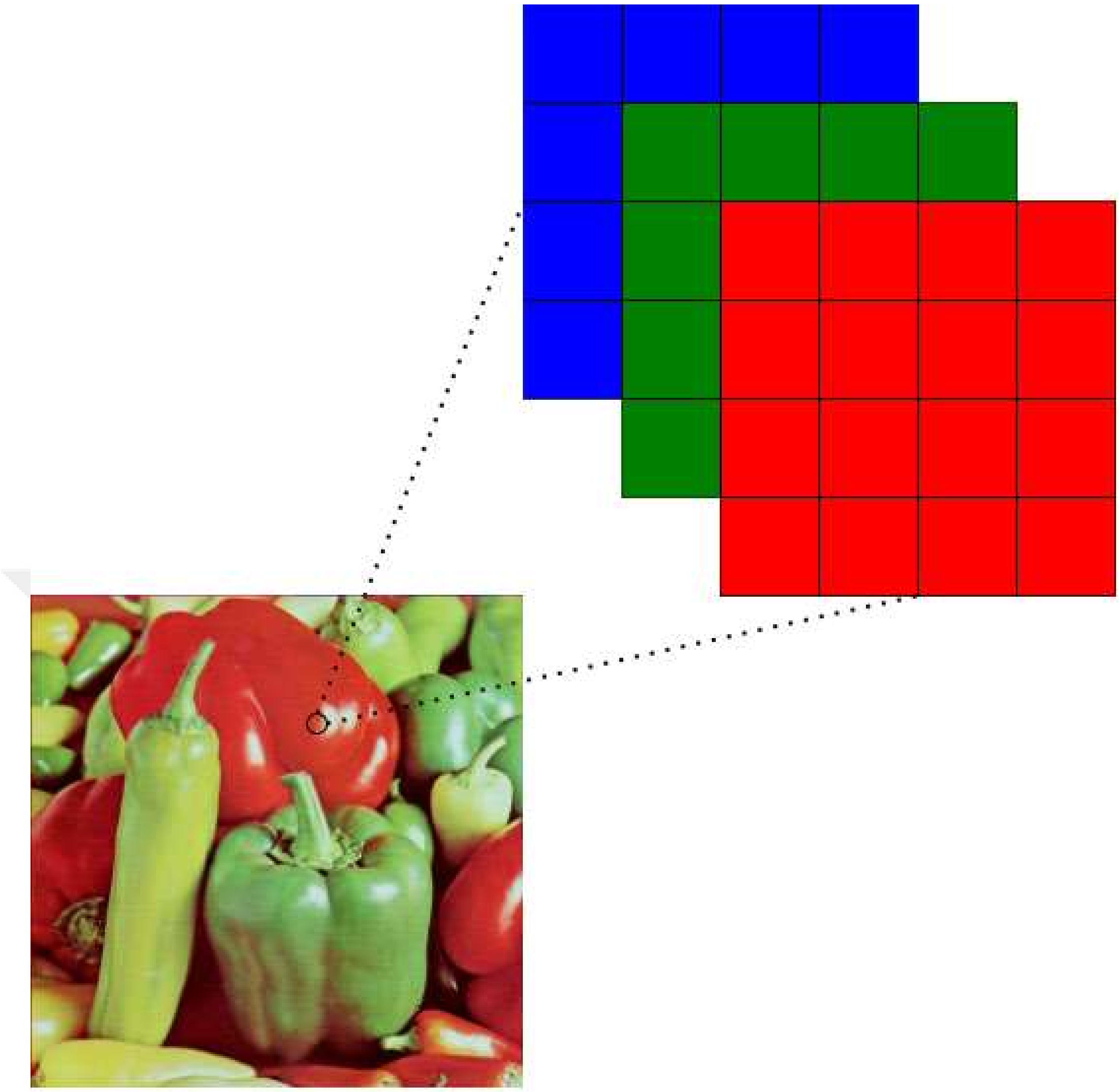


Figure 5.4 Pixels of RGB images

RGB images are used in my work as they are encrypted using the key generated by the simulated BB84 QKD protocol. Test images of different sizes are encrypted including 256x256 pixel images and 512x512 pixel images.

It is worth noting that the key which was generated in the previous step has to be at least the same size as that of the image. For example, 256x256 pixel RGB images are made out of 65,536 pixels, each of which containing 24 bits of information (8 bits for each of the three components). Therefore, the minimum key size required to encrypt such image is $65{,}536 \times 24 = 1{,}572{,}864$ bits, or 192 kilobytes. Similarly, to encrypt a 512x512 pixel image, a key size of at least 768 kilobytes is necessary.

The steps to encrypt the plain image using the key generated by the BB84 QKD protocol simulation are as follow:

1) The generated key that has the same dimensions as the plain image is split into three components that represent red, green, and blue respectively.
2) The plain image that will be encrypted is also split into its three components.
3) The red component of the key is XORed with the red component of the plain image to form the red component of the cipher image.
4) The green component of the key is combined with the green component of the plain image using modular arithmetic to form the green component of the cipher image.
5) The first four-bit planes of the key's blue component are XORed with the first four-bit planes of the plain image's blue component, while the last four-bit planes of both the key's blue component and the plain image's blue component are combined together using modular arithmetic to form the blue component of the cipher image.
6) The resulting three image components from the previous three steps are subsequently joined to form the final encrypted image.
7) The encrypted image is sent via a communications link to Bob who possesses the secret key to decrypt the encrypted image.

Figure 5.5 shows a schematic diagram for the proposed method.

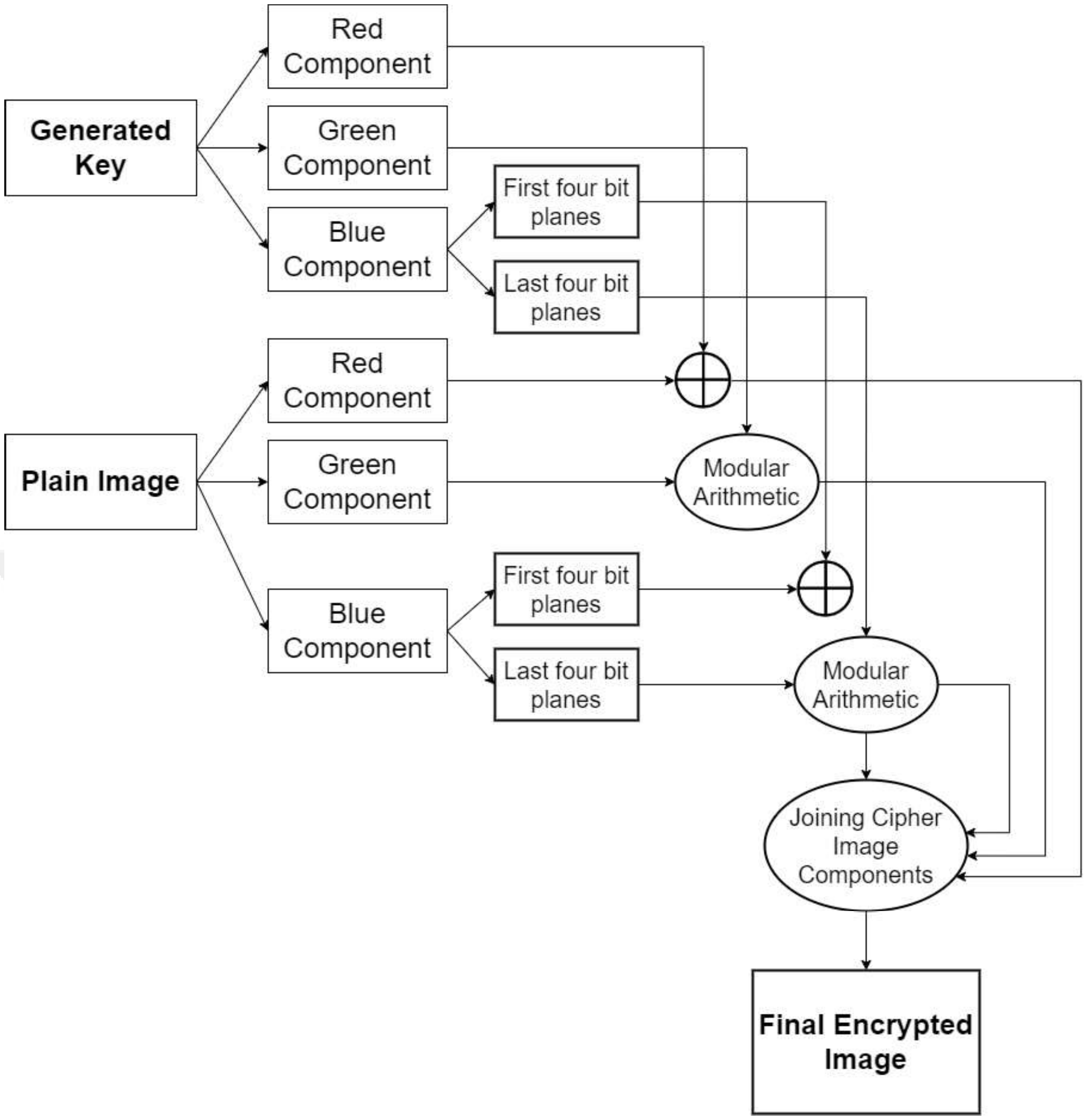


Figure 5.5 Schematic diagram for the proposed image encryption method

As for decrypting the cipher image at the recipient's side, the same steps are done in reverse as follows:

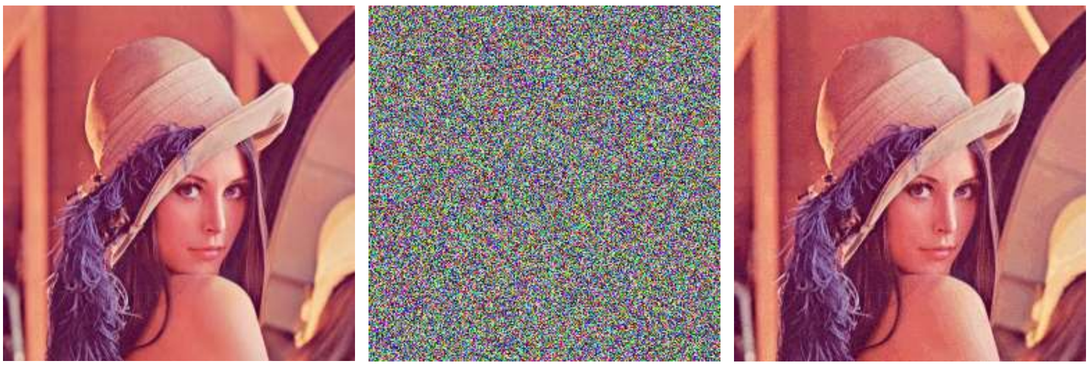
1) The encrypted image is decomposed into its three components.
2) The red component of the key is XORed with the red component of the encrypted image to form the red component of the plain image.
3) The green component of the key is combined with the green component of the encrypted image using modular arithmetic to form the green component of the plain image.

4) The first four-bit planes of the key's blue component are XORed with the first four-bit planes of the encrypted image's blue component, while the last four-bit planes of both the key's blue component and the encrypted image's blue component are combined together using modular arithmetic to form the blue component of the plain image.
5) The resulting three image components are joined to form the original plain image.

