1.1 Preface

The Network Information Security means protecting networks from unauthorized access, modification or destruction.

Data itself is the first level in security it critical for most businesses and even home computer users. Client information, payment information, personal files, bank account details - all of this information can be hard to replace and potentially dangerous if it falls into the wrong hands. Data lost due to natural causes or disasters such as flood or fire is crushing, but losing it to hackers or a malware infection can have much greater consequences.

The second level is the Access control; to ensure that the data was not altered during transmission (information integrity) and no one can read the data accept the intended receiver (Privacy and Confidentiality) also to ensure that the sender really sent this message (Non – repudiation).

Cryptography (traditionally) and the Steganography (Modern) are an example of Controlling Method used.

Cryptography is the science of writing in secret code and is an ancient art. There are, in general, several ways of classifying cryptographic algorithms. For our purposes here, it will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use.

The three types of algorithms based on the number of keys are:

- Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption

- Public Key Cryptography (PKC): Uses one key for encryption and another for decryption
- Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information

In all cases, the initial unencrypted data is referred to as plaintext. It is encrypted into cipher text, which will in turn (usually) be decrypted into usable plaintext.

Steganography is the process of hiding a message within an image, text file, audio file, video file and protocol.

Steganography is often mistaken for cryptography. Both are security measures taken to conceal messages from third parties. The difference between the two is in the techniques used to conceal the message. Cryptography only secures the message by scrambling it so it cannot be understood. While Steganography secures the message by hiding it so it cannot be seen. Hiding a message with Steganography methods reduces the chance of a message being detected. However, if that message is also encrypted, if discovered, it must also be cracked (yet another layer of protection).

The word Steganography is generated from the following two words whose origins come from ancient Rome and Greece.'Stegano' meaning covered and ‘graphy’ meaning writing. These two words are not focused on transformation of the words as in Cryptography; they are rather focused on the hiding of the message itself.

Classifications of Steganography techniques based on the types of cover files. Almost all digit file formats can be used for Steganography, however only those with a high degree of redundant bits are preferred. The larger size of audio and video files makes them less popular as compared to images.
One of the cover files is the digital image; Image Steganography has come quite far in recent years.

An image in a computer is an array of numbers that represent light intensities at various points (pixels). These pixels make up the image's raster data. Digital images are stored in either 24-bit (true color images) or 8-bit per pixel for grey-scale image or 1bit per pixel for binary image. Grey-scale images are preferred because the shades are changed very gradually between palette entries. This increases the image's ability to hide information.

The most well known techniques for data hiding in images are:

- Spatial Domain
- Transformation Domain
- Compression Techniques.

Both of Transformation Domain and Compression Techniques have higher level of robustness against simple Statistical analysis inverse of the Spatial Domain which will be discussed later.

1.2 *Problem statement*

The information security using cryptography algorithm with simple permutation and substitution or using image spatial domain algorithm with visible impact are vulnerable since any hacker can decrypt any intercepted data.

1.3 *Objective*

- To maintain the information security (privacy, confidentiality and accuracy of the data)
- To compare between the different spatial techniques
- To determine the Relation between data size and image size.
- To implementing a system which is able of transmitting and receiving data in an image form.

1.4 *Methodology*

Rivest, Adi Shamir, and Leonard Adleman (RSA) algorithm will be used for data encryption while a Digital image will represent the Steganography media where Least Significant Bit algorithm (LSB), Random approach algorithm, Convolution algorithm or bit XOR algorithm, Random approach based LSB algorithm and Random approach based bit XOR algorithm will be use as techniques for embedding data inside image.

To measure the quality of the Steganography image (Stego image) the Mean Square error (MSE), Peak Signal

to Noise Ratio (PSNR), Mean and Standard deviation will be calculate.

Steganography system will be implemented by graphical interface using the **MAT**rix **LAB**oratory (MATLAB) language.

1.5 *Thesis out line*

This thesis consists of five chapters.

Chapter one:

Include introduction to information security.

Chapter Two:

Present the Historical Background of the information security.

Chapter three:

Elaborate the Basic Concept of information security Techniques.

Chapter four:

Contain the simulation model and the Result discussion.

Chapter five:

The thesis Conclusion and future work.

This chapter discusses the information security background in ancient civilizations. Where there are many aspects to the information security and two of essential aspect are Cryptography and Steganography.

2.1 *Cryptography*:

Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. When an Egyptian scribe used non-standard hieroglyphs in an inscription and when the Roman kings used secret code when send dispatches. Julius Caesar is one of the Roman kings he enciphered his dispatches by writing D for A, E for B and so on [2]. When Augustus Caesar ascended the throne, he changed the imperial cipher system so that C was now written for A, D for B, and so on .But all Caesar substitution cipher consider very week because if the amount of

displacement is known there is no secret. Even if the displacement is not known it can be discovered very easily because the number of the possible cipher solution is only 25.

The Arabs generalized this idea to the monoalphabetic substitution, in which a keyword is used to permute the cipher alphabet. They were writing the plaintext in lowercase letters and the cipher text in uppercase, as shown in Figure2.1.

Figure2.1: Monoalphabetic substitution cipher.

But breaking ciphers of this kind is a straightforward because at any language there are some letters that occur more often than other. Artificial intelligence researchers have shown some interest in writing programs to solve monoalphabetic substitutions; using letter and digraph (letter- pair) frequencies alone. They typically succeed with about 600 letters of cipher text, while-smarter strategies, such as guessing probable words, can cut this to about 150 letters. A human cryptanalyst will usually require much less.

There are basically two ways to make a stronger cipher: the ***stream cipher*** and the ***block cipher***. In the former, the encryption rule depend on a plaintext symbol's position in the stream of plaintext symbols, while in the latter you encrypt several plaintext symbols at once in a block. Let's look at early examples.

2.1.1 *An Early Stream Cipher: The Vigenere*

An early stream cipher is commonly ascribed to the Frenchman Blaise de Vigenere, a diplomat who served King Charles IX. It works by adding a key repeatedly into the plaintext using the convention that A = 0, B = 1, . . . , Z = 25; and addition is carried out modulo 26—that is, if the result is greater than 25, we subtract as many multiples of 26 as are needed to bring us into the range [0, . . . ,25], that is, [A, . . . ,Z]. Mathematicians write this as:

$$C = P + K \bmod 26$$

..... 2.1

For example, when we add P (15) to U (20) we get 35, which we reduce to 9 by subtracting 26; 9 correspond to J, so the encryption of P under the key U (and of U under the key P) is J.

Vigenere encrypt the message by uses a key long as the message, usually the key is a repeating keyword for

