POPULAR DEMOCRATIC REPUBLIC OF ALGERIA

Higher education and scientific research's ministry

University of KASDI MERBAH OUARGLA

Faculty of new information technologies and communication

Department of computer and information technology

LMD Master 2019/2020

Option: Network Administration and Security

Dissertation

For the Master Degree in Network Administration and Security

Title:

Cryptography algorithm based on rotating view and genetic algorithm

Presented by :

Kaouthar Souigat

Saliha Hadji

Jury members:

Dr. Zga Adel	University of KASDI MERBAH OUARGLA	Jury-header
Dr. Boukhamla Akram	University of KASDI MERBAH OUARGLA	Examiner
Dr. Belekbir Djalila	University of KASDI MERBAH OUARGLA	Superviser

Academic Year 2019-2020

Acknowledgements

Above all, we would like to thank ALLAH the all-powerful for giving us health, faith and strength to complete our studies and the desire to begin and end this work.

We would like to thank Ms. Djalila belkebir our supervisor who mentored our work ,for shes orientation and his precious advices were really useful and appreciated.

We would like to thank our dear parents for their continued support and urge us to move forward and finish this work

Our respectful thanks to all members of the jury for the interest in our work and taking a time to evaluate it.

DEDICATION

This modest work is dedicated to:

To my beloved parents,

You have been an outstanding inspiration to me. Getting to this stage in my life has taken a lot of work, but it is nothing compared to how you worked and sacrificed for me. You are the number one reason I am where I am today.

Without your continued support I could never have accomplished so much. I love you.

To my brothers and my sisters may God preserve and protects them

The whole family for their moral and financial support, their encouragement.In particular, my uncle, Dr. Fateh Al-Din Shaneen, was my supporter throughout my academic career..

To my teacher, Ali Algayma who was the reason for my love for mathematics and supported me in my early years.

To all my teachers at the Department of Computer Science and Information Technology.

To all my dear friends and colleague of studies especially,

My wonderful partner hadji saliha .

SOUIGAT KAOUTHAR

DEDICATION

I dedicate this modest work to..

To my dear parents .." Mom"," Dad" ..

To my honorable sisters ..

Their husbands, their children ..

Each in his name ..

For the whole family ..

My family," Hadji" and" Kochi" ..

To all my close friends ..

"Roza"," Hadjer","Sihem","BF4".. All ..

To my business partner, "Anfal" ..

To all my dear colleagues ..

My fateful teachers ..

Everyone has love in the heart ..

He carries in his heart a love for me ..

This is for all of you ..

Thanks for everything ..I love you..

Saliha hadji

Abstract

Encryption has been a form of secure communication for decades, and with the development of human civilization, the demand for developing new forms of encryption is necessary to ensure the security of communications.

The encryption algorithm is a technology for converting data transferred from a readable form to an unreadable form to prevent unauthorized access while ensuring the preservation of what the data is. We specialized in this research in ways to encrypt images, so we presented several algorithms to carry out the operation, among which are the ones that are already in place: RC4 that is used in any secured wired equivalent privacy network (WEP), Under a common key[80], And LSB Which is used in the technique of hiding information in an image, and to improve and make it usable for image encryption, we proposed a new algorithm based on the projection, the idea of this method is to consider the position where each pixel is projected in a new image. This position is defined by the intersection of the light beam of the pixel and the image to be considered according to the virtual position of the observer, by a certain angle and distance.

In order to improve the algorithm more and make it less susceptible to fracture, we decided to develop this algorithm through hybridization between the projection algorithm and Genetic Algorithm, That is based on merging and transferring, we implemented the latter in two forms, the first by calculating fitness and the second without calculating it. This algorithm provides safer image encryption, minimum data loss, maximum speed and maximum distortion in encryption.

Keywords :

Rivest Cipher 4, Least Significant Bit Algorithm, Direct Image Projection, Genetic Algorithm, Fitness Function

Résumé :

Le cryptage est une forme de communication sécurisée depuis des décennies, et avec le développement de la civilisation humaine, la demande de développement de nouvelles formes de cryptage est nécessaire pour assurer la sécurité des communications.

L'algorithme de cryptage est une technologie permettant de convertir les données transférées d'une forme lisible à une forme illisible pour empêcher tout accès non autorisé tout en assurant la préservation de ce que sont les données. Nous nous sommes spécialisés dans cette recherche sur les moyens de crypter les images, nous avons donc présenté plusieurs algorithmes pour réaliser l'opération, parmi lesquels ceux qui sont déjà en place: RC4 qui est utilisé dans tout réseau de confidentialité équivalent filaire sécurisé (WEP), sous un clé commune [80], Et LSB qui est utilisé dans la technique de masquage d'informations dans une image, et pour l'améliorer et la rendre utilisable pour le cryptage d'image,nous avons proposé une nouvelle algorithme basée sur la projection, l'idée de cette méthode est de considérer la position où se projette chaque pixel dans une nouvelle image. Cette position est définie par l'intersection du rayon de lumière du pixel et l'image à considérer en fonction de la position virtuelle de l'observateur, Par un certain angle et d'une certaine distance.

Afin d'améliorer davantage l'algorithme et de le rendre moins sensible à la fracture, nous avons décidé de développer cet algorithme par hybridation entre l'algorithme de projection et l'algorithme génétique, c'est-à-dire basé sur la fusion et le transfert, nous avons implémenté ce dernier sous deux formes, la première par calculer la condition physique et la seconde sans la calculer. Cet algorithme fournit un cryptage d'image plus sûr, une perte de données minimale, une vitesse maximale et une distorsion maximale dans le cryptage.

Mots clés :

Rivest Cipher 4, Algorithme de Bit le Moins Significatif, Projection d'Image Directe, Algorithme Génétique, Fonction de Fitness

ملخص

كان التشفير شكلاً من أشكال الاتصال الآمن لعقود من الزمن، ومع تطور الحضارة الإنسانية، فإن الطلب على تطوير أشكال جديدة من التشفير ضروري لضمان أمن الاتصالات.