5.2 Results

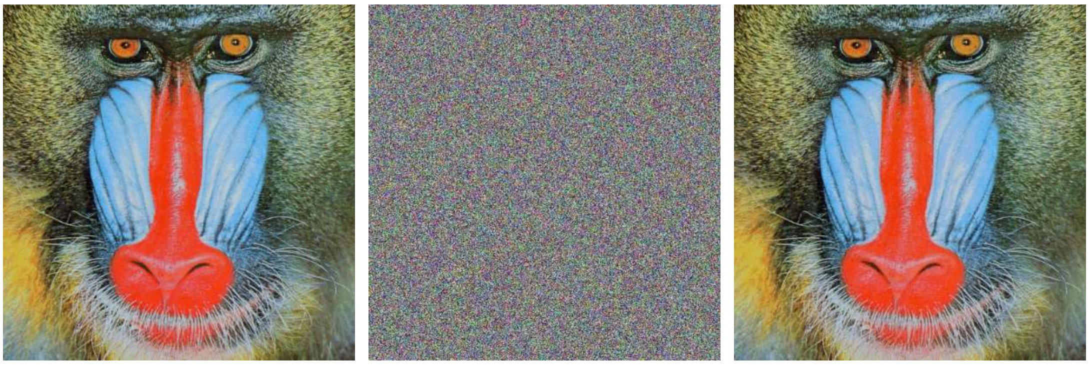
To test the proposed method, the QKD protocol simulation and encryption method were run on four standard test images in the field of image processing: Lena, mandril, peppers, and airplane. Two different image sizes were used, with the Lena image being a 256x256 pixel image, while the other three images are of size 512x512 pixels.

Figures 5.6, 5.7, 5.8, and 5.9 show the results of encrypting the images using the proposed method and decrypting it to obtain the original image back. The key is generated using BB84 QKD protocol. In each of the figures, (a) represents the original plain image, (b) is the result of encrypting the image using the key obtained from the QKD protocol implementation, and (c) is the result of decrypting the encrypted image using the same key to acquire the original image.



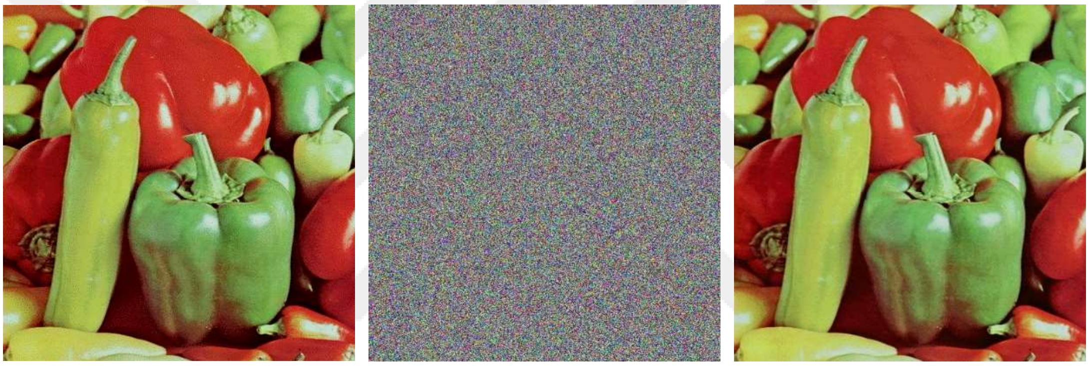
(a) (b) (c)

Figure 5.6 Results of encrypting "Lena" dataset using BB84 (a) original image (b) encrypted image (c) decrypted image



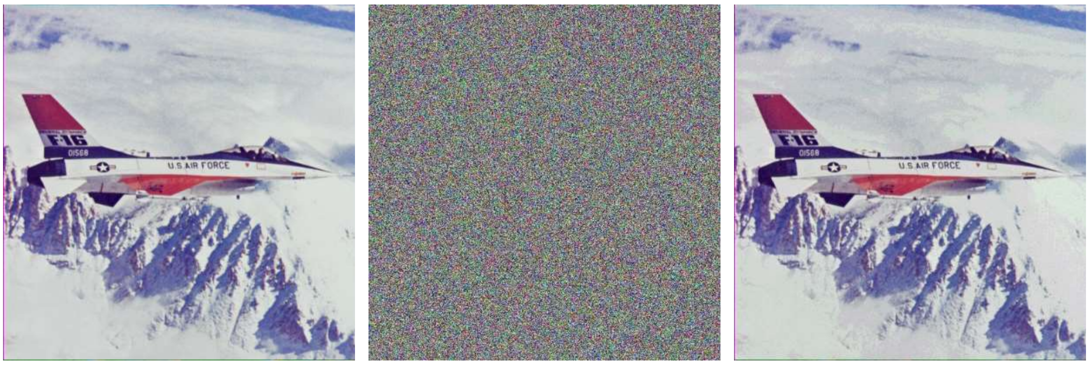
(a) (b) (c)

Figure 5.7 Results of encrypting "Mandril" dataset using BB84 (a) original image (b) encrypted image (c) decrypted image



(a) (b) (c)

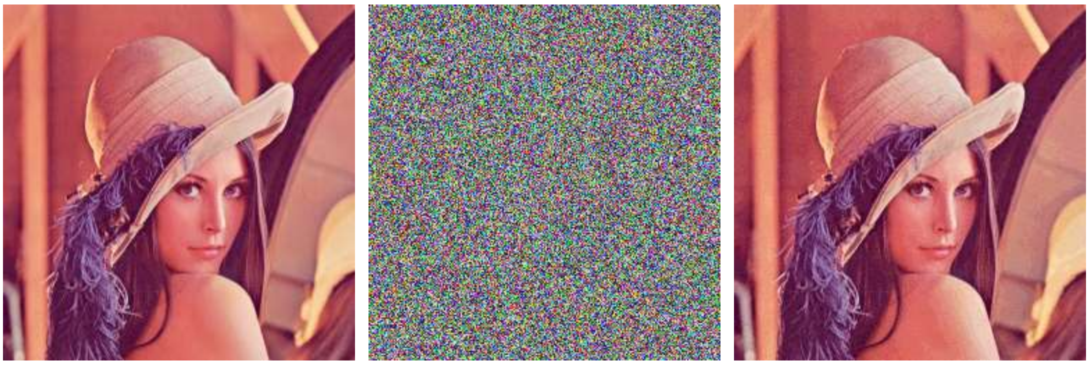
Figure 5.8 Results of encrypting "Peppers" dataset using BB84 (a) original image (b) encrypted image (c) decrypted image



(a) (b) (c)

Figure 5.9 Results of encrypting "Airplane" dataset using BB84 (a) original image (b) encrypted image (c) decrypted image

Figures 5.10 shows the result of encrypting the Lena image using the proposed method and decrypting it to obtain the original image back. The key is generated using E91 protocol. In the figure, (a) represents the original plain image, (b) is the result of encrypting the image using the key obtained from the E91 QKD protocol implementation, and (c) is the result of decrypting the encrypted image using the same key to acquire the original image.



(a) (b) (c)

Figure 5.10 Results of encrypting "Lena" dataset using E91 (a) original image (b) encrypted image (c) decrypted image

5.3 Analysis

In order to ensure that the proposed method is robust and has great resistance against different types of attacks, standard attack tests and statistical analysis are applied to the image data.

5.3.1 Correlation Analysis of Two Adjacent Pixels

The correlation in an image describes the relationship between two adjacent pixels. Plain images feature high correlation between two neighboring pixels. Neighboring pixels can be described as pixels that are adjacent either in the horizontal direction, vertical direction, or diagonal direction. The correlation of any two neighboring pixels is given by formula 5.1:

$$\mathrm{Coef} = \frac{\mathrm{E}\Big(\big(\mathrm{x}-\mathrm{E(x)}\big)\big(\mathrm{y}-\mathrm{E(y)}\big)\Big)}{\sqrt{\mathrm{D(x)D(y)}}} \tag{5.1}$$

where

$$\mathrm{E(x)} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{x_i} \tag{5.2}$$

$$\mathrm{D(x)} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \big(\mathrm{x_i}-\mathrm{E(x)}\big)^2 \tag{5.3}$$

When the result is close to 1, then it means that the two sequences x and y have strong correlation, whereas a result close to 0 means that both sequences have weak correlation. A strong encryption algorithm should result in an encrypted image that exhibits weak correlation between adjacent pixels. Figure 5.11 shows a visual representation of the correlation between two adjacent pixels for the red channels of both the plain Lena image and the BB84 encrypted Lena image. The x-axis represents the value of the pixel, while the y-axis represents the value of the neighboring pixel in one of the three directions. As shown in the figure, the points are concentrated on the diagonal line for the plain image, which indicates that there is a strong correlation between neighboring pixels. However, the points are randomly distributed in the data range, which indicates very weak correlation. Therefore, an attacker cannot gain any valuable information from the encrypted image since each pixel doesn't give any information about its adjacent pixels.

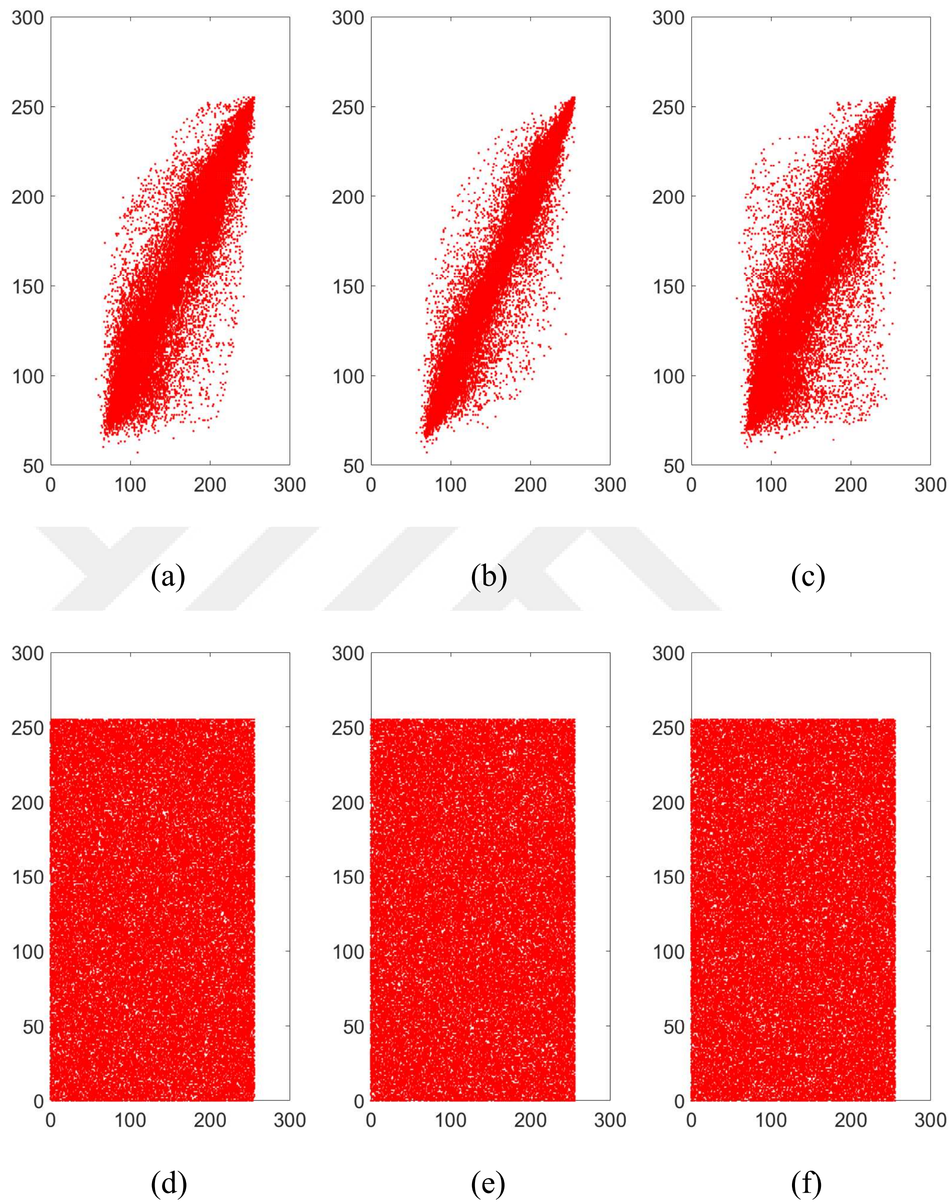


Figure 5.11 Correlation between two adjacent pixels results for the red channels of both the plain and encrypted Lena images (a) plain horizontal (b) plain vertical (c) plain diagonal (d) encrypted horizonal (e) encrypted vertical (f) encrypted diagonal