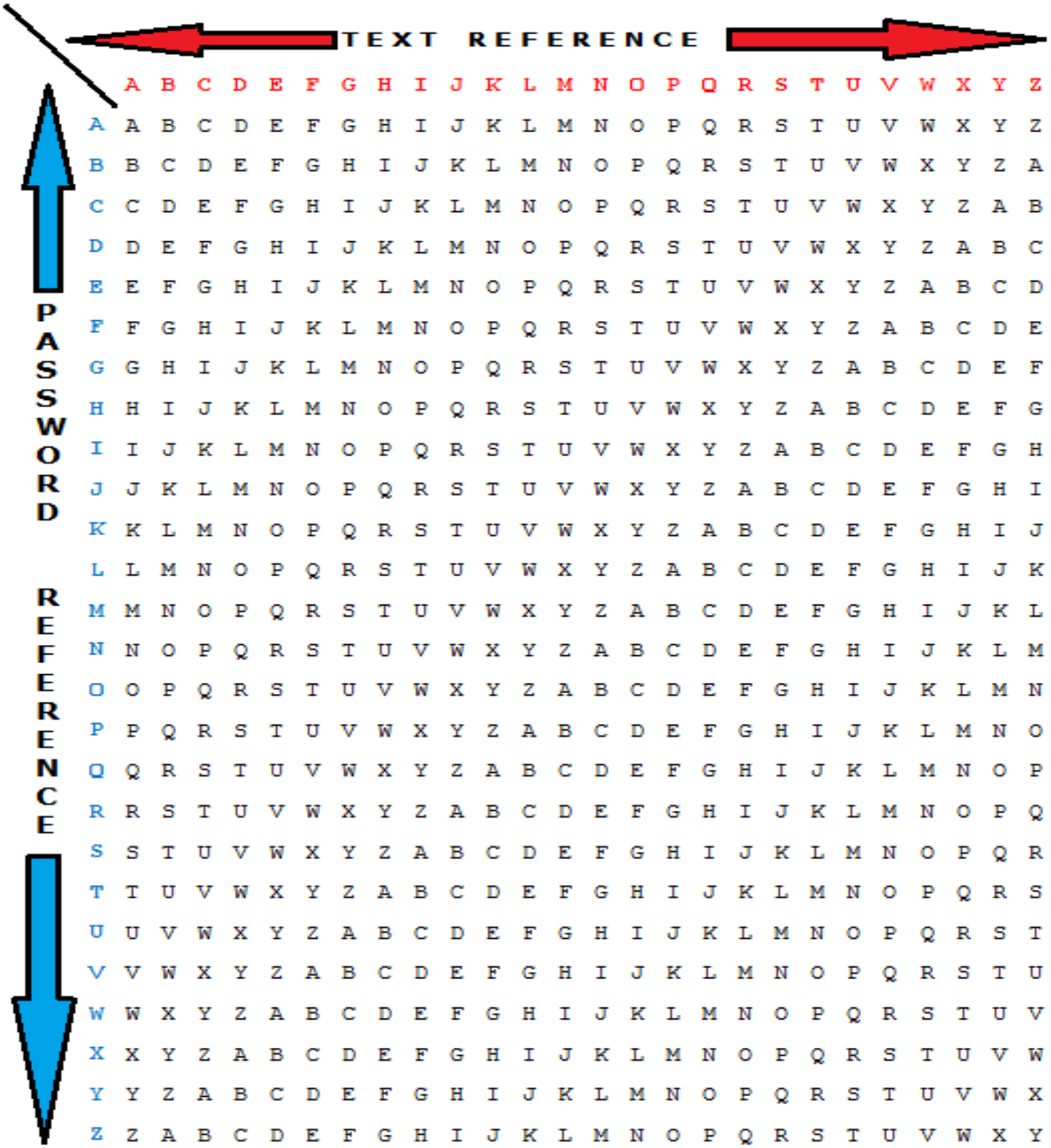
example, if the keyword is run, the message “to be or not to be that is the question “is encrypt as in figure*2.2*

Plain:	tobeornottobethatisthequestion
Key:	runrunrunrunrunrunrunrunrunrun
Cipher:	KIOVIEEIGKIOVNURNVJNUVKHVMGZIA

Figure 2.2: A Vigenere polyalphabetic substitution cipher.

To decrypt the message Vigenere formed a square, the vigenere table as in table 2.1, consisting of 26 horizontal alphabets, one below the other, with each shifted to the right by one letter. Vertical alphabet was used to define the key and, at the top, an additional alphabet was used for the plaintext letters. The key letter again identifies the row and the position of the cipher text letter in that row determines the column, and the plaintext letter is at the top of that column.

Table2.1: vigenere Table



2.1.2 *The One-Time Pad*

One way to make a stream cipher of this type proof against attacks is for the key sequence to be as long as the plaintext, and to never repeat. This was proposed by Gilbert Vernam during World War I [4]; its effect is that given any cipher text and any plaintext of the same length, there is a key that decrypts the cipher text to the plaintext. Regardless of the amount of computation that opponents can do, they are none the wiser, as all possible plaintexts are just as likely. This system is known as the ***one-time pad***. Leo Marks' engaging book on cryptography in the Special Operations Executive in World War II [5] relates how one-time key material was printed on silk, which agents could conceal inside their clothing; whenever a key had been used, it was torn off and burned. For example suppose you had intercepted a message from a wartime German agent, which you knew started with "Heil Hitler," and that the first 10 letters of cipher text were DGTYI BWPJA. This means that the first 10 letters of the onetime pad were wclnb tdefj, as shown in Figure2.3

Once he had burned the piece of silk with his key material, the spy could claim that he was actually a member of the anti-Nazi underground resistance, and that the message actually said "Hang Hitler." This is quite possible, as the key material could just as easily have been wggsb tdefj, as shown in Figure2.4

Now, we rarely get anything for nothing in cryptology, and the price of the perfect secrecy of the one-time pad is that it fails completely to protect message integrity.

Suppose that you wanted to get this spy into trouble; you could change the cipher text to DCYTI BWPJA, as shown in Figure2.5

During the World War II, Claude Shannon proved that a cipher has perfect secrecy if and only if there are as many possible keys as possible plaintexts, and if every key is equally likely; therefore, the one-time pad is the only kind of system that offers perfect secrecy [6, 9].

The one-time pad is still used for high-level diplomatic and intelligence traffic, but it consumes as much key material as there is traffic, hence is too expensive for most applications.

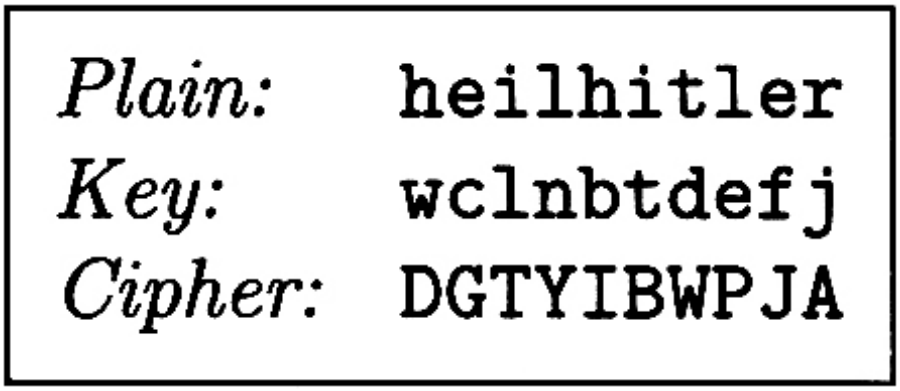


Figure2.3: A spy’s message.

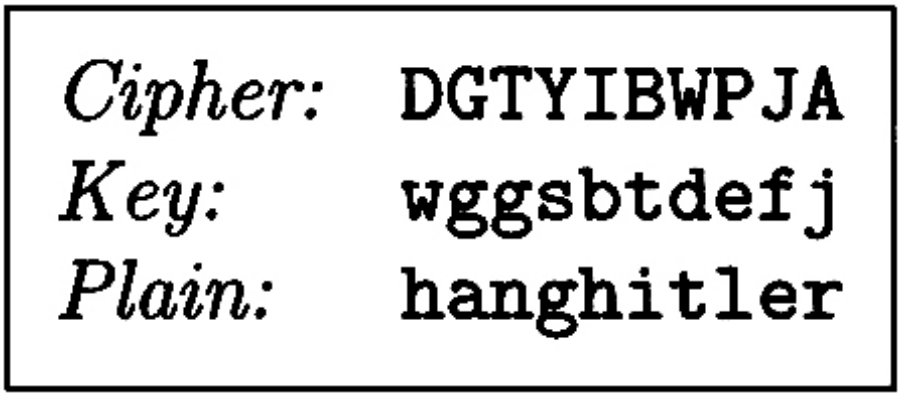


Figure2.4: What the spy claimed he said.

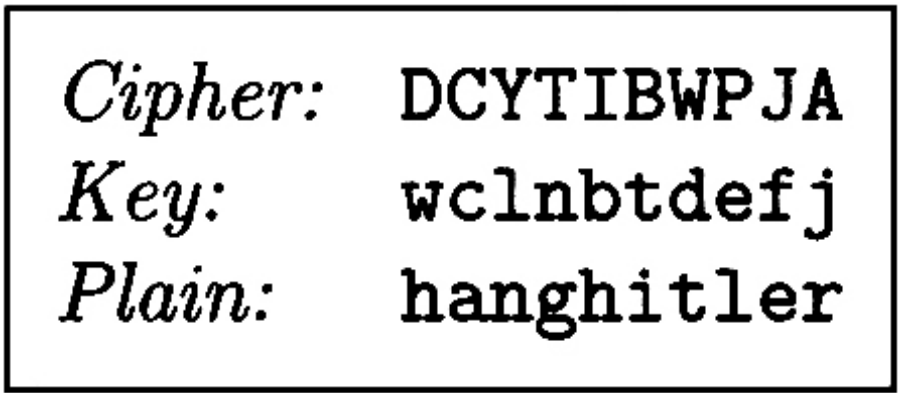


Figure2.5: Manipulating the message in Figure2.4to entrap the spy.