تعد خوارزمية التشفير، تقنية تحويل البيانات المنقولة من شكل مقروء إلى شكل غير قابل للقراءة لمنع الوصول غير المصرح به مع ضمان حفظ ماهية البيانات. وتخصصنا في هذا البحث في طرق تشفير الصور فقدمنا خوارزميات عدة لتنفيذ العملية منها المعمول بها من قبل المتمثلة في (RC4) التي تستعمل في شبكة الخصوصية المكافئة السلكية (WEP) المؤمنة ، تحت مفتاح مشترك[80]، و (LSB) التي تستعمل في تقنية إخفاء المعلومات في صورة و من أجل تحسين هذه التقنية وجعلها قابلة للاستعمال في مجال تشفير الصور، اقترحنا خوارزمية جديدة مبنية على الإسقاط تقوم بتشفير الصور، فكرة هذه الطريقة هي الاخفاء في الموضع الذي يعرض فيه كل بكسل في صورة جديدة، يتم تحديد هذا الموضع من خلال تقاطع شعاع ضوء البيكسل والصورة التي سيتم النظر فيها وفقًا للموضع الافتراضي للمراقب من خلال زاوية و مسافة معينين.

ومن اجل تحسين الخوارزمية أكثر وجعلها أقل عرضة للكسر ارتأينا تطوير هذه الخوارزمية من خلال التهجين بين خوارزمية الاسقاط والخوارزمية الجينية، التي تعتمد على أساس الدمج والتحويل، قمنا بتنفيذ هذه الأخيرة بشكلين الأول باحتساب اللياقة والثاني بدون احتسابها. حيث توفر هذه الخوارزمية تشفيرًا أكثر أمانًا للصور، الحد الأدنى من فقدان البيانات والتشويه الأقصى في التشفير.

الكلمات المفتاحية:

ريفست شيفر 4 (RC4)، خوارزمية البت الأقل أهمية، إسقاط مباشر للصورة، الخوارزمية الجينية، وظيفة اللياقة

List of contents

Figures list:

Tables list :

Abbreviations list

AES: Advanced Encryption Standard

BER: Bit Error Rate

CGP : Cartesian Genetic Programming

CRC: Cyclic Redundancy Rheck

CRC32 : the 32-bit CRC function

DE: Differential Evolution

DES: Data Encryption Standard

DIP: Direct Image Projection

DIP-GA: Direct Image Projection using Genetic Algorithm

DSA: Digital Signature Algorithm

DSS: Digital Signature Standard

EC : Evolutionary Computing

ECC: Elliptic Curve Cryptography

EP : Evolutionary Programming

ES : Evolution Strategies

FL : FLuzy Logic

GA : Genetic Algorithms

GF: Galois Felds

GP : Genetic Programming

IDEA: International Data Encryption Algorithm

JND: Just Noticeable Difference

KSA: Key-Scheduling Algorithm

LMSE: Laplacian Mean Square Error

LSB : Least Significant Bit

MD: Maximum Difference

MD5: Message Digest algorithm version 5

MIT: *Massachusetts Institute of Technology*

MSE: Mean Square Error

NAE: Normalized Absolute Error

NIST: National Institute of Standard and Technology

NK: Normalized Cross-Correlation

NN : Neural Networks

NSA: *National Security Agency*

PMSE: Peak Mean Square Error

PRGA: Pseudo-Random Generation Algorithm

PSNR: Peak Square Noise Ratio

RC4: Rivest Cipher 4" or "Ron's Code 4

RGB : Red, Green and Blue color

RSA: Rivest, Adleman and Shamir

SC: Soft Computing

SC: Structural Content

SHA: Secure Hash Algorithm

SHS: Secure Hash Standard

SSL: Secure Socket Layer

US: United State

WEP: Wi-Fi security protocol

INTRODUCTION GENERAL

After the Internet and technology were the monopoly of a certain class of people for years, the world has witnessed a high development in modern technologies and Internet communications, so the Internet has become available to all segments of society, and digital media has become the most common and used tool for exchanging data. Most of the digital data is in the form of images, and it is transmitted in several applications available such as websites, e-mail, chatting, e-commerce, e-books, news, etc.

Digital content faces many challenges, such as copyright protection, tampering and authentication. The biggest challenge facing application programmers is to ensure the security and privacy of the information transmitted through their applications and the communications networks used, and to solve these problems, there are several methods used to protect the transferred data.

Many methods are efficient and effective is encoding information before sending it and then decoding it at the recipient. Historically, encryption was known in its limited initial use by the ancient Egyptians about 4000 years ago and is still in the twenty-first century but with better technologies, and to this day students, teachers and scientists are still conducting research about it in the whole world. Because whenever a secret technology is revealed, it will be dismantled at some point. Therefore, the door to research and updating methods to ensure information security remains always open. Encryption has played an important role in protecting information in its digital form and providing secure services. The encryption algorithm is a technique for converting data transferred from a readable form to an unrecognizable form to prevent unauthorized access, unless you know specific information about the encryption process.

Despite all this, encryption algorithms are still vulnerable to breakage, especially those that are characterized by the classic character of encryption. The extensive use of encryption processes has led to the emergence of several gaps and defects in the algorithms, especially the most widespread, in addition, some of them are known to be insecure, and these issues highlight the need for more reliable methods of encryption and move us to the innovation of modern encryption algorithms.

In the field of encryption, many algorithms have appeared to ensure the safety and security of information, some of them have shown their strength and some of them have shown their weaknesses after code-breaking algorithms have bro-

ken them. Modern encryption techniques are the strongest solutions to most of these problems, and between this and that, we decided to suggest. In this work, a new projection-based algorithm encodes images; the idea of this method is to consider the position in which each pixel is displayed in a new image. This position is determined by the intersection of the pixel light beam and the image that will be considered according to the default position of the observer through an angle and a certain distance, [6], aims to enhance more safety.