Similarly, the correlation analysis for the green channels of the plain Lena image and the encrypted Lena image in the three directions is shown in figure 5.12. For the plain

Lena image, points are concentrated along the diagonal axis, which means that the correlation coefficient is close to 1. The points are randomly distributed in the encrypted image, which means that the correlation coefficient is close to 0.

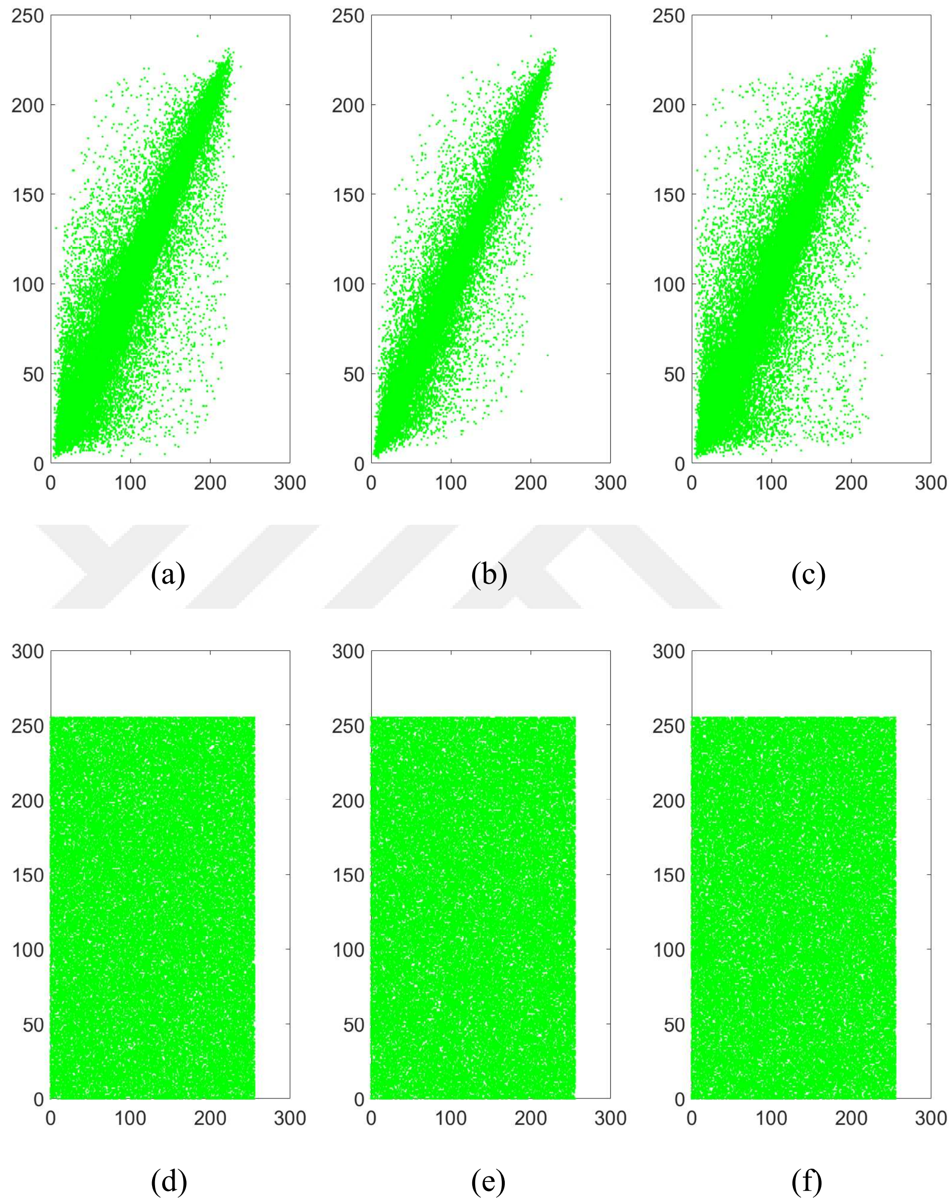
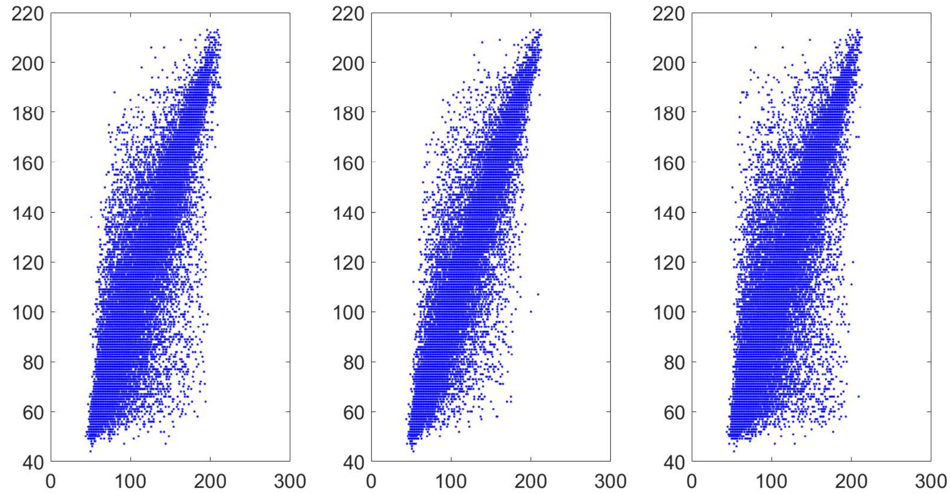
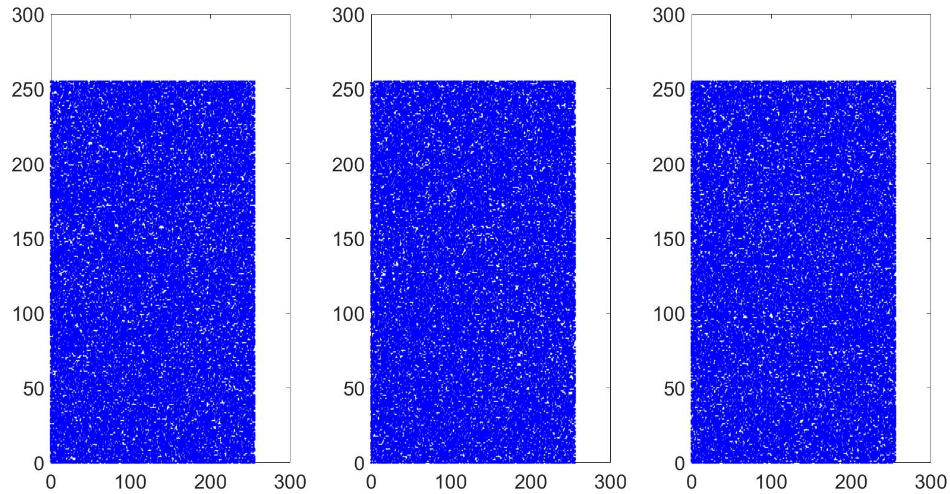


Figure 5.12 Correlation between two adjacent pixels results for the green channels of both the plain and encrypted Lena images (a) plain horizontal (b) plain vertical (c) plain diagonal (d) encrypted horizonal (e) encrypted vertical (f) encrypted diagonal

Finally, the results for the blue channels of both the plain and encrypted images are shown in figure 5.13.



(a) (b) (c)



(d) (e) (f)

Figure 5.13 Correlation between two adjacent pixels results for the blue channels of both the plain and encrypted Lena images (a) plain horizontal (b) plain vertical (c) plain diagonal (d) encrypted horizonal (e) encrypted vertical (f) encrypted diagonal

The quantitative results and values for the four images in both their plain and BB84 encrypted form are given in table 5.1. As shown in the table, the values for the plain images are all close to 1, while the values for their encrypted counterparts are close to 0.

Table 5.1 Correlation of two adjacent pixels results using BB84

Image	Plain / Encrypted	Correlation Direction	Red	Green	Blue
Lena	Plain	Horizontal	0.9558	0.9401	0.9189
		Vertical	0.9781	0.9695	0.9495
		Diagonal	0.9336	0.9180	0.8948
	Encrypted	Horizontal	0.0039	0.0020	0.0039
		Vertical	0.0029	0.0041	0.0050
		Diagonal	0.0019	0.0009	0.0032
Mandril	Plain	Horizontal	0.9105	0.8594	0.8953
		Vertical	0.8595	0.7755	0.8697
		Diagonal	0.8474	0.7434	0.8296
	Encrypted	Horizontal	0.0038	0.0039	0.0019
		Vertical	0.0009	0.0001	0.0001
		Diagonal	0.0014	0.0005	0.0017
Peppers	Plain	Horizontal	0.9635	0.9811	0.9665
		Vertical	0.9663	0.9818	0.9664
		Diagonal	0.9564	0.9687	0.9478
	Encrypted	Horizontal	0.0032	0.0013	0.0014
		Vertical	0.0005	0.0031	0.0006
		Diagonal	0.0020	0.0014	0.0001
Airplane	Plain	Horizontal	0.9726	0.9578	0.9640
		Vertical	0.9568	0.9678	0.9353
		Diagonal	0.9343	0.9326	0.9146
	Encrypted	Horizontal	0.0041	0.0035	0.0006
		Vertical	0.0026	0.0011	0.0013
		Diagonal	0.0020	0.0007	0.0014

For comparison, the quantitative results, and values for the images in both their plain and E91 encrypted forms are given in table 5.2.