2.1.3 *An Early Block Cipher: Playfair*

One of the best-known early block ciphers is the Playfair system. It was invented in 1854 by Sir Charles Wheatstone, a telegraph pioneer. This cipher uses a 5 by 5 grid, in which the alphabet is placed, permuted by the keyword, and

omitting the letter J (see table2.2).The plaintext is first conditioned by replacing J with I wherever it occurs, then dividing it into letter pairs, preventing double letters occurring in a pair by separating them with an x, and finally adding a z if necessary to complete the last letter pair. The Example of Playfair wrote on napkin was “Lord Granville’s letter,” which becomes “lo rd gr an vi lx le sl et te rz”.

It is then enciphered two letters at a time using the following rules:

- If two letters are in the same row or column, they are replaced by the succeeding letters. For example, “am” enciphers to “LE.”
- Otherwise, the two letters stand at two of the corners of a rectangle in the table, and we replace them with the letters at the other two corners of this rectangle. For example, “lo” enciphers to “MT.”

Table2.2: The Playfair enciphering tableau.

P	A	L	M	E
R	S	T	O	N
B	C	D	F	G
H	I	K	Q	U
V	W	X	Y	Z

Plain:	lo rd gr an vi lx le sl et te rz
Cipher:	MT TB BN ES WH TL MP TA LN NL NV

Figure2.6: Example of Playfair enciphering.

Variants of this cipher were used by the British army as a field cipher in World War I, and by the Americans and Germans in World War II. It's a substantial improvement on Vigenere, as the statistics an analyst can collect are of *digraphs* (letter pairs) rather than single letters, so the distribution is much flatter, and more cipher text is needed for an attack.

Again, it's not enough for the output of a block cipher to just look intuitively "random."

Playfair cipher texts do look random, but they have the property that if you change a single letter of a plaintext pair, then often only a single letter of the cipher text will change. Thus, using the key in table2.2, rd enciphers to TB while rf enciphers to OB and rg enciphers to NB. One consequence is that, given enough cipher text or a few probable words, the table (or an equivalent one) can be reconstructed [3]. We will want the effects of small changes in a block cipher's input to diffuse completely through its output: changing one input bit should, on average, cause half of the output bits to change.

The security of a block cipher can be greatly improved by choosing a longer block length than two characters. For example, the *Data Encryption Standard* (DES), which is widely used in banking, has a block length of 64 bits, which equates to eight ASCII characters and the Advanced Encryption Standard (AES), which is replacing it in many applications, has a block length of twice this.

2.1.4 *One-Way Function*

The third classical type of cipher is the one-way function. This to protect the integrity and authenticity of messages, which as we've seen is not protected at all by many simple ciphers, where it is often easy to manipulate the cipher text in such a way as to cause a predictable change in the plaintext.

After the invention of the telegraph in the mid-nineteenth century, banks rapidly became its main users, and

developed systems for transferring money electronically. Of course, it isn't the money itself that is "wired," but a payment instruction, such as:

To Lombard Bank, London. Please pay from our account with you no. 1234567890 the sum of £1000 to John Smith of 456 Chesterton Road, who has an account with HSBC Bank Cambridge no. 301234 4567890123, and notify him that this was for "wedding present from Doreen Smith." From First Cowboy Bank of Santa Barbara, CA, USA. Charges to be paid by us.

Since telegraph messages were relayed from one office to another by human operators, it was possible for an operator to manipulate a payment message. Banks, telegraph companies and shipping companies developed ***code books***, which not only could protect transactions, but also shorten them—which were very important given the costs of international telegrams at the time. A code book was essentially a block cipher that mapped words or phrases to fixed-length groups of letters or numbers.

Thus, "Please pay from our account with you no" might become "AFVCT." A competing technology was *rotor machines*, mechanical cipher devices that produce a very long sequence of pseudorandom numbers, and combine them with plaintext to get Cipher text; these were independently invented by a number of people, many of whom dreamed of making a fortune selling them to the

banking industry. Banks weren't in general interested, but rotor machines became the main high-level ciphers used by the combatants in World War II.

The banks realized that neither mechanical stream ciphers nor code books protected message authenticity. If, for example, the codeword for 1000 is mauve and for 1,000,000 is magenta, then the crooked telegraph clerk who can compare the coded traffic with known transactions should be able to figure this out and substitute one for the other.

The critical innovation was to use a code book, but make the coding one-way by adding the code groups together into a number called a *test key.* (Modern cryptographers would describe it as a *hash value* or *message authentication code*) .Here is a simple example. Suppose that the bank has a code book with a table of numbers corresponding to payment amounts, as in table 2.3 In order to authenticate a transaction for $376,514, we add 53 (no millions), 54 (300,000), 29 (70,000) and 71 (6,000). (It's common to ignore the less significant digits of the amount.) This gives us a test key of 207.

Most real systems were more complex than this; they usually had tables for currency codes, dates, and even recipient account numbers. In the better systems, the code groups were four digits long rather than two; and to make it harder for an attacker to reconstruct the tables, the test keys were compressed: a key of 7549 might become 23 by adding the

first and second digits, and the third and fourth digits, and ignoring the carry.

Test keys are not strong by the standards of modern cryptography. Given somewhere between a few dozen and a few hundred tested messages, depending on the design details, a patient analyst could reconstruct enough of the tables to forge a transaction.

With a few carefully chosen messages inserted into the banking system by an accomplice, it's even easier still. But the banks got away with it: test keys worked fine from the late nineteenth century through the 1980s. In several years working as a bank security consultant, and listening to elderly bank auditors' tales over lunch, I only heard of two cases of fraud that exploited it: one external attempt involving cryptanalysis, which failed because the attacker didn't understand bank procedures, and one successful but small fraud involving a crooked staff member.

For now, test keys are the classic example of a one-way function used for authentication.

Table2.3: A simple test key system.

4	5	6	7	8	9
9	93	71	35	06	58
1	82	00	29	64	57

2.1.5 *Asymmetric Primitives*

Finally, some modern cryptosystems are asymmetric, in that different keys are used for encryption and decryption. For example, I publish

on my Web page a *public key* with which people can encrypt messages to send to me; I can then decrypt them using the corresponding *private key.*

There are some precomputer examples of this too; perhaps the best is the postal service. Another asymmetric application of cryptography is the *digital signature.* The idea here is that I can sign a message using a *private signature key*, and then anybody can check this using my *public signature verification key.* Again, there are precomputer analogues in the form of manuscript signatures and seals.

2.1.6 *modern research in the cryptography algorithm:*

In may-2012 Dr. Khaled Hamed Bilal and Obada Anwer Ahmed Osman presented a dissertation in the RSA algorithm and Data Encryption Standard algorithm (DES) which it was aimed in first place in implement a system for data encryption and decryption using RSA algorithm. Also in july-2009 Dr. Khaled Hamed Bilal and Hassan Harith Sharief presented a system for data encryption and decryption using DES algorithm.

2.2 *Steganography*:

Steganography is the process of hiding a message within an image, text file, audio file, video file or ect. Fundamentally, the Steganography goal is not to hinder the adversary from

decoding a hidden message, but to prevent an adversary from suspecting the existence of covert communications [3].

The word Steganography is generated from the following two words whose origins come from ancient Rome and Greece.'Stegano' meaning covered and 'graphy' meaning writing. These two words are not focused on transformation of the words as in Cryptography; they are rather focused on the hiding of the message itself. The early instances of Steganography mentioned in Herodotus' Histories [5]. Which clearly show that the hidden message has not been encrypted in any form, but that the sender solely relied on hiding the message so that the probability of the message being intercepted by a third party was diminished. A few of the early Steganography techniques include:

2.2.1 *Shaving a Slave's Head*

Herodotus mentions how the head of slave's had been shaved and then tattooed. The tattoo was a message warning of Persian invasions. Once the slave's hair had grown back the message had been concealed and the slave had been sent to deliver the message. Once the slave had reached the recipient their head had been shaved again, and the hidden message became visible.

2.2.2 *Modifying Ancient Tablets*

The old wooden writing tablets were used to hide messages. The tablets were covered with wax and then someone could scribe on the wax. Using this approach, messages would be visible to anyone who retrieved the tablets. A Greek devised a plan to remove the wax, write on the wood tablet itself and then cover the message with wax. If the tablets happened to meet intercepted during transit between the sender and recipient they would appear to be clean unused tablets, pass inspection and continue on to the recipient.

In more recent times, Steganography played a major role during World War II. Here are a few examples of how Steganography was used by governments and militaries to assist them during war.