With the increase in gaps and problems in the encryption process and its complexity and the inability of traditional software, solutions to understand them prompted researchers to delve into finding effective algorithms that help in finding suitable and ideal solutions to complex issues. It also help in the speed of accessing, storing and retrieving solutions, and with the emergence of advanced mathematical procedures in encryption techniques. Decoding encryption algorithms have become more complex daily, a coherent structure has been reached for an intelligent software structure, and one of these effective algorithms, the genetic algorithm that gains most of its strength in the search from reproduction, intersection. Mutation, is one of the algorithms of computational development and it is one of the modern methods where the importance of using these the method for solving large complex problems that have a large number of alternative solutions [4].

Genetic algorithms are basically the techniques of machine learning, heuristic research and optimization, based on the Darwinian theory of the survival of the fittest natural genes. The genetic algorithm examines complex problems by developing a set of possibilities for possible solutions to these problems. Therefore, in our work, we will be using it in order to improve our projection based algorithm. One of the main reasons for working with genetic algorithms is that it provides more safety and security for the data, which can greatly preserve the reliability and protect the images. The aim of our work is to study and evaluate the algorithm proposed above. All this study is organized in a document of three chapters, each chapter completing the previous one.

In the first chapter, the various concepts and main parts in the field of cryptography and evolutionary algorithms will be described. We will first define what cryptography is and the most important concepts related to it. Talk about encryption algorithms of all kinds with mentioning some common examples and this is after dealing with the history of encryption, and then we will shed light on

the concepts Fundamental to evolutionary algorithms and explain some of these algorithms, without forgetting to talk about coding using evolutionary computing algorithms.

In the second chapter, devoted to implementation, in this chapter we will explain the concept, methods of work, advantages and disadvantages of all the algorithms we used in the process of encoding image data. Among them are the previously applied algorithms represented by the RC4 algorithm and the LSB algorithm, which we proposed represented by the projection-based algorithm and the genetic algorithm, and we provided the hybrid algorithm for the projection method and the genetic algorithm, in two ways, the first without calculating fitness and the second by calculating it.

While in the third chapter, which aims to provide a comparison between the various algorithms that we used in our project, by studying visual analysis, graph analysis and results of image quality tests.

Finally, a conclusion will make it possible to evaluate the work performed with a summary of the results of the tests taken and to discuss the proposed ideas with an openness to future work.

CHAPTER I: OVERVIEW OF CRYPTOGRAPHY AND EVOLUTIONARY ALGORITHMS

I.1. Introduction

Encryption have been one of form of secure communication for the last decades. As cryptanalysis attacks get stronger, the need to invent new forms of encryption algorithm as a cryptanalysis attacks-defender is necessary. This chapter is divided into two main sub-sections. The first one is related to defining the cryptography, as well as some related preliminaries. We will also detail the main used algorithms on the most current time. The second part of this chapter is related to describing the research domain of the contribution of our work, which is the hybrid encryption using evolutionary algorithms (EAs). First, we will define some concept of EAs. Then, we will detail the most used EAs on literature. Finally, we will end up this chapter with related works about the hybrid use of cryptography and EAs.

I.2. Historical introduction to cryptography

The first attempts for cryptography are certainly rooted in historical tradition, so we'll talk a little bit about the history of cryptography. Strictly speaking, encryption begins with the origin of language writing, the first sign of cryptography in ancient times goes back as early as 1900 B.C, with the Egyptians, The Egyptians used hieroglyphs for communicating among a selected category of people, usually upper nobility, A substitution cipher is when one symbol/character is replaced with another. In ancient Greece, moving forward to 500 B.C, the Spartans developed the Scytale (Figure 1.1). The Scytale was a cylindrical device with parchment wrapped around in a spiral formation; secret writing was established by the so-called scythe4 that became famous for its military purpose. Around 100 B.C., and from ancient Rome, the famous Caesar cipher arose (Figure 1.2), Julius Caesar concerned with messages being intercepted by enemy soldiers and the lack of trust he had for his massagers, used a variety of simple substitutions ciphers for military purposes and acquaintances [8].

Figure I.1 Scytale [14].

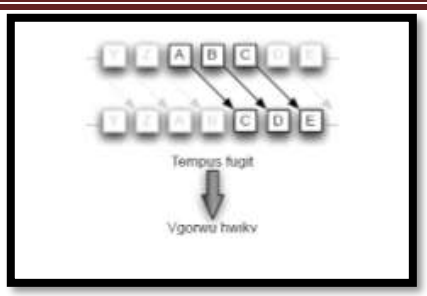


Figure I.2 Caesar Cipher with a shift of two to the right [52].

Cryptography has been in use for centuries, the earliest ciphers were either used transposition or substitution, and messages were encoded and decoded by hand. However, these schemes satisfied only the basic requirement of confidentiality. In more recent times, with the invention of processing machines, more robust algorithms were required, as the simple ciphers were easy to decode using these machines [36]. As we all know, Cryptography played a significant role in World War II. The Germans and its allies invested a lot of time and revenue on building cryptosystems. They were a combination of mechanical and electronic systems. Enigma is a well-known cryptosystem (Figure 1.3). Enigma was used by all the branches of the German army [15].

Figure I.3 three-wheel Enigma machine [52].

Information Security made some significant leaps with the advancement of modern digital computers, Algorithms which were considered to be unbreakable are now deemed to be trivial to break [31].

Cryptography is part of our modern life: Telecommunication, Financial, Transport, Identification, Recreational …

I-3 Terminology of Cryptography

I-3.1 Information Security

Information security is essential for today's world since, for profitable and legal trading, confidentiality, integrity and non-reputability of the associated information are necessary [35]. Desired properties of a secure communication system may include any or all of the following:

- **Confidentiality:** Only an authorized recipient should be able to extract the contents of the encoded data.
- **Data integrity:** The recipient should be able to establish if the message has been altered during transmission.