Table 5.2 Correlation of two adjacent pixels results using E91

Image	Plain / Encrypted	Correlation Direction	Red	Green	Blue
Lena	Plain	Horizontal	0.9558	0.9401	0.9189
		Vertical	0.9781	0.9695	0.9495
		Diagonal	0.9336	0.9180	0.8948
	Encrypted	Horizontal	0.0049	0.0062	0.0088
		Vertical	0.0117	0.0013	0.0013
		Diagonal	0.0002	0.0111	0.0042
Mandril	Plain	Horizontal	0.9105	0.8594	0.8953
		Vertical	0.8595	0.7755	0.8697
		Diagonal	0.8474	0.7434	0.8296
	Encrypted	Horizontal	0.0005	0.0001	0.0000
		Vertical	0.0003	0.0009	0.0047
		Diagonal	0.0002	0.0030	0.0010
Peppers	Plain	Horizontal	0.9635	0.9811	0.9665
		Vertical	0.9663	0.9818	0.9664
		Diagonal	0.9564	0.9687	0.9478
	Encrypted	Horizontal	0.0019	0.0049	0.0028
		Vertical	0.0000	0.0025	0.0021
		Diagonal	0.0001	0.0002	0.0037
Airplane	Plain	Horizontal	0.9726	0.9578	0.9640
		Vertical	0.9568	0.9678	0.9353
		Diagonal	0.9343	0.9326	0.9146
	Encrypted	Horizontal	0.0016	0.0027	0.0008
		Vertical	0.0016	0.0001	0.0035
		Diagonal	0.0021	0.0010	0.0012

5.3.2 Histogram Analysis

An image histogram is a graphical representation of how the various intensities in an image are distributed. Depending on the image, plain images usually have uneven distribution of intensities since there exists colors and intensities that are more dominant than others. A good encryption scheme, however, should even out the intensity distribution and make it as flat as possible. Figures 5.14, 5.15, 5.16 and 5.17 show the histograms for the plain image and the BB84 encrypted image of Lena, Mandril, Peppers, and Airplane respectively.

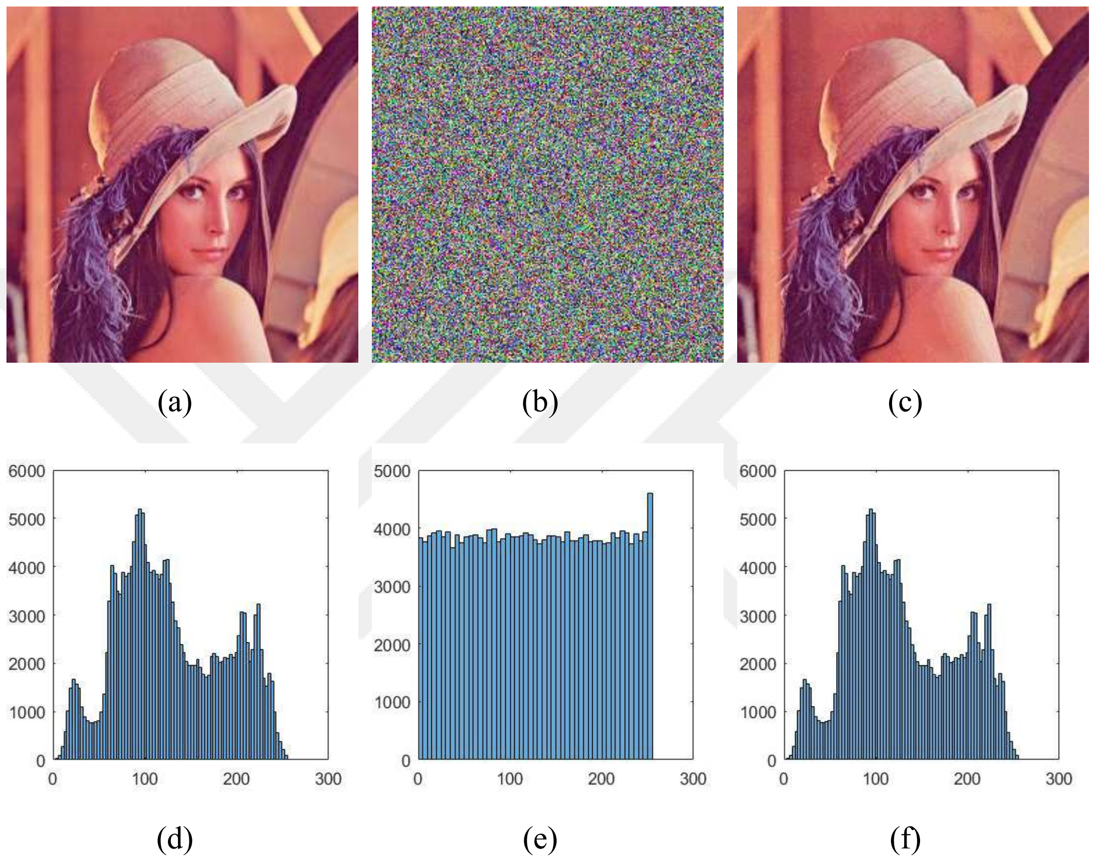
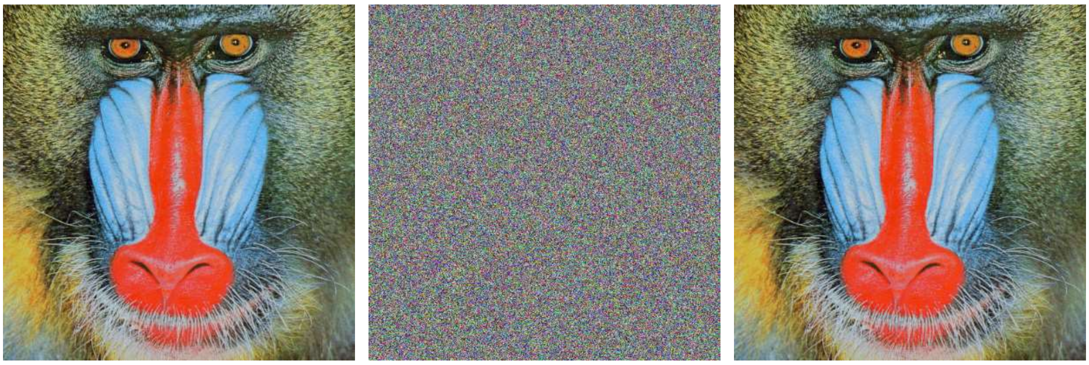
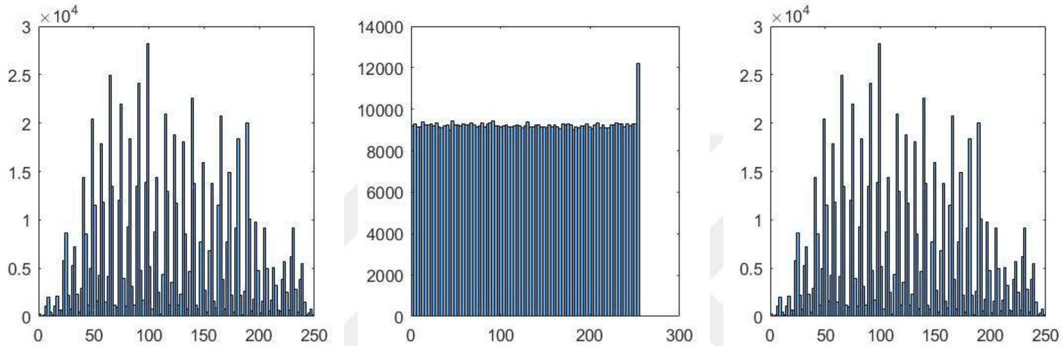


Figure 5.14 Histogram analysis of "Lena" image (a) original image (b) BB84 encrypted image (c) decrypted image (d) original mage histogram (e) encrypted image histogram (f) decrypted image histogram

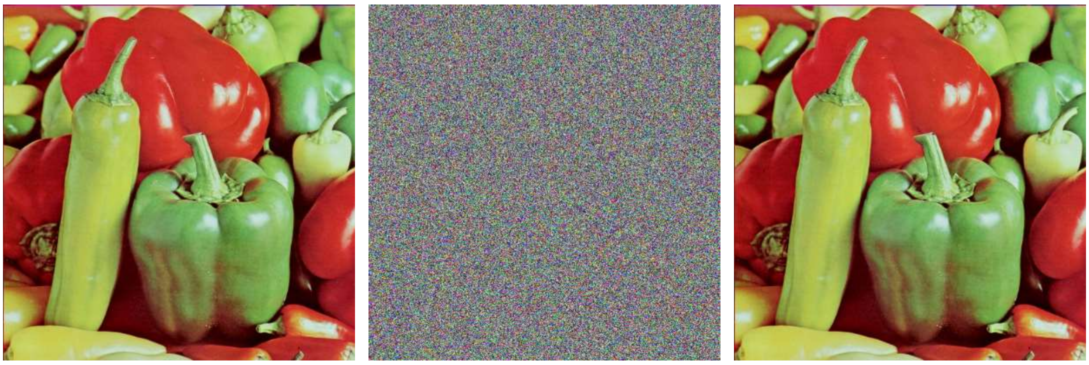


(a) (b) (c)

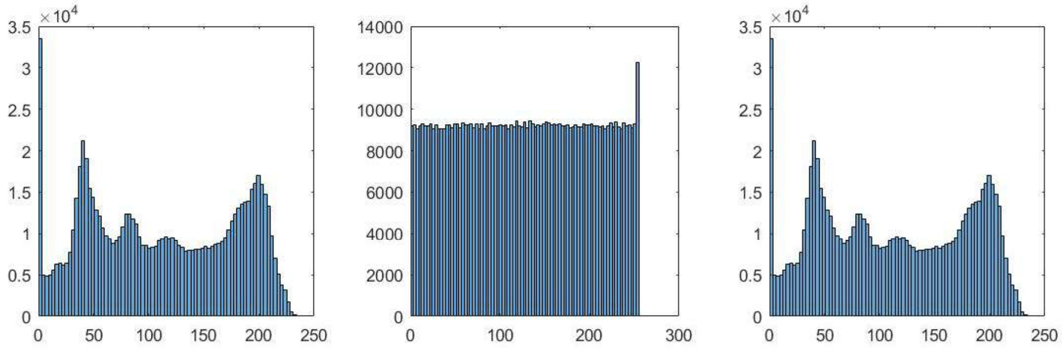


(d) (e) (f)

Figure 5.15 Histogram analysis of "Mandril" image (a) original image (b) encrypted image (c) decrypted image (d) original mage histogram (e) encrypted image histogram (f) decrypted image histogram

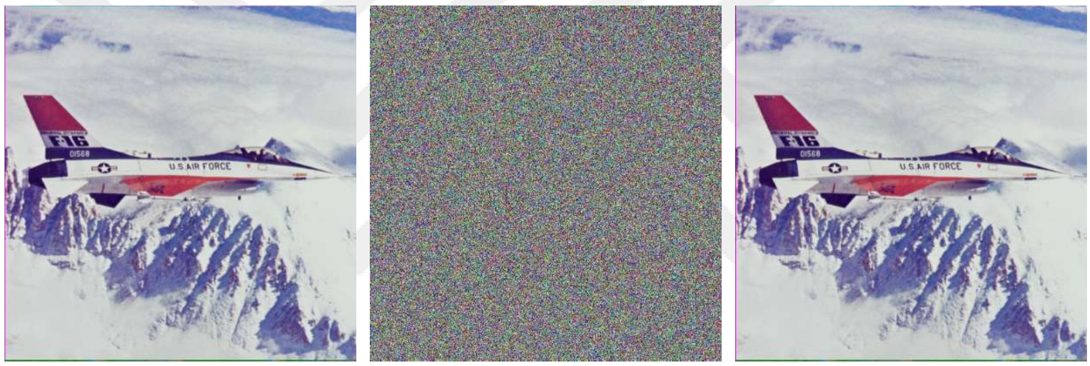


(a) (b) (c)