2.2.3 *Invisible Inks*

During World War II hidden messages were embedded in normal appearing messages. A typical message would be typed using double space lines. On the white space of the message a hidden message would be written using fluids which dried clear. When warmed up these fluids would darken so that the hidden message was visible.

2.2.4 *Microdots in Microfiche :*

As message detection improved, new technologies were developed which could pass more information and be even

less conspicuous. The first microdots were discovered masquerading as a period on a typed envelope carried by a German agent in 1941. The message was not hidden, nor encrypted. It was just so small as to not draw attention to itself. Besides being so small, microdots permitted the transmission of large amounts of data including drawings and photographs.

With many methods being discovered and intercepted, the Office of Censorship took extreme actions such as banning flower deliveries which contained delivery dates, crossword puzzles and even report cards as they can all contain secret messages. Censors even went as far as rewording letters and replacing stamps on envelopes.

2.2.5 *Null Ciphers:*

The sender and recipient of the hidden message are aware of the logic being applied to the hidden message. Character shifting, substitution and algorithms are used to hide the message within a normal appearing block of text. One of the most notable null ciphers intercepted during World War II from a German spy is shown below .The real message is "camouflaged" in an innocent sounding message. Due to the "sound" of many open coded messages, the suspect communications were detected by mail filters. However

"innocent" messages were allowed to flow through. An example of a message containing such a null cipher is:

Fishing freshwater bends and saltwater
Coasts reward anyone feeling stressed.
Resourceful anglers usually find masterful
Leapers fun and admit swordfish rank
Overwhelming any day.

By taking the third letter in each word, the following message emerges:

Send Lawyers, Guns, and Money.

Another example is the following message which was actually sent by a German Spy in WWII is:

Apparently neutral's protest is thoroughly discounted
and ignored. Isman hard hit. Blockade issue affects
pretext for embargo on by products, ejecting suets and
vegetable oils.

Taking the second letter in each word the following message emerges:

Pershing sails from NY June 1.

There are rumors that during the 1980's Margareth Thatcher, then Prime Minister in UK, became so irritated about press leaks of cabinet documents, that she had the word processors programmed to encode the identity of the writer

in the word spacing, thus being able to trace the disloyal ministers.

Even the layout of a document can provide information about that document. Brassil et al authored a series of publications dealing with document identification and marking by modulating the position of lines and words [Brassil-Infocom94, Brassil- Infocom94, and Brassil-CISS95]. Similar techniques can also be used to provide some other "covert" information just as 0 and 1 are informational bits for a computer. As in one of their examples, word-shifting can be used to help identify an original document [Brassil-CISS95]. Though not applied as discussed in the series by Brassil et al, a similar method can be applied to display an entirely different message. Take the following sentence (S0):

We explore new steganographic and cryptographic

Algorithms and techniques throughout the world to

Produce wide variety and security in the electronic web

Called the Internet.

and apply some word shifting algorithm (this is sentence S1).

We explore new steganographic and cryptographic

Algorithms and techniques throughout the world to

Produce wide variety and security in the electronic web

Called the Internet.

By overlapping S0 and S1, the following sentence is the result:

We explore new steganographic and cryptographic

Algorithms and techniques throughout the world to

Produce wide variety and security in the electronic web

Called the Internet.

This is achieved by expanding the space before explore, the, wide, and web by one point and condensing the space after explore, world, wide and web by one point in sentence S1. Independently, the sentences containing the shifted words appear harmless, but combining this with the original sentence produces a different message: explore the World Wide Web.

2.2.6 *Principles of Steganography*

There are three categories of Steganography; Pure Steganography, Secret key Steganography, Public key Steganography.

Pure Steganography requires no prior exchange of the information between the two parties communicating and relies on secret through obscurity. This means that the algorithms not publicly known, and therefore the level of testing is also unknown, making the tool unproven. One has to go on faith alone in those involved in the tool's creation to be assured covert communication. Numerous instances of the false sense of security through obscurity can be cited [7].

Secret key Steganography usual uses a publicly known algorithm, and relies on a secret key chosen beforehand by the two parties communicating. This key is needed to both embed and extract the hidden information, and if the proper key is not used, it cannot be known if data is actually hidden in a given cover object [8]. If prior secure or, if desired, covert communications cannot be conducted to share the secret key before covert communications, another possibility is public key Steganography. It entails the sender using the recipient's public key to embed the information, which can only be detected using the recipient's private key. This is analogous to how the public key infrastructure works in cryptography. The interesting characteristic with public key Steganography is that even the sender should not be able to detect the secret message in the resulting stego object. As another alternative, proposes a steganographic key exchange protocol, where the communicating parties

exchange a sequence of messages that look like normal communications, and at the end of the sequence each party is able to compute a shared key. This shred key can then be used for secret key Steganography. No matter how it carried out, Steganography is not useful if the existence of secret information can be proven by outside parties.[7-2]. Steganalysis is the method by which to detect the presence of a hidden message and attempt to reveal the true contents of this message. This technology has also substantially evolved throughout history [2].

2.2.7 *Applications of Steganography*

Steganography can be used for wide range of applications such as, in defense organizations for safe circulation of secret data, in military and intelligence agencies, in smart identity cards where personal details are embedded in the photograph itself for copyright control of materials. In medical imaging, patient’s details are embedded within image providing protection of information and reducing transmission time and cost1, in online voting system so as to make the online election secure and robust against a variety of fraudulent behaviours2, for data hiding in countries where cryptography is prohibited, in improving mobile banking security3, in tamper proofing so as to prevent or detect unauthorized modifications and other numerous applications.

2.2.8 *modern research in the Steganography algorithm:*

Younes[10], et al. proposed a method in which data is inserted into Least Significant Bit (LSB) of each byte within the cover-image in encrypted form. Mandal[11] proposed a method with minimum deviation of image fidelity resulting high quality stego-image with better embedding capacity. Daneshkhah[18], et al. proposed convolution decoder-based data hiding method. Results show that embedding capacity can increase up to two bits per pixel for this method. Each time 4(LSB) of a pixel enter the decoder machine. Three XOR operations create three outputs n1, n2, and n3. Suppose n2, n3 as the hidden message. If n2, n3 be the same as hidden information, then there is no need to manipulate the original image; if not then change the original image in a way to cause the output of the decoder to be equal to the hidden message.

Nirmalya Chowdhury and Puspita Manna proposed algorithm based on taking pixel information from the cover image and forming a matrix, each of size 5×5. In each matrix, 4 bits from the secret message can be embedded.

Today, Steganography is researched both for legal and illegal reasons. Among the first ones there is war telecommunications, which use spread spectrum or meteor

scatter radio in order to conceal both the message and its source.

In the industry market, with the advent of digital communications and storage, one of the most important issues is copyright enforcement, so digital watermarking techniques are being developed to restrict the use of copyrighted data.

Another important use is to embed data about medical images, so that there are no problems with matching patient's records and images.

Among illegal ones is the practice of hiding strongly-encrypted data to avoid controls by cryptography export laws.

With every discovery of a message hidden using an existing application, a new steganographic application is being devised. There are even new twists to old methods. Drawings have often been used to conceal or reveal information. It is simple to encode a message by varying lines, colors or other elements in pictures. As we will see later.

During this period time when the Internet provides essential communication between tens of millions of people and is being increasingly used as a tool for commerce, security becomes a very important issue to deal with. This chapter discusses the methods or the techniques that we will use to protect the information.

3.1 *General Concepts*

3.1.1 *Encryption*:

The process of scrambles the message using cipher algorithm so it cannot be understood.

3.1.2 *Encryption key:*

A Sequence of value used with a cipher algorithm to encrypt a message. The choice of random (or cryptographically pseudorandom) keys, a secure key exchange mechanism, frequent key refreshments, and good secrecy protection of keys are all essential ingredients for the security of the integrity verification mechanism.

3.1.3 *Steganography:*

The Process of hiding message in digital image using Steganography algorithm so it cannot be seen

3.1.4 *Steganography algorithm:*

A technical technique use for embedded the message in to cover media (image).

3.1.5 *Decryption:*

The process of making an encrypted message recognizable using a decipher algorithm

3.1.6 *Extraction:*

The process of extract the message from cover media using the inverse of the embedded algorithm

3.2 *Types of Cryptographic Algorithms:*

There are several ways of classifying cryptographic algorithms. They will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The type of algorithms that we will use is

- Public Key Cryptography (PKC): Uses one key for encryption and another for decryption

3.2.1 *Public-key cryptography algorithm:*

Public-key cryptography has been said to be the most significant new development in cryptography in the last 300-400 years. PKC depends upon the existence of so-called *one-way functions*, or mathematical functions that are easy to computer whereas their inverse function is relatively difficult to compute, *Multiplication vs. factorization and Exponentiation vs. logarithms* represent two of the functional pairs that are used with PKC.

Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key. One key is used to encrypt the plaintext and the other key is used to decrypt the cipher text. Because pair of keys is required, this approach is also called ***asymmetric cryptography*.** One of the keys is designated the *public key* and may be advertised as widely as the owner wants. The other key is designated the *private key* and is never revealed to another party.

An example of the Public-key cryptography algorithms that we use is:

3.2.1.1 Rivest, Adi Shamir, and Leonard Adleman algorithm *(RSA):*

The first, and still most common, PKC implementation, named for the three MIT mathematicians who developed it — Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, *n*, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an *n* with roughly twice as many digits as the prime factors. The public key information includes *n* and a derivative of one of the factors of *n*; an

attacker cannot determine the prime factors of *n* (and, therefore, the private key) from this information alone and that is what makes the RSA algorithm so secure. To create an RSA public/private key pair, here are the basic steps:

1. Choose two distinct prime numbers p and q. (For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test).
2. Compute n

$$n = p*q \quad \text{..... 3.1}$$

(n is used as the modulus for both the public and private keys)

3. Compute φ (n)

$$\varphi(n) = (p-1)*(q-1) \quad \text{..... 3.2}$$

Where

Φ is Euler's totient function.

4. Choose an integer e such that 1 < e < φ (n) and greatest common divisor of (e, φ (n)) = 1; i.e., e and φ (n) are co prime.
5. Determine d as:

$$d \equiv e^{-1} \pmod{\varphi(n)}$$ 34

.....

..... 3.3

i.e., d is the multiplicative inverse of e mod φ (n).

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d which must be kept secret. (P, q, and φ (n) must also be kept secret because they can be used to calculate d.).To encrypt a message, M, with the public key, creates the cipher text, C, using the equation:

$$C = M^e \bmod n$$

3.4.....

The receiver then decrypts the cipher text with the private key using the equation:

$$M = C^d \bmod n$$

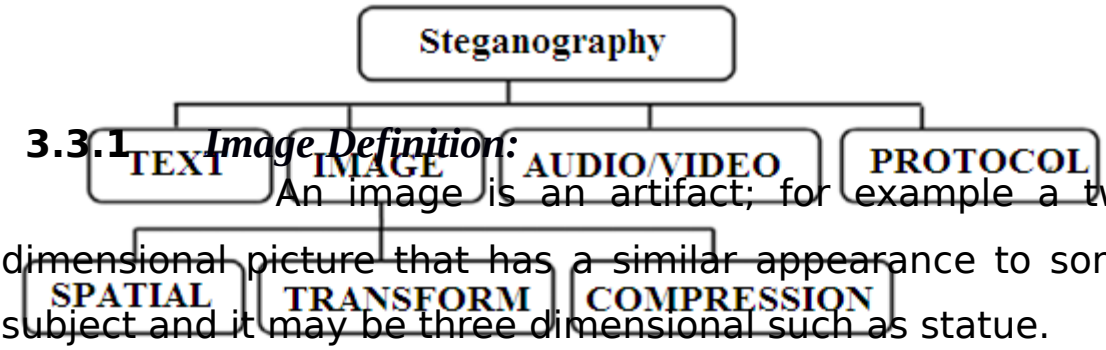
.....3.5

3.3 *Steganography techniques:*

The classifications of Steganography techniques based on the types of cover files shown in Figure 3.1. Almost all digital file formats can be used for Steganography, however only those with a high degree of redundant bits are preferred. The larger size of audio and video files makes

them less popular as compared to images. The term protocol Steganography refers to embedding information within network protocols such as TCP/IP.

Figure3.1: Classification of the Steganography techniques



3.3.1 *Image Definition:*

An image is an artifact; for example a two dimensional picture that has a similar appearance to some subject and it may be three dimensional such as statue.

Digital image is a representation of two dimensional image using ones and zeros (binary) .digital image in the computer is an array of numbers that represent light intensities at various points (pixels). These pixels make up the image's raster data. Digital images are stored in either 24-bit (true color images) or 8-bit per pixel files. A common image size is 640 × 480 pixels and 256 colors (or 8 bits per pixel). Such an image could contain about 300 Kb of data. Such large size images should be avoided since the attention when sending over a network or the Internet. Hence 8-bit color images, like GIF files, can be used to hide information. Here, each pixel is represented as a single byte, and the pixel's value is between 0 and 255. Grey-scale images are preferred because the shades are changed very gradually between palette entries. This increases the image's ability to hide information.

3.3.1.1 *Spatial domain:*

The ways of embedded the data in the digital image shown in figure3.1. No one best than other All have advantage and drawback .Spatial Domain is way we will use to hide the data in an image because it give high embedding capacity, ease of implementation and imperceptibility of hidden data. The major drawback is its vulnerability to various simple statistical analysis methods, so in our project this drawback can be reduce by adding another level of security (Encrypt the data before embedded) to difficult the statistical analysis.

In spatial domain cover-image is first decomposed into bits planes and then the bits planes are replaced with the secret data bits based on the Steganography algorithm.

3.3.1.1.1 *Lest Significant Bit (LSB) algorithm:*

It’s a simple approach to embedding information in an image. It exploits the fact that the level of precision in many image formats is far greater than that perceivable by average human vision. Therefore, an altered image with slight variations in its colors will be indistinguishable from the original by a human being, just by looking at it. By using the least significant bits of the pixels’ color data method to store the hidden message, the image itself will seem unaltered [2]. The embedding process consists of the

sequential substitution of each Least Significant Bit (LSB) of the image pixel for the bit message. So this method can camouflage a great volume of information. Applying (LSB) technique to each byte of a 24-bit image, three bits can be encoded into each pixel, as each pixel is represented by three bytes as in figure3.2 Applying (LSB) technique to each byte of an 8-bit image, only one bit can be encoded into each pixel, as each pixel is represented by one byte [6].

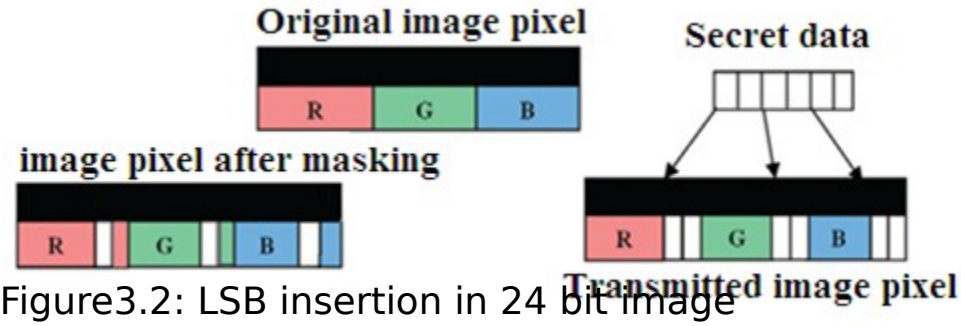


Figure3.2: LSB insertion in 24 bit image

3.3.1.1.2 *Random approach:*

This method is based on generating a matrix, the maximum element in this matrix less than or equal to the image size and the minimum element equal to one .the elements of the Matrix represents the embedded position for the secret data inside the image

3.3.1.1.3 *Data Hiding using Convolution or bit XOR:*

This method was proposed by Daneshkhah, et al. the Results show that the embedding capacity increases up to two bits per pixel. A convolution decoder is used in this method as shown in the Figure3.3. Each time 4(LSB) of a pixel enter the decoder machine. Three XOR operations create three outputs n1, n2, and n3. Suppose n2, n3 as the hidden message. If n2, n3 be the same as hidden information, then there is no need to manipulate the original image; if not then change the original image in a way to cause the output of the decoder to be equal to the hidden message.