  Plaintext = Decrypted (Encrypted (Plaintext)) [8]
- **Authentication:** The recipient should be able to identify the sender, and verify that the purported sender actually sent the message.
- **Non-Repudiation:** The sender should not be able to deny sending the message, if he actually did send it [53].
- **Service Reliability and Availability:** Since secure systems usually get attacked by intruders, which may affect their availability and type of service to their users. Such systems provide a way to grant their users the quality of service they expect [64].
- **Anti-replay:** The message should not be allowed to be sent to multiple recipients, without the sender's knowledge.
- **Proof of Delivery:** The sender should be able to prove that the recipient received the message [53].

I-3.2 What is Cryptography

The first use of the term cryptography dates back to the 19th [66], until modern times. Cryptography derivates from the two Latin words crypta and graph (in English, secret and writing) and is therefore referred as the science of secret writing [8]. Cryptography is an art of hiding confidential information from a third party by implementing keys which are only known by the communicating parties [15], The first use of the term cryptography dates back to the 19th [66], until modern times. Cryptography derivates from the two Latin words crypta and graph (in English, secret and writing) and is therefore referred as the science of secret writing [8]. Cryptography is an art of hiding confidential information from a third party by implementing keys which are only known by the communicating parties [15], to study of mathematical techniques related to aspects of information security. Cryptography referred to encryption, which is the process of converting plaintext into form ciphertext [33], Cryptography is not the only means of providing information security, but rather one set of techniques [53]. As defined in RFC 2828, "cryptographic system is a set of cryptographic algorithms together with the key management processes that support use of the algorithms in some application context" [29].

I-3.3 Why do we need Cryptography?

Cryptography plays an important role in today's and future's confidential data communication. For example, communications over telephone lines including faxes and e-mail messages, financial transactions, medical histories, e-banking and even other types of important information need secure communication medium. Sometimes, the medium is hacked by intruders and gets all of your information. Therefore, cryptographic applications provide the secure communication medium to transfer your data reliably, so that if any one tries to hack data, then it is not useful to him because the data is in encrypted form. Cryptography is one of the tools, which ensures more privacy. The ability to encrypt data, communications and other information, gives individuals, the power to restore personal privacy. Cryptography protects the world's banking systems as well, which is the requirement of today's world. Many banks and other financial institutions conduct their business over open switched networks like Internet. Without the ability to protect bank transactions and communications, criminals could interfere with the transactions and rob banks without a trace [47].

I-3.4 Concepts

When discussing cryptography, you must understand a few key terms like encryption, decryption, key, ciphertext, and plaintext...etc.

Cryptanalysis describes the discipline of analyzing and breaking cryptosystems (and more generally, information security services) for either improving the cryptosystem or finding vulnerabilities to gain access to undisclosed information [*8*].

A cryptanalyst is someone who engages in cryptanalysis.

Cryptology is the branch of mathematics encompassing both cryptography and cryptanalysis is called cryptology and its practitioners are cryptologists. We have already discussed about cryptography, so we now talk about cryptanalysis. Cryptanalysis is the study of mathematical techniques for attempting to defeat cryptographic techniques and more generally, the art and science of breaking ciphertext without the knowledge of proper method of decryption. A cryptanalyst is a person or machine that engages in cryptanalysis [*47*].

A cryptosystem is a general term referring to a set of cryptographic primitives used to provide information security services. Most often the term is used in conjunction with primitives providing confidentiality, i.e., encryption [*53*].

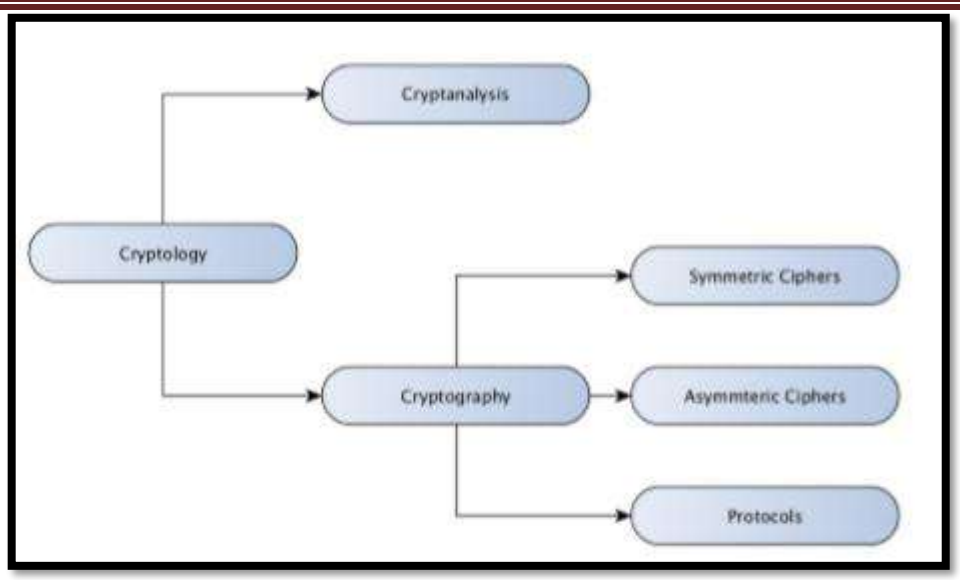


Figure I.5 Areas of expertise in cryptology [12].

- **Plaintext** is the original message presented in an intelligible form.
- **Encryption** is the transformation of the original message or plaintext to unreadable form.
- **Ciphertext** the encrypted form is referred, When the original message has been encrypted.
- **Decryption** is process of transforming the ciphertext back into plaintext in the readable form.
- **key**: During encryption and decryption, there is normal rules or algorithms that identify how the messages are to be transformed; the rules/algorithms are referred as keys [13].