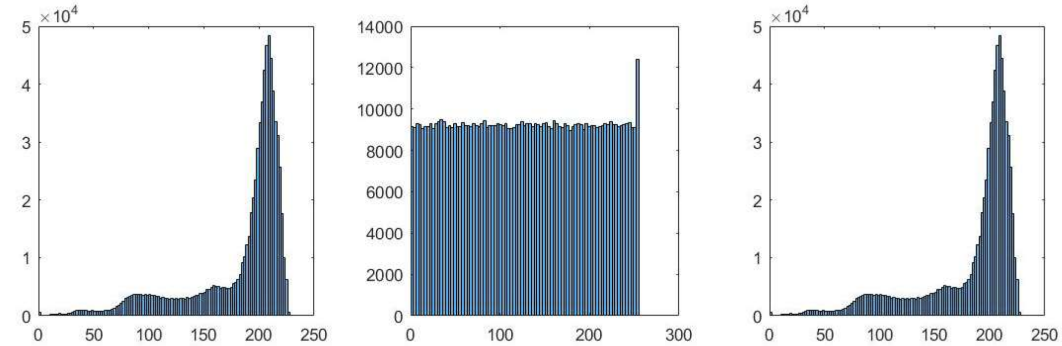


(d) (e) (f)

Figure 5.16 Histogram analysis of "Peppers" image (a) original image (b) encrypted image (c) decrypted image (d) original mage histogram (e) encrypted image histogram (f) decrypted image histogram



(a) (b) (c)



(d) (e) (f)

Figure 5.17 Histogram analysis of "Airplane" image (a) original image (b) encrypted image (c) decrypted image (d) original mage histogram (e) encrypted image histogram (f) decrypted image histogram

As shown in the figures, the histograms of the plain images have uneven distribution as specific intensities have much higher frequencies than other intensities. However, the histograms for the encrypted images have more even distribution among all the intensities. This makes it harder for the adversary to gain valuable information by using statistical attacks on the encrypted images.

Similarly, figure 5.18 shows the histograms for the plain image and the E91 encrypted image of Lena.

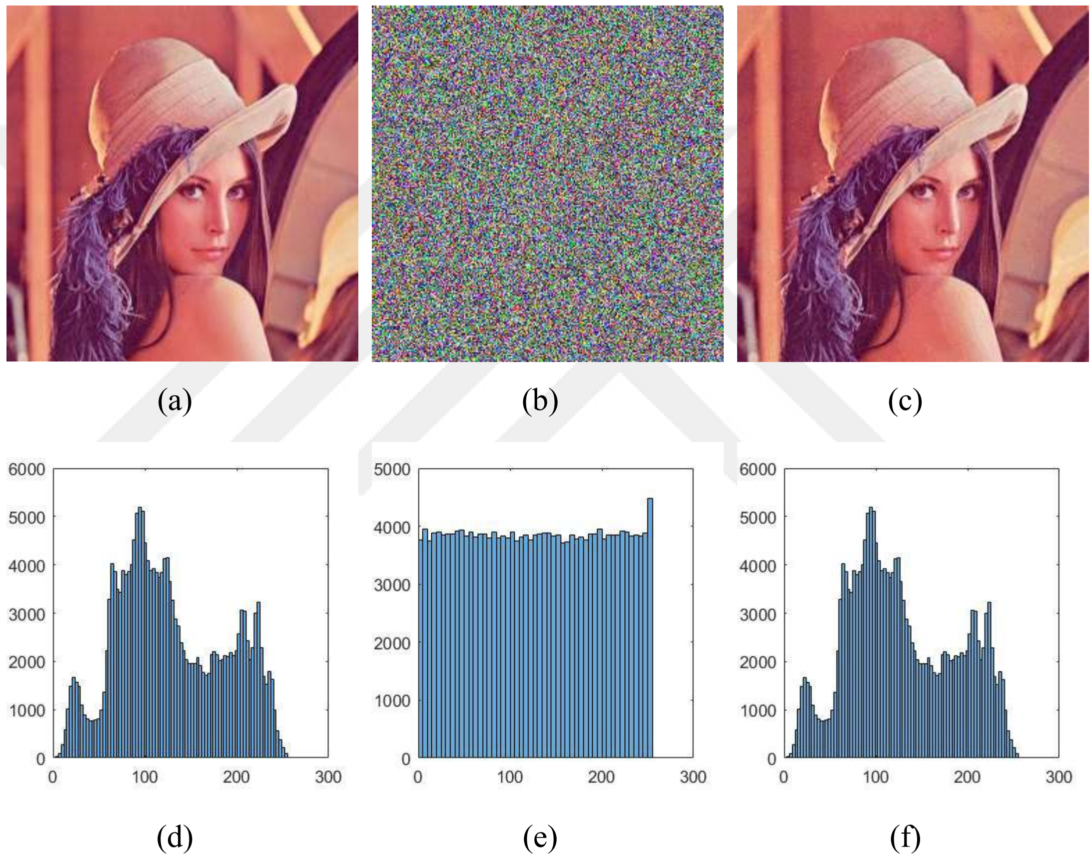


Figure 5.18 Histogram analysis of "Lena" image (a) original image (b) E91 encrypted image (c) decrypted image (d) original mage histogram (e) encrypted image histogram (f) decrypted image histogram

5.3.3 Information Entropy

The concept of information entropy, which was introduced by Shannon in 1948 [40], is a measure of unpredictability and randomness in the content. Entropy tests whether an image is random enough to have its pixel values uniformly distributed among the entire range of intensities. The information entropy is given by equation 5.4:

$$H(K)=\sum_{i=1}^{256} P(K_i)\log_2 \frac{1}{P(K_i)} \tag{5.4}$$

Where K_i represents one of the intensities. Since the entropy analysis will be applied to each of the three channels of the RGB image separately, the number of various intensities an 8-bit pixel has is 256. $P(K_i)$ denotes the probability of intensity K_i. Assuming that an image has its intensities distributed with equal probabilities, then that would result in the ideal value of $H(K) = 8$. Plain images feature correlation between its pixel intensities, which results in a lower entropy value. Table 5.3 presents the entropy values for the plain images and their BB84 encrypted counterparts, as well as the average for the entire image. As shown in the table, the values for the encrypted images are higher than that of the plain images, and they are close to 8. This indicates that the values are uniformly distributed.

Table 5.3 Entropy results for BB84

Image	Plain / Encrypted	Red	Green	Blue	Average
Lena	Plain	7.2353	7.5683	6.9176	7.2404
	Encrypted	7.9970	7.9974	7.9974	7.9973
Mandril	Plain	6.9294	6.3175	7.2895	6.8455
	Encrypted	7.9993	7.9993	7.9993	7.9993
Peppers	Plain	7.3388	7.4963	7.0583	7.2978
	Encrypted	7.9993	7.9993	7.9992	7.9993
Airplane	Plain	6.7178	6.7990	6.2138	6.5769
	Encrypted	7.9994	7.9992	7.9993	7.9993

Table 5.4 presents the entropy values for the layers of the four plain image and their E91 encrypted counterpart, as well as the average for the entire images.

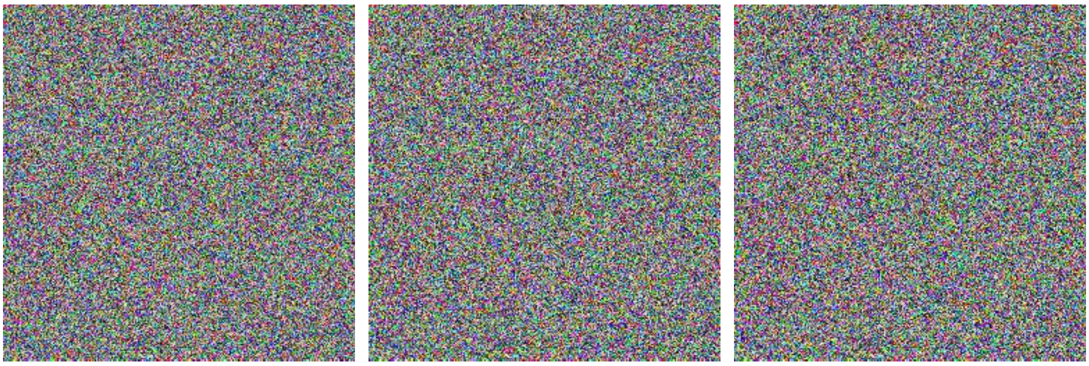
Table 5.4 Entropy results for E91

Image	Plain / Encrypted	Red	Green	Blue	Average
Lena	Plain	7.2353	7.5683	6.9176	7.2404
	Encrypted	7.9971	7.9973	7.9976	7.9974
Mandril	Plain	6.9294	6.3175	7.2895	6.8455
	Encrypted	7.9994	7.9992	7.9993	7.9993
Peppers	Plain	7.3388	7.4963	7.0583	7.2978
	Encrypted	7.9993	7.9994	7.9991	7.9993
Airplane	Plain	6.7178	6.7990	6.2138	6.5769
	Encrypted	7.9993	7.9993	7.9993	7.9993

5.3.4 Noise Attack

After the image has been encrypted using the secret key that is generated from the QKD protocol, the image might get slightly altered by different types of noise. Transmitting images over networks may result in noise contamination or data loss. This occurs during the transmission of the encrypted image from the sender to the receiver via a classical channel. A good encryption algorithm should have the ability to successfully decrypt the encrypted image after noise or data loss attacks [41]. The final decrypted image at the receiver's end must be reconstructed without suffering from a great loss in its quality or any significant degradation in its features. To test the algorithm's robustness against noise attacks, 3 different types of noise will be applied to each of the encrypted images.

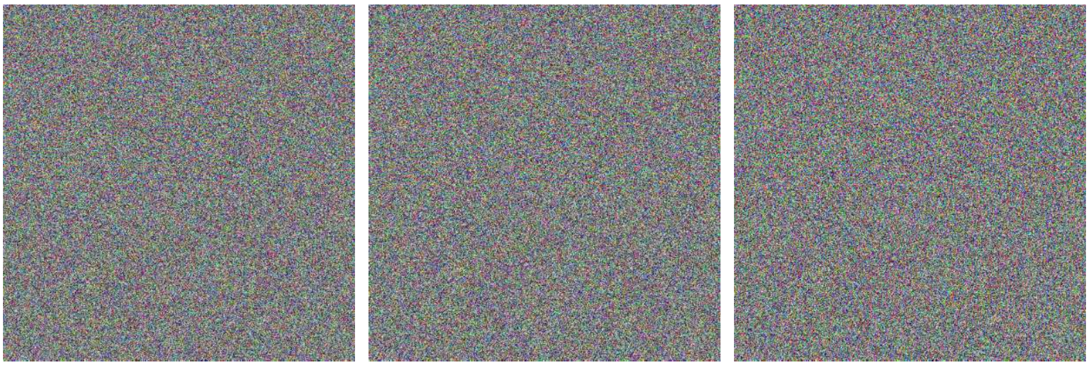
Firstly, salt and pepper noise with densities of (0.05, 0.075, 0.1) is applied to both the Lena image and the Mandril image. The noisy BB84 encrypted images along with their corresponding decrypted images are shown in figures 5.19 and 5.20.