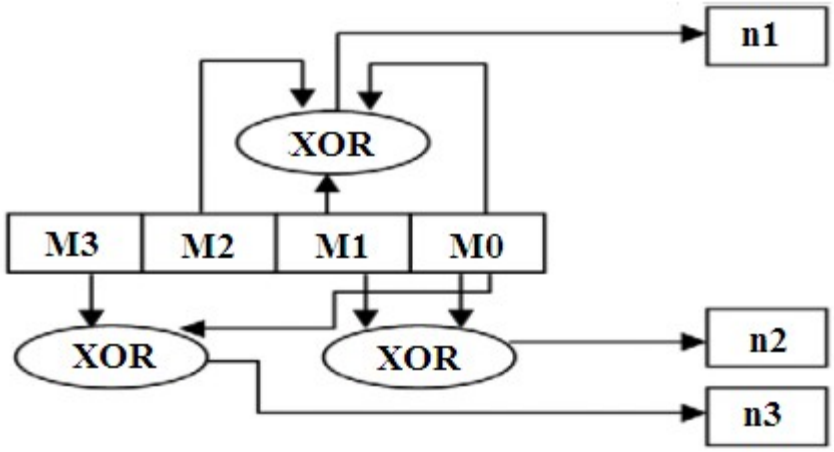


Figure 3.3: Convolution decoder machine

3.4 *Algorithm Performance:*

To measure the system (project) performance we calculate the following parameters:

3.4.1 *Mean Square Error (MSE):*

The mean squared error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the quantity to be estimated.MSE is then calculated as follow:

$$MSE=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[C(x,y)-S(x,y)\right]^2$$

..... **.....3.6**

Where

C: is the cover image (original image) with dimensions M.

S: is the Stego image with dimensions N.

X, Y: are the positions of pixel in image

3.4.2 *Peak signal to Noise ratio (PSNR):*

PSNR is a standard measurement used in Steganography technique in order to test the quality of the

stego images. The higher the value of PSNR, the more quality the stego image will have.

If the cover image is C of size M × M and the stego image is S of size N × N, then each cover image C and stego image S will have pixel value (x, y) from 0 to M-1 and 0 to N-1 respectively. The PSNR is then calculated

As follows:

$$PSNR = 10 \log \left(\frac{MAX^2}{MSE} \right)$$

.....

.....3.7

Where

MAX is the maximum possible pixel value of the images.

3.4.3 *Mean & Stander deviation:*

The "mean" is the average of the numbers divide by the number of numbers. It is often calculate by

$$\overline{\mathrm{X}} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

.....

.....3.8

The "Stander deviation "is a measure of the dispersion of a set of data from its mean, it represented by the symbol

sigma(σ).A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data points are spread out over a large range of values. To calculate the standard deviation, first compute the difference of each data point from the mean, and square the result of each.

$$s_N = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

.....

.....3.9

In this chapter, communication system model (A Graphical User Interface) was designed using MATLAB to load data (file or short message), encrypt it automatically and transmit it as well as retrieve it, and to determine the effect of the Data size and Stego algorithm on the image by calculate PSNR, MSE after

the data embedded, and Mean, Stander Deviation before and after the Embedding.

4.1 *Simulation Environments:*

MATrix **LAB**oratory or (MATLAB) is a programming language for technical computing that used for a wide variety of scientific and engineering calculations, especially for automatic control, image and signal processing. MATLAB is also noted for its extensive graphics capabilities (creation of user interfaces).

The simulation is two user interfaces (2GUI); Transmitter and the Receiver in each interface there are component programmed by MATLAB language.

4.2 *Simulation Description:*

Simulation model consist of two user interface; transmitter and receiver.

4.2.1 *Transmitter Model:*

Transmitter model consist of two stages, the first stage is the preparing stage and the second is the encryption, embedding and image save stage which we called Embedding stage.

The preparing stage is stage for image and Stego algorithm selection, calculation and display of the image mean, image

stander deviation, minimum pixel, and maximum pixel and image total pixel. Figure4.1 show the flow chart for this stage and Figure4.2 show the model for this stage.

Figure4.1: preparing stage flow chart

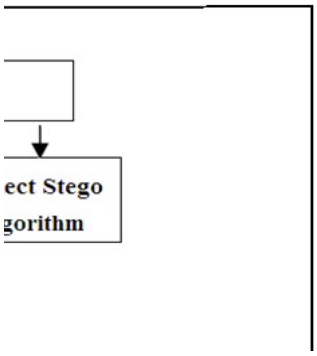


Figure4.2: The preparing stage model

The second stage start by the data loading, which it can be insert directly (short message) or loaded from the computer (file) then the data is encrypt automatically using modification RSA algorithm, the image and the Stego algorithm selected in the preparing stage are used to embedding the encrypted data successfully .and again the image mean, image stander deviation, MSE and PSNR are calculate to show the quality of Stego image. Image save is the last part in the second stage which it help in keeping the hide data in any location in the computer. Figure4.3show the second stage model and Figure4.4 show the flow chart for this stage.

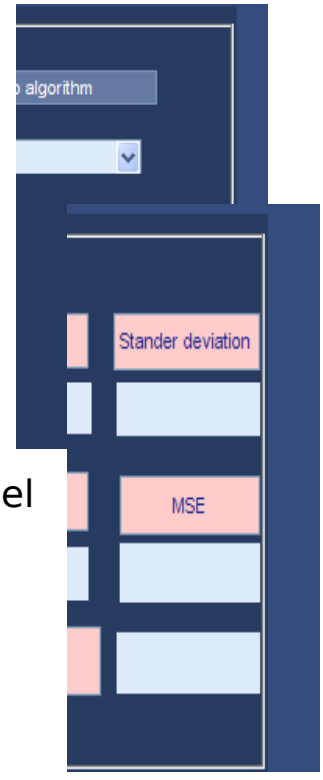


Figure4.3: The Second stage model

Figure4.4: Second stage flow chart

4.2.2 *Receiver Model:*

Receiver model includes text edit for data display and push button to extract the data from the received image. Figure4.5 show the receiver model and Figure4.6 Show the receiver flow chart

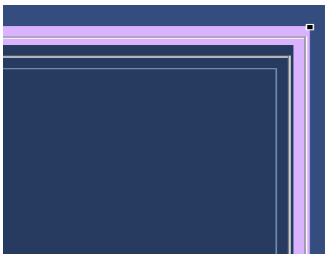


Figure4.5: the receiver model

Figure4.6: Receiver flow chart

4.3 *Result and discussion:*

The effect of the Steganography algorithm on the image is determined by calculates the MSE and PSNR.

In all figures; MSE value multiply by 10 and PSNR divided by 10

Figure 4.7 shows the effect of the Least Significant bit algorithm on the different images size when the embedded data is 10KB (81920bit).

Figure 4.7: The effect of LSB algorithm on the different images sizes

Figure 4.7 shows that when the size of the embedded data is small compared with the size of the image the MSE is decrease and PSNR is increase so there is no visible impact on the image.

Figure 4.8 shows the effect of Convolution algorithm in the different images size when the embedded data is 10KB (81920bit).

Figure 4.8: The effect of the Convolution algorithm on different images sizes

Figure 4.8 shows that when the size of the embedded data is small compared with the size of the image the MSE is small and PSNR is large so there is no visible impact on the image. Also shows that convolution algorithm is lower quality than LSB algorithm because for the same size of the data the PSNR is small.

Figure 4.9, Figure 4.10 and Figure 4.11 shows the effect of the Random approach algorithm on the different images size when the embedded data is 10KB (81920bit).

Figure 4.9: The effect of the Random approach algorithm on different images sizes