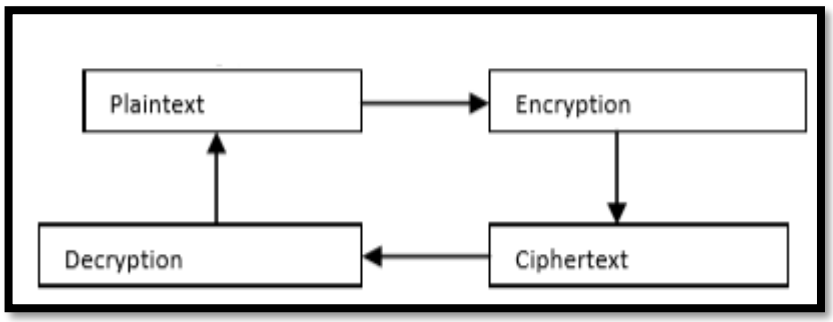


Figure I.6 Conventional Encryption Mod [64].

I-4 Types of Cryptography

There are many types of cryptography, including codes, steganography, ciphers and mathematical algorithms. Codes rely on codebooks. Steganography relies on different ways to hide or disguise writing. Ciphers include both computer-generated ciphers and those created by encryption methods. Nowadays, mostly cryptographic algorithms are implemented on theories of mathematics. Number theory and Group theory are the most widely used in the com-

puterized cryptographic algorithms. Following is the detail information about different types of cryptography.[47]

I-4.1. Codes and Codebooks

A well-developed code can represent an idiom and entire sentences with symbols, such as three or four letters group. A properly constructed code can give a high degree of security but the more same codebook is used, the less secure it becomes [47].

I-4.2 Steganography

Steganography is an ancient Egyptian technique of hiding the message using liquid like invisible inks, microscopic writing and hiding code words within sentences of a message. Cryptographers may apply steganography to electronic communications and that application is called transmission security [47].

I-4.3 Cipher

The word "cipher" means "secret" and it is the most popular and secure technique as compared with codes and codebooks as well as steganography. Ciphers are the secret codes used to encrypt plaintext messages. There are two general types of ciphers, namely Substitution and Transport ciphers.

In substitution cipher, an alphabet is to replace plaintext with other letter or symbol. In transposition ciphers, the mix up of letters in a word or sentence is made to convert it into unintelligible form [47].

I-4.4 Substitution Cipher

In simple substitution ciphers, an alphabet or symbol is substituted for each single or multiple letters. For example, a substitution cipher was devised long ago by Julius Caesar, which shifted all the letters in the alphabet by three places. To express Caesar's cipher mathematically, first replace each letter by an integer from 1 to 26, based on its position in the alphabet. That is 'A' is replaced by 1, 'B' by 2, 'C' by 3 and so on. Caesar cipher is represented by function f that has non-negative integer value 'a', where 'a' less than or equal to 26 [47].

$f(a) = (a+3) \bmod 26$

I-4.5. Transposition Cipher

In a transposition cipher, the order of plaintext alphabets is changed to get the ciphertext. The message is usually written without word divisions in rows of letters arranged in a rectangular block. The letters are then transposed in a prearranged order, such as by vertical columns, diagonals, or spirals, or by more complicated systems [47].

I-5 Cryptographic Algorithm

The cryptographic algorithm is based on cryptographic protocols developed by different scientist of the world. National Institute of Standard and Technology (NIST) and RSA Data Security System Inc. published algorithms that are using nowadays. For choosing and evaluating algorithm, we have several conditions:

- Choose the well-known agencies published algorithms that nobody broken the algorithm yet and cryptographers have scrutinized that algorithm and give their remarks in positive manner.
- We can trust a manufacturer based on the principle that a renowned manufacturer has a reputation to uphold and is unlikely to risk that reputation by selling equipment.
- We can also trust the government based on the belief that the government is responsible and wouldn't guide its citizen's wrong.
- We can write our own algorithms based on the idea that the cryptographic ability is secured as compare to other renowned algorithms.

I-5.1 Classes of Algorithms

There are several types of cryptographic algorithm having different features having confidentiality, authentication, key management etc. Below is the summarize table 3.1 of different algorithms and their capability [47].

Classes of Algorithms				
Algorithm	*Confidentiality*	*Authentication*	*Integrity*	*Key Management*
One-way hash functions	No	No	Yes	No
Message authentication codes	No	Yes	Yes	No
Digital signature algorithms	No	Yes	Yes	No
Symmetric encryption algorithms	Yes	No	No	Yes
Public-key encryption algorithms	Yes	No	No	Yes

Table I.1 Classes of Algorithms [47].

I-5.2 Confusion and Diffusion

The two basic techniques for obscuring the redundancies in a plaintext message are confusion and diffusion, introduced by Claude Shannon. Confusion is trying difficult to understand the relationship between the plaintext and the ciphertext. The easiest way to do this is through substitution. A simple substitution cipher, like the Caesar Cipher, is one in which every identical letter of plaintext is substituted for a single letter of ciphertext. Also, confusion is achieved in computers through the XOR binary operation. Diffusion is achieved through

the numerous permutations to increase the complexity. The simplest way to cause diffusion is through transposition. A simple transposition cipher, like columnar transposition, simply rearranges the letters of the plaintext [47].

I-5.3 Types of Cryptographic

I-5.3.1 Symmetric key Algorithms

Other terms used in the literature are single-key cryptography, one-key, private key, and conventional encryption [53]. The general perception of symmetric algorithms is that the two communicating parties have encryption and decryption method for which they have a secret key. A sender and a recipient share the same secret key, this key is used at the same time. both for encryption and for deciphering and must remain secret from any enemy observer [20]. Until 1976, the cryptographic techniques used were exclusively based on symmetric methods. Applications like data encryption and integrity checking of messages still use symmetric ciphers [15].

Each key K has an associated EK encryption function and a DK decryption function. The sender encrypts the plain text m to obtain the streaked text c = EK (m) and sends c to the recipient. The recipient restores the plain text m by calculating m = DK (c) [20].

This technique though simple and easy to implement, has obvious drawbacks, some of which are listed here:

- A shared secret key must be agreed upon by both parties.
- If a user has n communicating partners, then n secret keys must be maintained, one for each partner (as every communication group cg holds its own key, there are cgk*(cgk−1)/ 2 possible key pairs k within a domain).
- Authenticity of origin or receipt cannot be proved because the secret key is shared.
- Management of the symmetric keys becomes problematic [36].