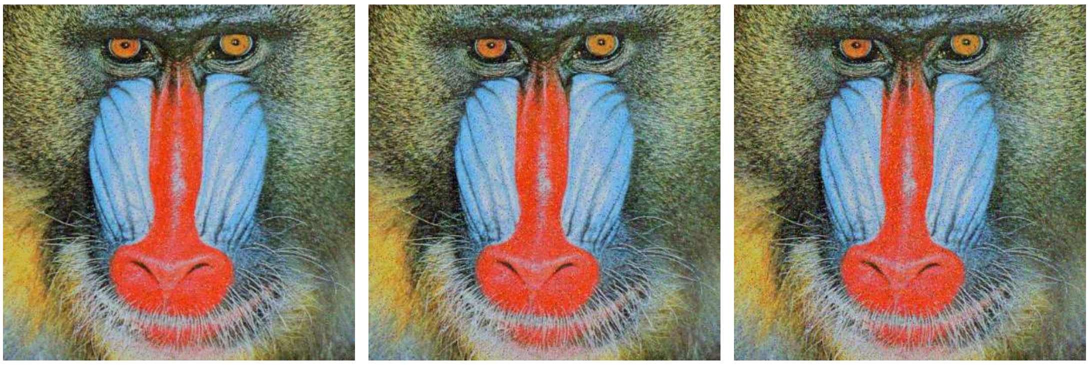
(a) (b) (c)

(d) (e) (f)

Figure 5.19 Salt and pepper noise results for "Lena" image using BB84



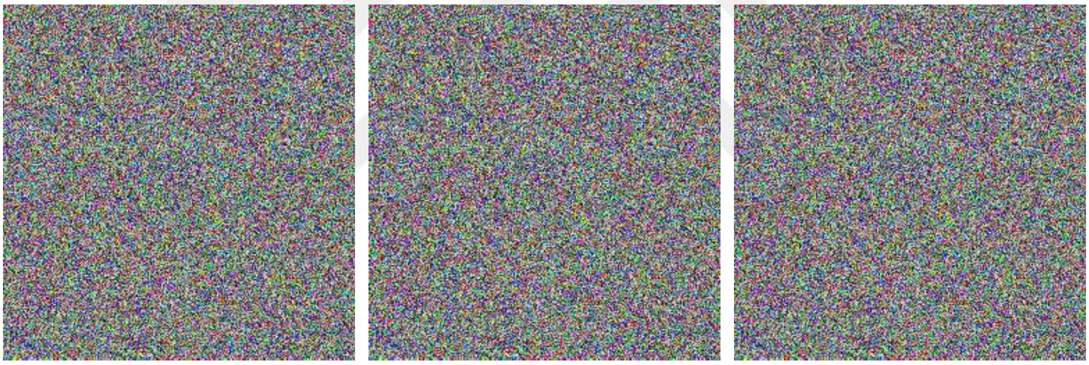
(a) (b) (c)



(d) (e) (f)

Figure 5.20 Salt and pepper noise results for "Mandril" image using BB84

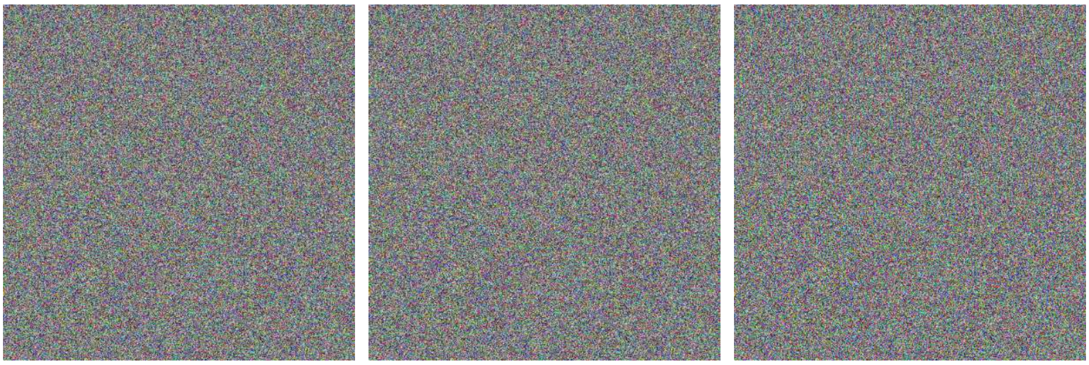
Secondly, gaussian noise with zero-mean and variance of (0.00001, 0.00003, 0.00005) is added to both the Lena image and the Mandril image as shown in figures 5.21 and 5.22.



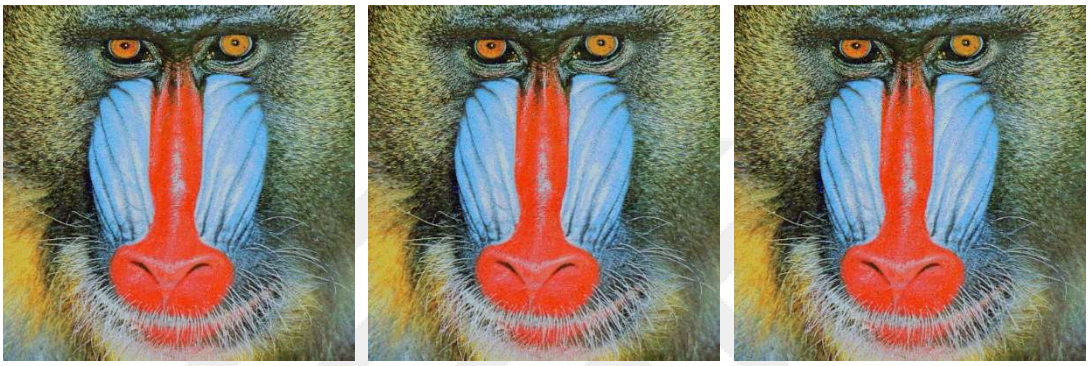
(a) (b) (c)

(d) (e) (f)

Figure 5.21 Gaussian noise results for "Lena" image using BB84



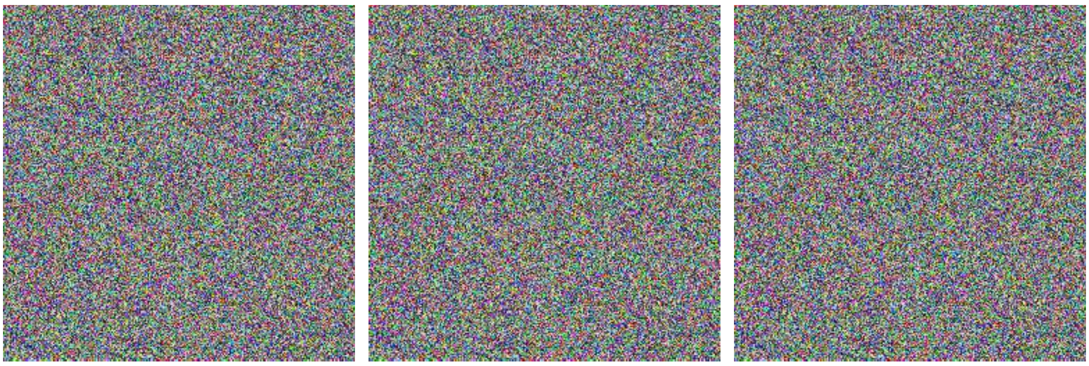
(a) (b) (c)



(d) (e) (f)

Figure 5.22 Gaussian noise results for "Mandril" image using BB84

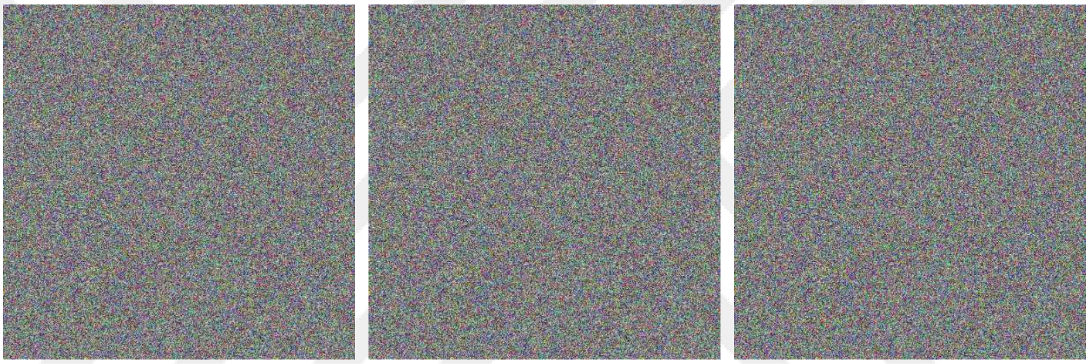
Finally, speckle noise with zero-mean and variance of (0.005, 0.0075 0.01) is added to the noisy BB84 encrypted images as shown in figures 5.23 and 5.24.



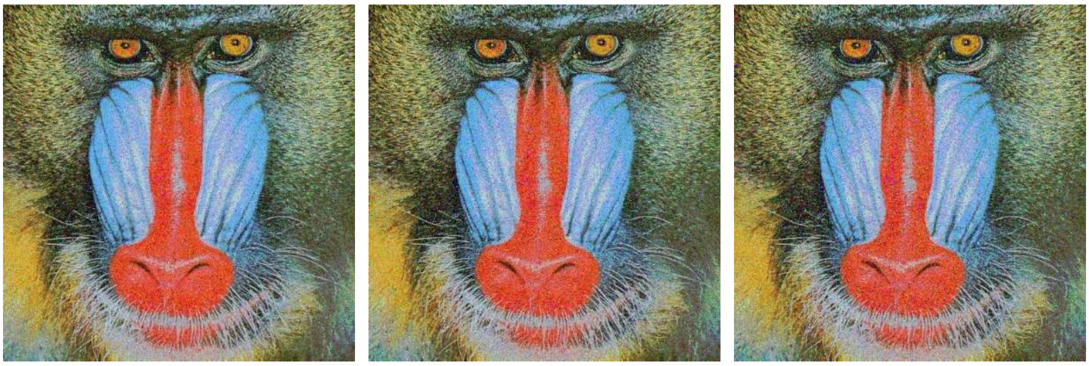
(a) (b) (c)

(d) (e) (f)

Figure 5.23 Speckle noise results for "Lena" image using BB84



(a) (b) (c)



(d) (e) (f)

Figure 5.24 Speckle noise results for "Mandril" image using BB84

The Peak-Noise-to-Signal Ratio (PSNR) is a quality measurement parameter that can assess the quality of the decrypted image [42,43]. PSNR is given by:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}} \tag{5.5}$$

where

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum_{x=1}^{M} \sum_{y=1}^{N} \left(\mathrm{O}(x,y) - \mathrm{D}(x,y)\right)^2 \tag{5.6}$$

Table 5.5 PSNR results for BB84

Image	Attack	Parameter	Red PSNR	Green PSNR	Blue PSNR
Lena	Salt & Pepper	0.05	20.9099	21.2463	22.7466
		0.075	19.2410	19.6597	20.8632
		0.1	17.9406	18.4647	19.5683
	Gaussian	0.00001	28.4366	27.3311	49.4921
		0.00003	25.9764	25.7577	45.0950
		0.00005	24.9103	24.7546	42.8856
	Speckle	0.005	17.7787	18.0272	28.1499
		0.0075	17.0347	17.3040	26.4453
		0.01	16.2776	16.6834	25.2862
Mandril	Salt & Pepper	0.05	21.6837	21.8333	21.2038
		0.075	20.0710	20.2076	19.3867
		0.1	18.8291	19.0407	18.1860
	Gaussian	0.00001	29.3443	27.9241	38.0688
		0.00003	26.7227	26.2128	36.5605
		0.00005	25.6096	25.3465	35.6746
	Speckle	0.005	18.4655	19.0080	22.8507
		0.0075	17.6327	18.2265	21.0647
		0.01	17.0384	17.6642	19.5789

In equations 5.5 and 5.6, 255 represents the maximum pixel value, MSE represents the mean square error function, M represents the number of rows, N represents the number of columns, $\mathrm{O}(x, y)$ and $\mathrm{D}(x, y)$ represent the original image and the decrypted image respectively. A large value of PSNR signifies a better image quality. Table 5.5 shows the PSNR results after polluting the BB84 encrypted images with different types of noise.