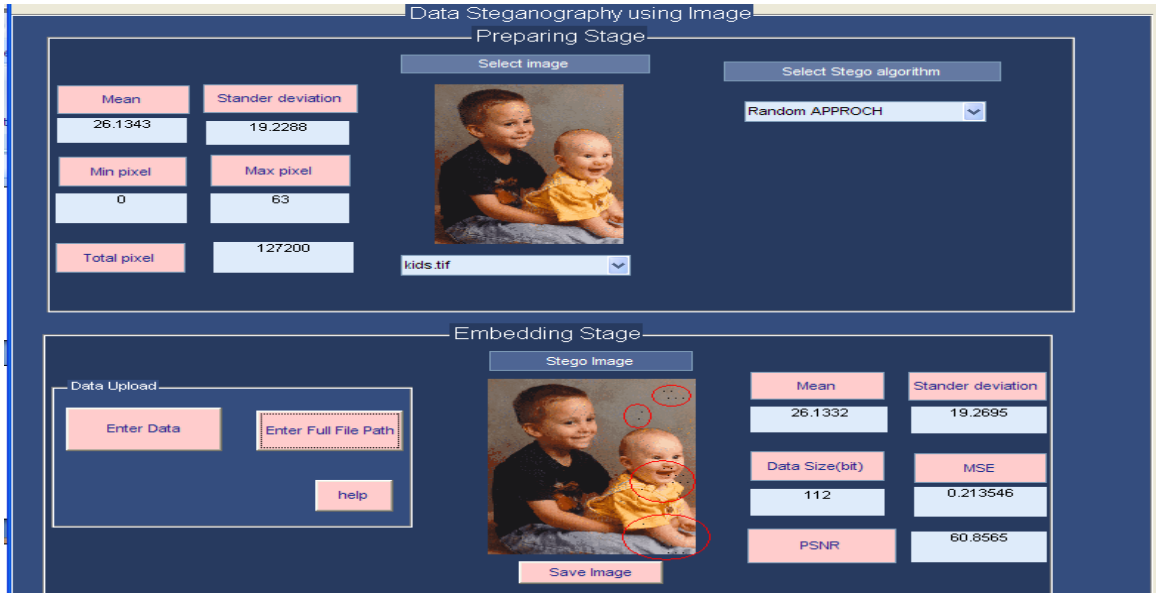


Figure4.10: The effect of Random approach algorithm on the kids image

Figure4.11: The effect of Random approach algorithm on the football image

Figure 4.9, Figure 4.10 and Figure 4.11 shows that the MSE is small and PSNR is large when the size of the data is small compared with the image size, and the visible impact of this algorithm depends on the exchange value, therefore there is no visible impact on the image if the value of the embedded data is same as the value of pixel but there is a black spot

appear on the image if there is large different between them, Figure 4.13 and Figure 4. 14 show the effect of this algorithm on the kid's image and football image respectively. Also it appears that the Random approach algorithm is lower quality than convolution algorithm and LSB algorithm.

Figure 4.12 shows the effect of LSB algorithm based Random approach algorithm on the different images size when the embedded data is 10KB (81920bit).

Figure 4.12: The effect of the LSB algorithm based Random approach algorithm on different images sizes

Figure 4.12 shows that the MSE is small and PSNR is large when the size of the data is small compared with the image size, and there is no visible impact on the image because only the LSB bit possible to change. Also it appears that the LSB algorithm based Random approach algorithm have better quality than all algorithms mentioned.

Figure 4.13 shows the effect of Convolution algorithm based Random approach algorithm on the different images size when the embedded data is 10KB (81920bit).

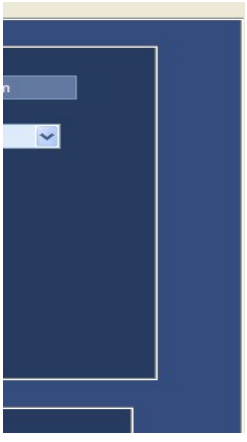
Figure 4.13:The effect of the Convolution algorithm based Random approach algorithm on different images sizes

Figure 4.13 shows that the MSE is small and PSNR is large when the size of the data is small compared with the image size, and there is no visible impact on the image because only the LSB bit possible to change. Also it appears that the convolution algorithm based Random approach algorithm is lower quality than the LSB algorithm based Random approach.

4.4 *Simulation Software:*

Transmitter model provide two functions for the user to use; hide &save to transmit or hide & save to hide. Both of them start by the image and the Stego algorithm selection and then load the data which it can be short message or file.

4.4.1 *Case Study: Case One*



In our system we embed and extract the data using the LSB based Random approach .so if we want to transmit the message ‘help’ we start by select the image we want and then press on ‘enter data’ button to write the message and click on ‘ok’ button when finish. GUI for this portion shown in Figure4.14

Figure4.14: GUI for the short transmit

After the data insertion was finish the embedding process is achieved and the Stego image appears to the user to press on save image to save it on the computer in any location. Figure4.15 shows the process of image saves. And Figure4.16 shows the successful of the save operation.

Figure 4.15: The process of image save

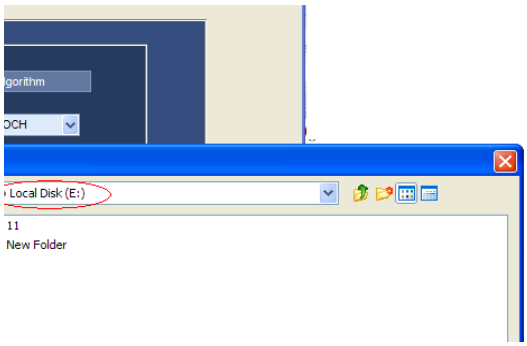


Figure4.16: The successful of image save

At the Receiver side to Extract and display the data only presses on ‘get data’ button. And for our purpose here (ensure that the data was embedded in the image and transmitted) the receiver and the transmitter in the same device .figure 4.17 show the extraction and the display process of the data from the Received image in the local disk (E).

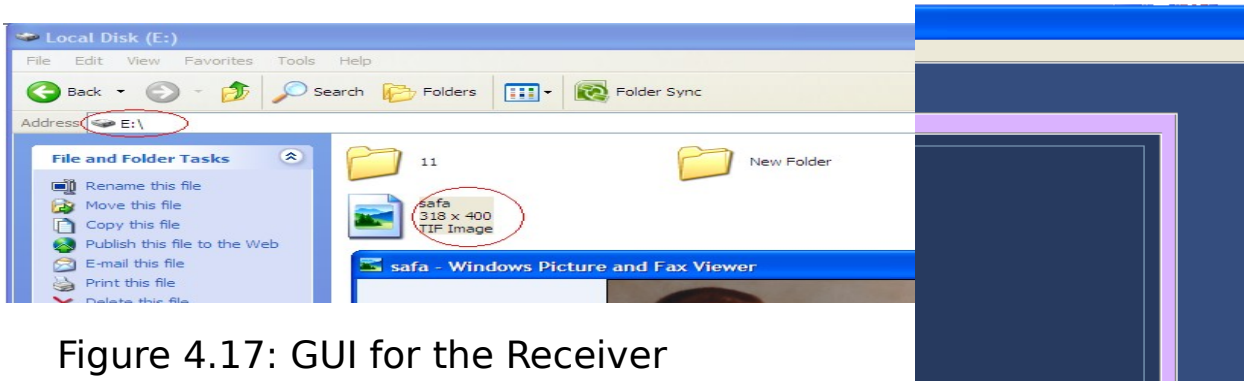


Figure 4.17: GUI for the Receiver

4.4.2 *Case two:*

Another task for system is file transmit, this task start normally by select the image and then press on ‘enter full file path’ to write the fully file path then press on ‘ok’ when finish. if the user forget the file path press on ‘help’ button to get the full path as in Figure 4.19 and again press on ‘enter full file path’ to write it .Figure 4.18 show the file load operation .

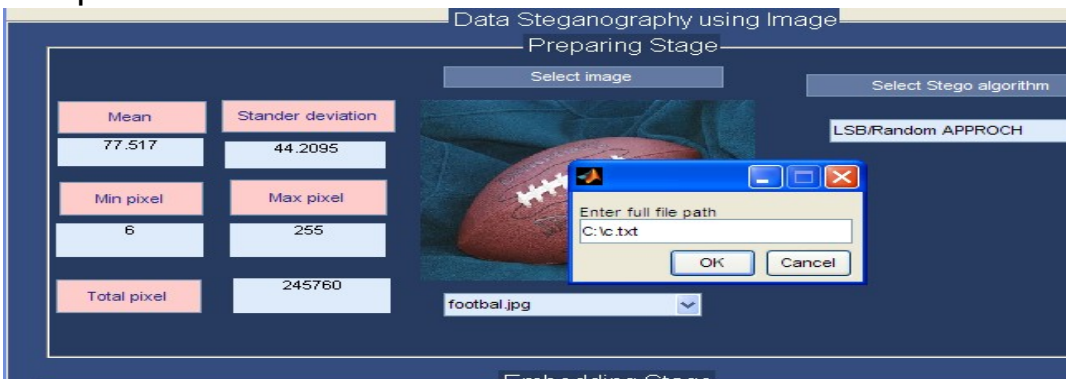


Figure 4.18: GUI for file load operation

Figure4.19: GUI for help operation

After the file was load the embedding process is achieved and the Stego image appears to the user to press on save image to save it on the computer in any location as in Figure 4.20. And Figure 4.21 show the successful of save operation

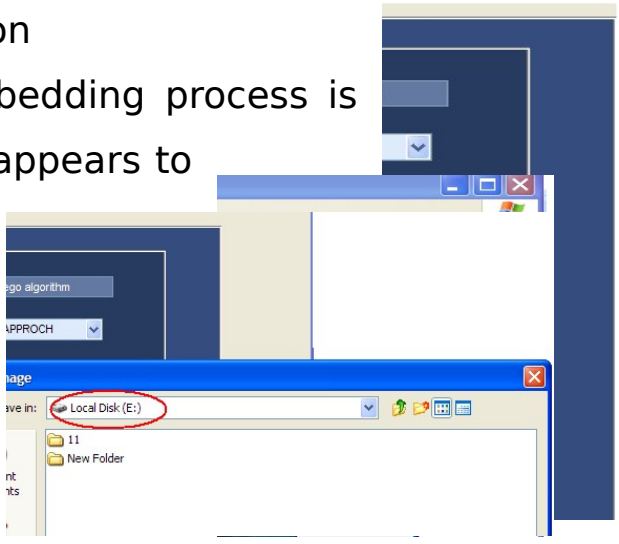


Figure4.20: GUI for image save operation

Figure4.21: The successful of image save operation

Again at the Receiver side presses on ‘get data’ button to Extract and display the data. Figure4.22 shows the extraction and the display process of the data from the Received image in the local disk (E).