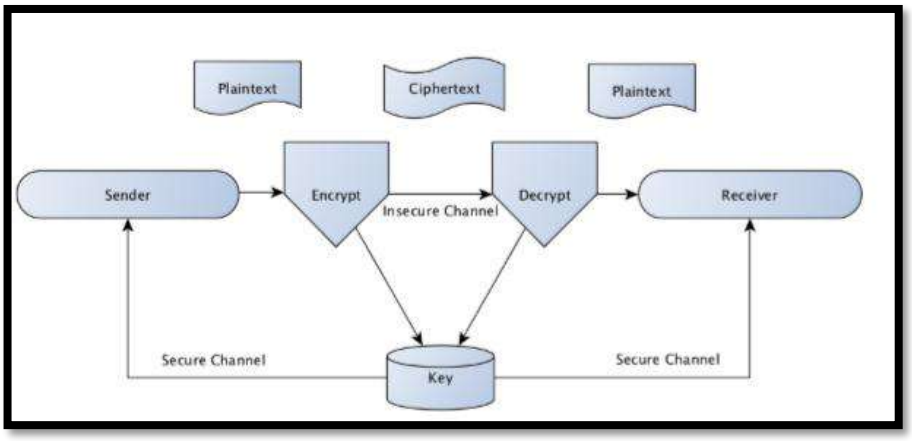


Figure I.7 Basic proceeding of symmetric key algorithms [53].

Symmetric key algorithms can be applied by Stream ciphers or Block ciphers

I-5.3.1.1 Stream ciphers

A streamed encryption method operates individually on each clear text bit using a transformation that varies depending on the place of the input bit. The Vermam cryptosystem is the prototype of these systems. it uses a very long secret key which should ideally represent a random sequence of bits. If we have a message m of n bits to encrypt, we consider the first n bits of the key which constitute a word K and we calculate the "or exclusive bit by bit" between the message and this part of the key. Thus, the part K of the key serves as a mask. The recipient who shares the same key extracts part K in the same way and then retrieves the plain text m by calculating m = c⊕K. The two interlocutors discard the part K used and can carry out a new transaction by doing the same with the rest of the key [20].

I-5.3.1.2 Block Ciphers

For any given key k, a block cipher specifies an encryption algorithm for computing the n-bit ciphertext for a given n-bit plaintext, together with a decryption to a given n-bit ciphertext." [41]. The distinction between block and stream ciphers is often misleading since block ciphers are used as well to encrypt streams of data. But instead of encrypting every single character after another, block ciphers encrypt a finite set of data from plaintext. In other words, block ciphers encrypt blocks of plaintext into blocks of ciphertext under the action of a secret symmetric key.

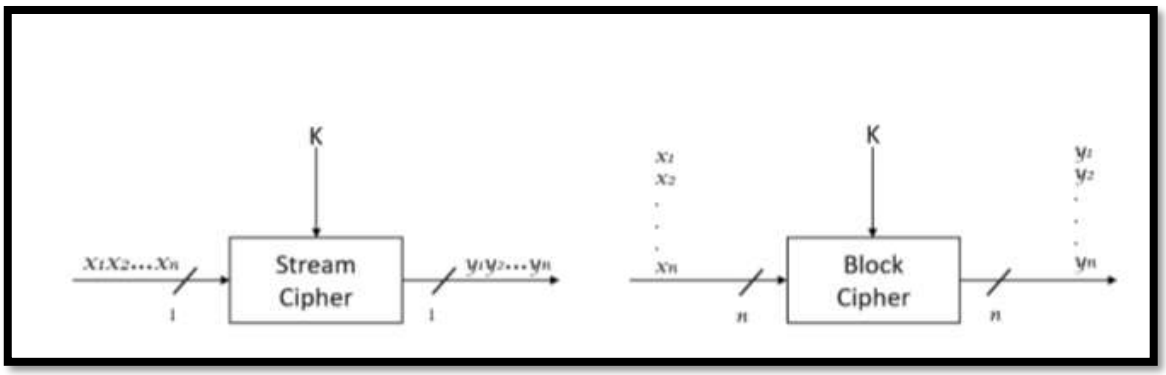


Figure I.9 Principles of encrypting n1 bits with stream and block ciphers.

I-5.3.1.3 Symmetric key Algorithms & Block Cipher

Symmetric key algorithms are those algorithms which share the same key for encryption and decryption purposes. There are allot of symmetric key algorithms proposed by different scientists of the world but I focus on the most widely used algorithms.

A. Data Encryptions Standard (DES)

(Data Encryption Standard), was the first encryption standard to be published by NIST (National Institute of Standards and Technology). It was designed by IBM based on their Lucifer cipher. DES became a standard in 1974 [44]. DES uses a 56 bit key, and maps 64 bit input block into a 64 bit output block. The key actually looks like a 64 bit quantity, but one bit in each of the 8 octets is used for odd parity on each octet. There are many attacks and methods recorded till now those exploit the weaknesses of DES, which made it an insecure block cipher.