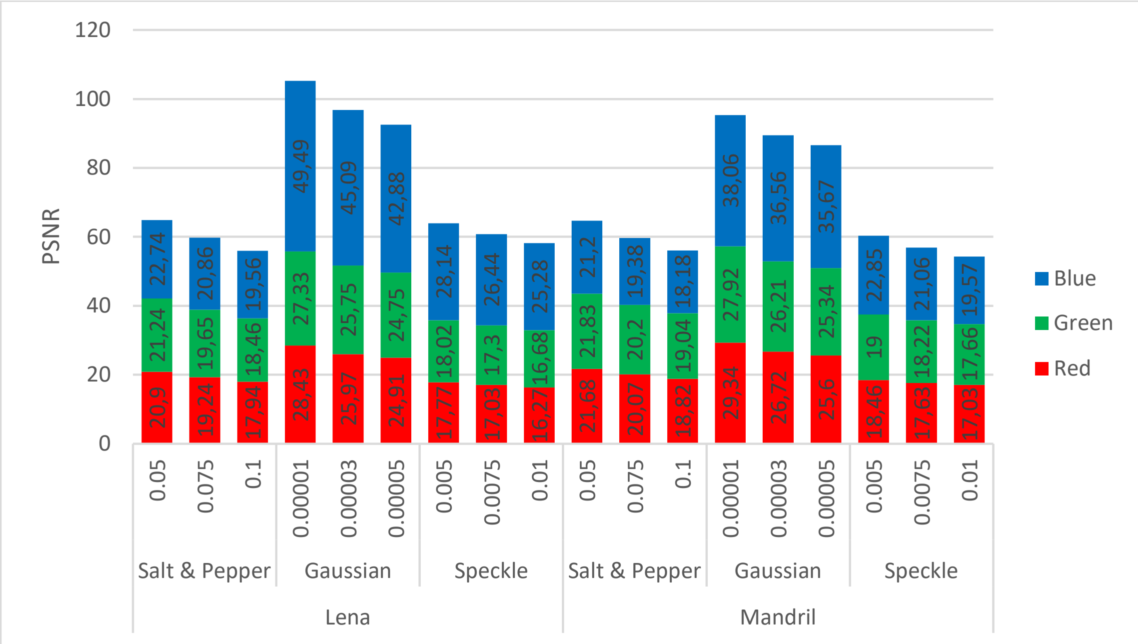


Figure 5.25 PSNR bar chart for BB84

Figures 5.25 shows a visual representation of the PSNR values in bar chart after polluting the BB84 encrypted images with noise.

Table 5.6 shows the PSNR results after polluting the E91 encrypted Lena and Mandril images with different types of noise.

Table 5.6 PSNR results for E91

Image	Attack	Parameter	Red PSNR	Green PSNR	Blue PSNR
Lena	Salt & Pepper	0.05	20.8115	21.1077	22.8832
		0.075	19.2646	19.8048	21.0057
		0.1	17.7711	18.4348	19.7085
	Gaussian	0.00001	28.4336	27.4803	49.4997
		0.00003	25.8263	25.4965	45.0804
		0.00005	24.8066	24.7946	42.9458
	Speckle	0.005	17.7921	17.9834	28.1314
		0.0075	16.9247	17.2658	26.4651
		0.01	16.2611	16.7121	25.2546
Mandril	Salt & Pepper	0.05	21.7666	21.7926	21.2451
		0.075	19.9675	20.1639	19.4413
		0.1	18.7428	19.0036	18.2281
	Gaussian	0.00001	29.2624	28.0203	38.2864
		0.00003	26.5412	26.3349	36.4973
		0.00005	25.5344	25.4241	36.4104
	Speckle	0.005	18.4427	19.0230	22.9559
		0.0075	17.6734	18.2055	20.9871
		0.01	17.0230	17.7223	19.5903

Figures 5.26 shows a visual representation of the PSNR values in bar chart after polluting the E91 encrypted images with noise.

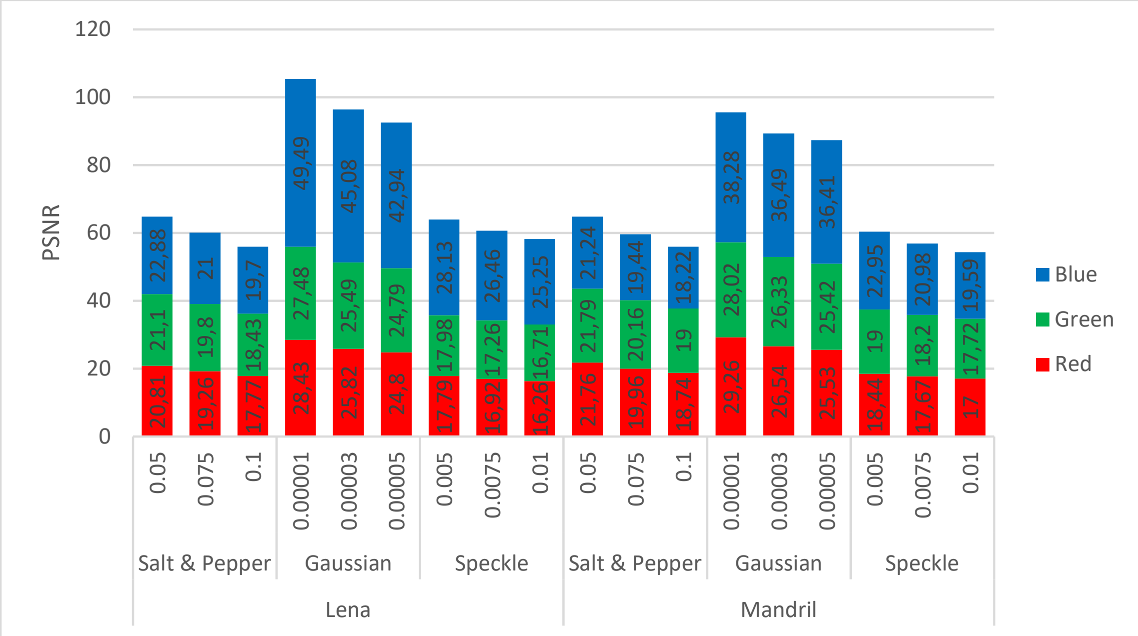


Figure 5.26 PSNR bar chart for E91

5.3.5 Structural Similarity Index

"Structural Similarity Index Method is a perception-based model. In this method, image degradation is considered as the change of perception in structural information. It also collaborates some other important perception-based fact such as luminance masking, contrast masking, etc. The term structural information emphasizes about the strongly inter-dependent pixels or spatially closed pixels. These strongly inter-dependent pixels refer some more important information about the visual objects in image domain. Luminance masking is a term where the distortion part of an image is less visible in the edges of an image. On the other hand, contrast masking is a term where distortions are also less visible in the texture of an image. SSIM estimates the perceived quality of images and videos. It measures the similarity between two images: the original and the recovered." [43]

In my study, the standard version of SSIM is used to assess the quality of a compressed image when it's decrypted in comparison with the original image. The results of this test for the BB84 protocol are shown on table 5.7.

Table 5.7 SSIM results for BB84

Image	SSIM
Lena	0.9875
Mandril	0.9909
Peppers	0.9753
Airplane	0.9870

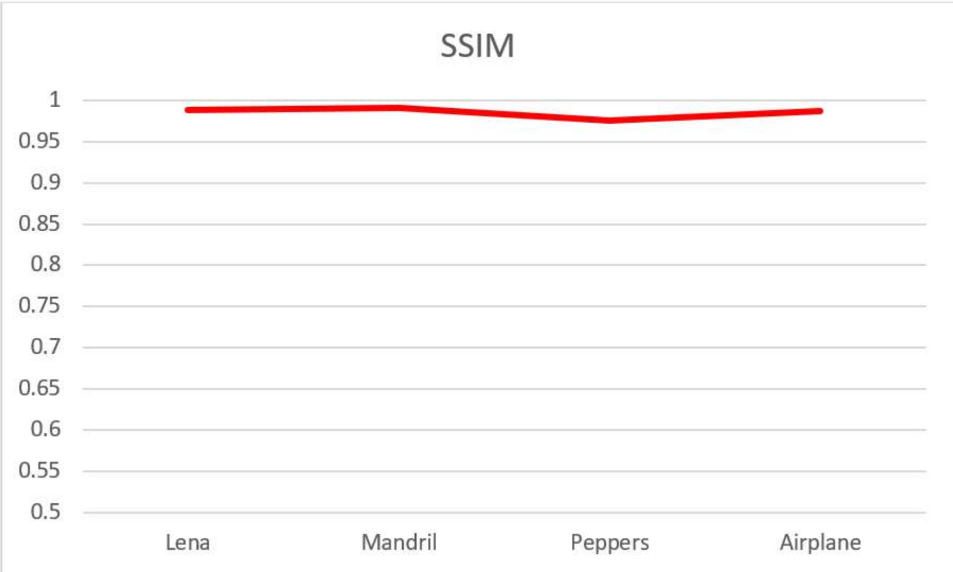


Figure 5.27 SSIM chart for BB84

Figures 5.27 and 5.28 show a visual representation of the SSIM values in a graph for both BB84 and E91 respectively.