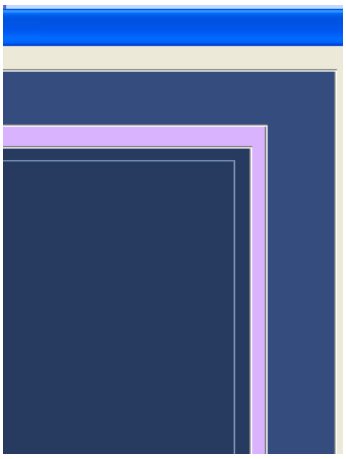
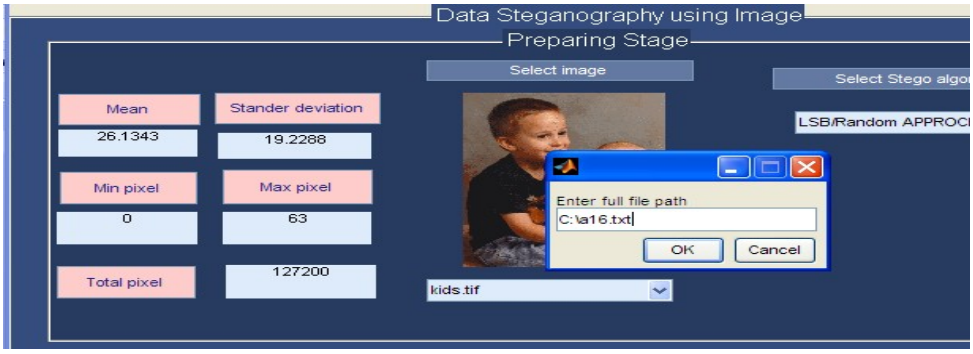


Figure4.22: GUI for the Receiver

4.4.2 *Help Dialog:*

Help dialog is warning message appears when the file size is greater than the image size to warn the user to select another image to avoid the loss of any portion of the file .Figure 4.23 show the process of enter large file and Figur4.24 show the appearance of help dialog and to prove that this warning message is true we draw a Red circle around the image size and data size.



`Figure4.23: insertion of large file

Figure4.24: the appearance of the help dialog

Another case study of help dialog at large image and large data, this is show as in Figure 4.25and Figure 4.26

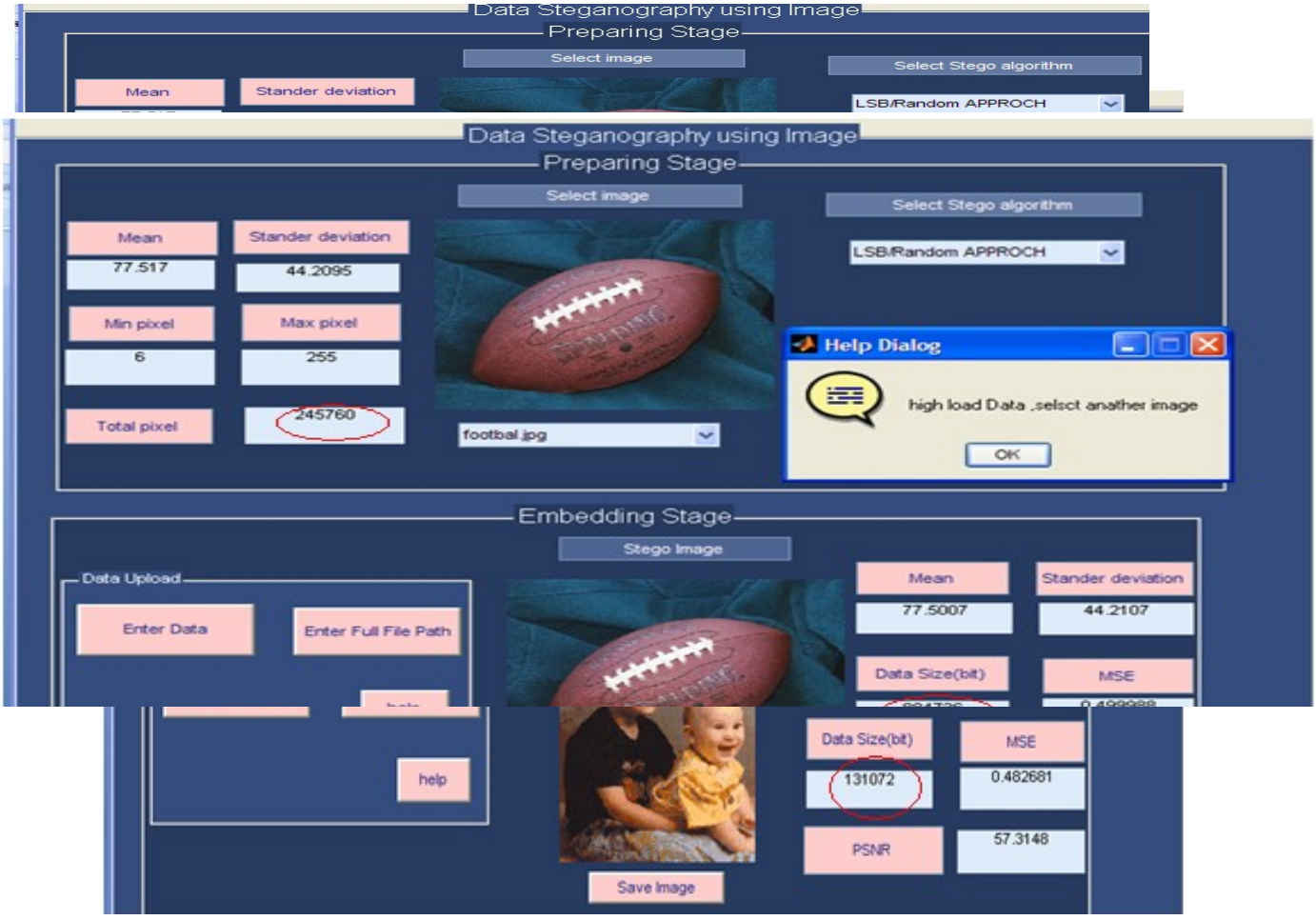


Figure4.25: insertion of large file

Figure4.26: the appearance of the help dialog

5.1 *Conclusion:*

The Steganography system was implemented using Mat lab language and digital image (color image) which represented a very good media for data Steganography, the results was obtained, discussed and it shows that The Random approach algorithm have a visible impact appears as black spot on the image because of the replaced process of pixel value by data value which mean the completely change of the image details (data or color).

Also the results show that when LSB was replaced using Convolution algorithm or directly there was no difference between the original image and the Steganography image because of the partial change on image details. LSB exploits the fact that the level of precision in many image formats is far greater than that perceivable by average human vision. Therefore, an altered image with slight variations in its colors will be indistinguishable from the original by a human being, just by looking at it. By using the least significant bits of the pixels' color data to store the hidden message and the image itself will seem unaltered.

At the receiver side (extraction process) the results show that it is more secret and more complex for data security to change LSB based on the random approach to pixel.

At the end of the research we found that; the relation between the data size and their effect on the image depend on the image size compared with the data size and therefore No visible impact on the image when the PSNR is greater than 40 db.

5.2 *Recommendation:*

Steganography system at the Receiver successful of short message Extraction, so we recommend working on file Extraction, the transmitter and the Receiver will be at different side to see the real effect of image sending process on the data.