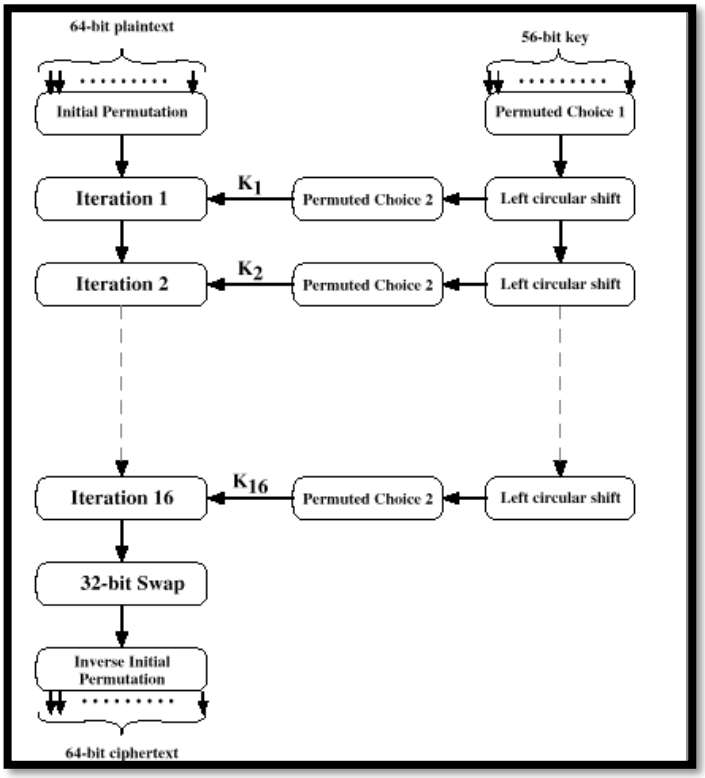


Figure I.10 DES Encryption Algorithm [47].

B. Advanced Encryption Standard (AES)

(Advanced Encryption Standard), also known as the Rijndael (pronounced as Rain Doll) algorithm, is a symmetric block cipher that can encrypt data blocks of 128 bits using symmetric keys 128, 192, or 256. AES was introduced to replace the DES. Brute force attack is the only effective attack known against this algorithm [7].

C. International Data Encryption Algorithm (IDEA)

International Data Encryption Algorithm (IDEA) is based on some impressive theoretical foundations and although cryptanalysis has made some progress against reduced-round

variants, the algorithm still seems strong. In my opinion, it is the best and most secure block algorithm available to the public at this time. IDEA is a block cipher and it operates on 64-bit plaintext blocks. The key is 128 bits long. The same algorithm is used for both encryption and decryption [47].

D. Blowfish

Blowfish is an algorithm designed by Bruce Schneire, planned for implementation on large microprocessors. He designed Blowfish to meet the following design criteria.

1. Fast. Blowfish encrypts data on 32-bit microprocessors at a rate of 26 clock cycles per byte.

2. Compact. Blowfish can run in less than 5K of memory.

3. Simple. Blowfish uses only simple operations: addition, XORs, and table lookups on 32-bit operands. Its design is easy to analyse which makes it resistant to implementation errors.

4. Variably Secure. Blowfish's key length is variable and can be as long as 448 bits.[47]

E. Ron's Code 4 (RC4)

RC4 a famous and widely used variable key-size stream cipher in symmetric encryption, developed by Ronald Linn Rivest in 1987. RC4 is used in Secure Socket Layer (SSL), Wi-Fi security protocol WEP, etc. It works like a finite automaton and consists of a Key-Scheduling Algorithm (KSA) and a Pseudo-Random Generation Algorithm (PRGA). KSA uses a key k, usually of the length $40 \leq l \leq 128$ bits and the key array K to generate a pseudorandom permutation S{0,1,2,...,255},also called S-Box. The S-Box is used as a nonlinear substitution operation to obscure the relationship between the key and the ciphertext. The PRGA uses the S-Box as the initial value of the internal state.

I-5.2 Asymmetric key Algorithms

The invention of Public-Key algorithms (also called asymmetric key algorithms) goes back to the year of 1979, where Whitfield Diffie and Martin Hellman proposed. This milestone paper introduced how two parties could communicate privately with two different keys using a public channel. The first key, the encryption key, is different and cannot be calculated (within a meaningful time) from the second key, the decryption key. In other words, it is not necessary to keep the Public-Key for encrypting a plaintext secret as long as the Private-Key for decrypting the ciphertext is confidential. With this approach, even a stranger could encrypt messages but still cannot decrypt a ciphertext without the corresponding Private-Key. By means of this approach, the key is only a secret of one person/entity and not a secret of a pair or group of entities like with symmetric key algorithms.[8] Generally the encryption of a plaintext P is done by using the Public-Key kpub (Enckpub(P) = C), whereas decryption is

realized with the associated Private-Key kpriv (Deckpriv(C) = P). The so-called one-way functions and trapdoor one-way permutations are applied for the proceeding of en/decryption. This approach additionally implies the advantage of applying digital signature schemes of message authentication and non-repudiation [10].The most common algorithms is :

I-5.2.1 Rivest, Adleman and Shamir (RSA)

RSA stands for Rivest, Adleman and Shamir, who devised this algorithm in 1977 at MIT. RSA is the most widely used public-key encryption scheme today. The US patent on the RSA algorithm expired in 2000, but as the algorithm was already published prior to patent application, it precluded patents elsewhere. RSA is secure because of the difficulty in factoring large integers that are a multiple of two prime numbers. Finding the product of two prime numbers is easy, but determining the original prime numbers from the total factoring is considered infeasible even with today's supercomputers. The generation of public and private keys is the most complex part of RSA cryptography. The following protocol displays the steps in RSA key generation:

1. Choose large prime numbers p and q.

2. Calculate n = p*q.

3. Caluclate φ(n) = (p-1)*(q-1).

4. Choose the exponent e ∈ {1,2,,(n)−1} such that gcd(e, φ(n))=1.

5. Compute private key d such that d*e=1 mod φ(n).

Where, Public key (Kpub) =(n,e) and Private key (Kpr) = d.

I-5.2.2 Elliptic curve cryptography (ECC)

takes advantage of the characteristics of a elliptic curve: an elliptic curve can be described as a finite set of points s(p) with p(x,y) over the binary Galois fields GF(2n) or over the prime Galois fields GF(prime) by (Jao, 2010):

$$y^2 = x^3 + a*x + b \quad a,b,x,y \in R .$$

The secret lies within the parameters a and b. It is almost impossible to reconstruct the elliptic curve, and therefore the implied set of points s(p), without the corresponding a and b parameters. Any small aberration from the original a and b results in a significantly diff erent elliptic curve [42].