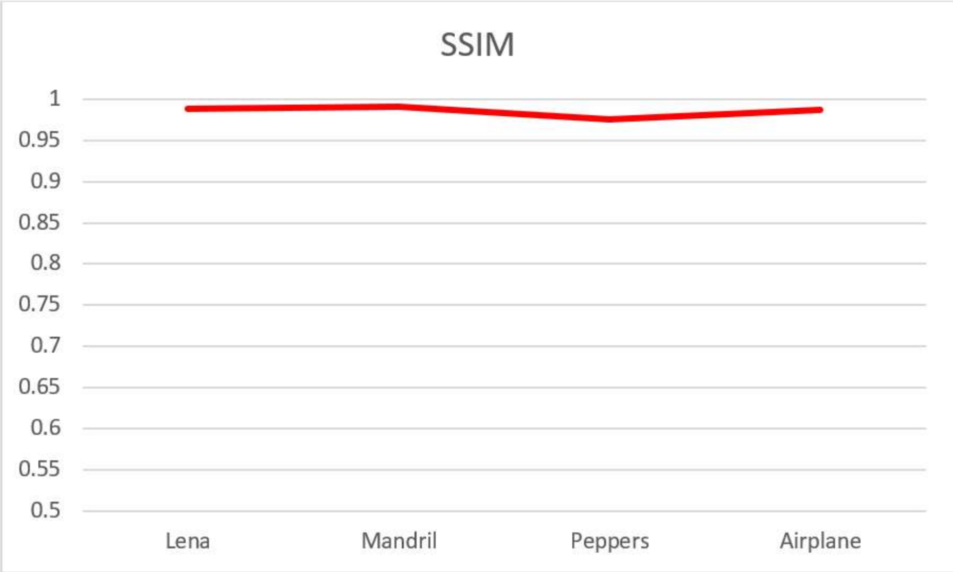


Figure 5.28 SSIM chart for E91

Table 5.8 shows he results of SSIM test for the E91 protocol.

Table 5.8 SSIM results for E91

Image	SSIM
Lena	0.9875
Mandril	0.9909
Peppers	0.9753
Airplane	0.9870

5.3.6 Known Plaintext Attacks (KPAs)

Given any piece of plaintext and its corresponding ciphertext, it is considered trivial to obtain the key used in the encryption process since the plaintext is combined with the key using XOR and modular arithmetic operations to obtain the ciphertext. Therefore, combining the plaintext with the corresponding ciphertext using the same operations would result in obtaining the key [44].

Let P represent the plaintext, C represent the ciphertext, and K represent the key. As shown in equation 5.7, it is considered simple to obtain the key.

$$P \oplus C = P \oplus (K \oplus P) = K \tag{5.7}$$

However, obtaining the key does not make this encryption method any less secure since the key is used only once to encrypt and decrypt an image, and it will later be discarded. Since the key is never reused, attacks that involve obtaining a pair of plaintext and ciphertext (e.g., CPA, CCA, KPA) are useless.

CHAPTER 6

CONCLUSION

In this study, the weaknesses of classical encryption methods were examined and the need to switch to QKD protocols was emphasized. One of the popular protocols, the BB84 QKD protocol, has been implemented using Python. The software operates on two modes; a mode where an eavesdropper isn't present, and a mode where an eavesdropper is trying to infiltrate the communication process. The generated key from this simulation has been used to encrypt digital RGB images using a proposed method. This encryption method consists of a combination of XOR and modular arithmetic operations in order to combine the secret key with the plain image. The encrypted image is then sent via classical channels to the receiver, who uses the same key to decrypt the image and obtain the original image.

Several standard attack checks and statistical tests have been made in order to ensure the strength and robustness of this encryption method. Based on the results, the following conclusions can be drawn:

- The encryption method is immune to statistical-based attacks since there is no correlation between adjacent pixels in the encrypted image, and the histograms obtained from using both protocols do not resemble those of the plain image.
- The method is also resistant against entropy-based attacks, as the pixel values are evenly distributed throughout the color spectrum.
- If noise was introduced during the transmission of the encrypted image, the method can recover the original image to some degree without significant loss to the quality.
- The disadvantage of the method is the key size. Since the key has to be the same size as the image, that would require a lot of storage just to store the key.
- Keys generated using BB84 resulted in a 21.6% better correlation result than that of E91.

- Key generated using both E91 and BB84 resulted in an entropy result of 7.99, which is very close to the ideal value 8.
- PSNR values obtained using E91 were 0.04% better than that of BB84.
- The SSIM test results are the same for both BB84 and E91.

Future work revolves around revolves around encrypting media that are larger than images, as well as implementing these protocols on actual quantum computers rather than using simulations.

REFERENCES

[1] Larew, K., & Kahn, D. The Codebreakers: The Story of Secret Writing. The American Historical Review, (*74*), (2) 1968

[2] Singh, S. The Code Book: The Science of Secrecy from Ancient Egypt To Quantum Cryptography. Anchor Books. EDICION, ISBN, 1999

[3] Franco, P. "Understanding bitcoin: Cryptography, engineering, and economics" 2014

[4] Qadir, A. M., & Varol, N. A review paper on cryptography, *7th International Symposium on Digital Forensics and Security, ISDFS 2019*, 2019

[5] Wolf, R. Introduction, *Lecture Notes in Physics*, 2021

[6] Limbong, T., & Silitonga, P. D. P. Testing the Classic Caesar Cipher Cryptography using of Matlab. International Journal of Engineering Research & Technology, (*6*), (2) 2017

[7] Senthil, K., Prasanthi, K., & Rajaram, R. A modern avatar of Julius Ceasar and Vigenere cipher, *2013 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*, 2013

[8] Mohammed, A.-A., & Olaniyan, A. Vigenere Cipher: Trends, Review and Possible Modifications. International Journal of Computer Applications, (*135*), (11) 2016

[9] Buchmann, J., Braun, J., Demirel, D., & Geihs, M. Quantum cryptography: A view from classical cryptography. Quantum Science and Technology, (*2*), (2) 2017

[10] Patil, P. A., & Boda, R. Analysis of Cryptography: Classical verses Quantum Cryptography. International Research Journal of Engineering and Technology, 2016

[11] Mavroeidis, V., Vishi, K., Zych, M. D., & Jøsang, A. The impact of quantum computing on present cryptography. International Journal of Advanced Computer Science and Applications, (*9*), (3) 2018

[12] Diamanti, E., Lo, H. K., Qi, B., & Yuan, Z. Practical challenges in quantum key distribution. Npj Quantum Information, (*2*), (1) 2016

[13] Pawar, H. R., & Harkut, D. G. Classical and Quantum Cryptography for Image Encryption Decryption, *Proceedings of the 2018 3rd IEEE International Conference on Research in Intelligent and Computing in Engineering, RICE 2018*, 2018

[14] Zhang, Q., Xu, F., Chen, Y.-A., Peng, C.-Z., & Pan, J.-W. Large scale quantum key distribution: challenges and solutions [Invited]. Optics Express, (*26*), (18) 2018

[15] Li, S., & Li, L. Round robin differential quadrature phase shift quantum key distribution by using odd coherent states. Optik, (*227*), 2021

[16] Nurhadi, A. I., & Syambas, N. R. Quantum Key Distribution (QKD) Protocols: A Survey, *Proceeding of 2018 4th International Conference on Wireless and Telematics, ICWT 2018*, 2018

[17] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. "Handbook of applied cryptography" 1996

[18] Sangeeta, & Kaur, Er. A. A Review on Symmetric Key Cryptography Algorithms. International Journal of Advanced Research in Computer Science, (*8*), (4) 2017

[19] Ubaidullah, M., & Makki, Q. A Review on Symmetric Key Encryption Techniques in Cryptography. International Journal of Computer Applications, (*147*), (10) 2016

[20] Kumar, Y., Munjal, R., & Sharma, H. Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures.

IJCSMS International Journal of Computer Science and Management Studies, 2011

[21] Annapoorna Shetty, S. S. and K. A Review on Asymmetric Cryptography – RSA and ElGamal Algorithm. International Journal of Innovative Research in Computer and Communication Engineering, (*2*), (Special issue 5) 2014

[22] Sheth, R. K. Analysis of Cryptography Techniques. INTERNATIONAL JOURNAL OF RESEARCH IN ADVANCE ENGINEERING, (*1*), (2) 2015

[23] Zhang, Q. An overview and analysis of hybrid encryption: The combination of symmetric encryption and asymmetric encryption, *Proceedings - 2021 2nd International Conference on Computing and Data Science, CDS 2021*, 2021

[24] Gupta, R. K. A Review Paper On Concepts Of Cryptography And Cryptographic Hash Function. European Journal of Molecular & Clinical Medicine , (*7*), (7) 2020

[25] Prashant P. Pittalia A Comparative Study of Hash Algorithms in Cryptography. International Journal of Computer Science and Mobile Computing, (*8*), (6) 2019

[26] R. M, S., C. V, Raghavendra., & Koddaddi, M. S. Quantum Computers: A Review on its Working Mechanism, Uses and Developments. International Journal of Advanced Research in Science, Communication and Technology, 2021

[27] Kjaergaard, M., Schwartz, M. E., Braumüller, J., Krantz, P., Wang, J. I. J., Gustavsson, S., et al. Superconducting Qubits: Current State of Play. Annual Review of Condensed Matter Physics, (*11*), 2020

[28] Wang, Y. Quantum computation and quantum information. Statistical Science, (*27*), (3) 2012

[29] Majot, A., & Yampolskiy, R. Global catastrophic risk and security implications of quantum computers. Futures, (*72*), 2014

[30] Bernstein, D. J., & Lange, T. Post-quantum cryptography. Nature, (*549*), (7671) 2017

[31] Sen, D. The uncertainty relations in quantum mechanics. Current Science, (*107*), (2) 2014

[32] Wootters, W. K., & Zurek, W. H. A single quantum cannot be cloned. Nature, (*299*), (5886) 1982

[33] Bennett, C. H., & Brassard, G. Quantum cryptography: Public key distribution and coin tossing. Theoretical Computer Science, (*560*), (P1) 2014

[34] Bennett, C. H., & Brassard, G. Experimental quantum cryptography: the dawn of a new era for quantum cryptography: the experimental prototype is working]. ACM SIGACT News, (*20*), (4) 1989

[35] Qiao, H., & Chen, X.-Y. Simulation of BB84 Quantum Key Distribution in depolarizing channel. Proceedings of 14th Youth Conference on Communication Simulation, 2009

[36] Brassard, G., & Salvail, L. Secret-key reconciliation by public discussion, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1994

[37] Bennett, C. H., Brassard, G., & Robert, J. M. Privacy amplification by public discussion. SIAM Journal on Computing, (*17*), (2) 1988

[38] Mahmoud, M. A., Mekkawy, T., & Elbayoumy, A. D. Enhanced image encryption algorithm by quantum key distribution. IOP Conference Series: Materials Science and Engineering, (*1172*), (1) 2021

[39] Bouquin, D. R. GitHub. Journal of the Medical Library Association, (*103*), (3) 2015

[40] Shannon, C. E. A Mathematical Theory of Communication. Bell System Technical Journal, (*27*), (3) 1948

[41] Hosny, K. M., Kamal, S. T., & Darwish, M. M. A color image encryption technique using block scrambling and chaos. Multimedia Tools and Applications, (*81*), (1) 2022

[42] Kaushik, P., & Sharma, Y. Comparison of different image enhancement techniques based upon Psnr & Mse. International Journal of Applied Engineering Research, (*7*), (11 SUPPL.) 2012

[43] Sara, U., Akter, M., & Uddin, M. S. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. Journal of Computer and Communications, (*07*), (03) 2019

[44] Stallings, W. "Cryptography and Network Security: Principles and Practices" 2005