I-5.3 Hash Algorithms

I-5.3.1 MD5 Algorithm

Initially created in 1991 by Ronald Rivest, MD5 is technically known as the Message-Digest Algorithm. As a hash function, An MD5 hash is created by taking a string of an any length and encoding it into a 128-bit fingerprint. Encoding the same string using the MD5 algorithm will always result in the same 128-bit hash output. MD5 hashes are commonly used with smaller strings when storing passwords, credit card numbers or other sensitive data in databases such as the popular MySQL. This tool provides a quick and easy way to encode an MD5 hash from a simple string of up to 256 characters in length. MD5 hashes are also used to ensure the data integrity of files. Because the MD5 hash algorithm always produces the same output for the same given input, users can compare a hash of the source file with a newly created hash of the destination file to check that it is intact and unmodified. [21]

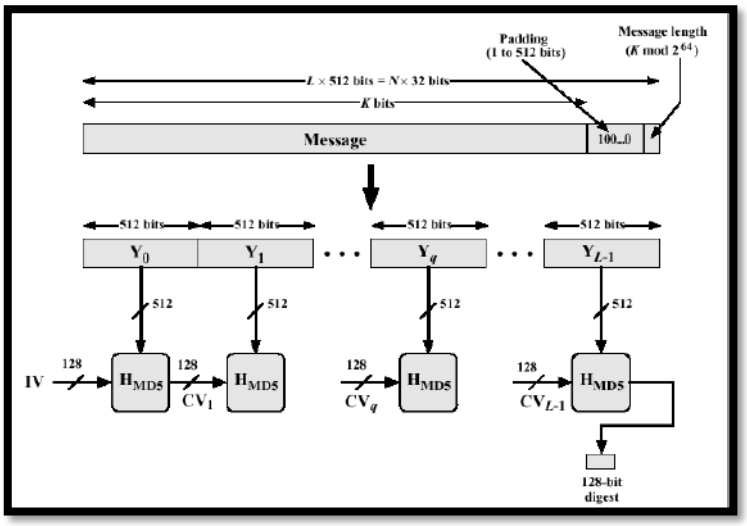


Figure I.11 MD5 Process [47].

I-5.3.2 Secure Hash Algorithm (SHA)

Secure Hash Algorithm is the algorithm used in Digital Signature Standard (SHS). NIST with NSA developed this as standard and it is to be used whenever a secure hash algorithm is required. The SHA is based on principles similar to MD4 message digest algorithm and is closely modeled after that algorithm. When a message of any length < 264 bits is input, the SHA produces a 160-bit output called a message digest. The message digest is then input to the DSA, which computes the signature for the message [47].

I-5.3.3 Digital Signature Algorithm (DSA)

NIST proposed the Digital Signature Algorithm (DSA) for use in Digital Signature Standard (DSS). This proposed standard specifies a public-key DSA appropriate for digital signature applications. The proposed DSS uses a public key to verify to a recipient the integrity of data and identity of the sender of the data [47].

I-6 cryptography using algorithm evolutionary computing methods

I-6.1-Overview of evolutionary algorithms

Evolutionary computation is another field, that is strongly inspired by. This field was pioneered independently in the 1960s by Fogel et al. 1966, Holland 1975, Rechenberg 1973. The latter two authors published their work in a widely accessible form only in the 1970s. Rechenberg used evolutionary strategies to develop highly optimized devices, such as irregularly shaped reduction pieces for pipes, e.g., for an air conditioning system, which proved to have a lower air flow resistance than ordinary reduction pieces.

In evolutionary computation, the process of natural evolution is used as a role model for a strategy for finding optimal or near optimal solutions for a given problem. In genetic algorithms, an important class of evolutionary computing techniques, candidates for a solution are encoded in a string, often a binary string only containing '0's and '1's. Evolution takes place by modifying the genetic code of a candidate.

Evolutionary techniques are generally applied to optimization problems. Many of those problems are combinatorial optimization problems, which are computationally hard [66]

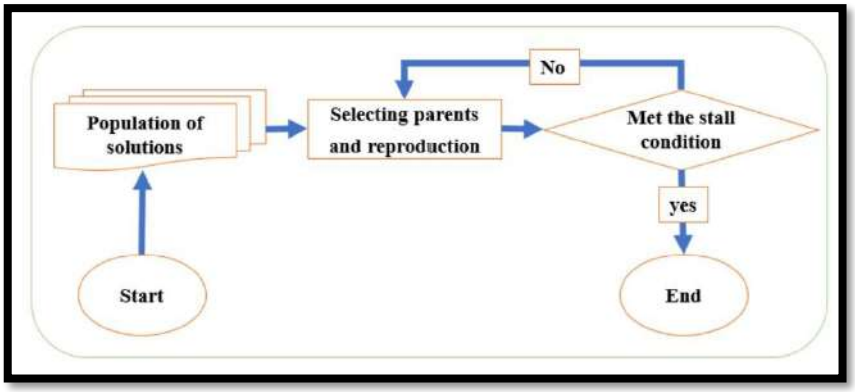


Figure I.12 General process of evolutionary algorithms [19].

The Soft Computing –history of development	
Soft Computing by Zadeh (SC)1981	
Evolutionary Computing (EC)	Rechenberg 1960
Neural Networks (NN)	McCulloch 1943
Luzy Logic (FL)	Zadeh 1965
Evolutionary Computing (EC)	
Genetic Programming (GP)	Koza 1992
Evolution Strategies (ES)	Rechenberg 1995
Evolutionary Programming (EP)	Fogel 1962
Genetic algorithms (GA)	Holland 1970

Table I.2 The History of Soft Computing Algorithms Development [63].

I-6.3 Evolutionary Computation Methods

Though evolutionary computation is not only optimization, most applications in Statistics are concerned, at least at present, with optimization problems. Schematically, an optimization problem may be defined as a pair (f ,Ω) where f is a function from Ω to the set of real numbers R, and Ω is the set of possible solutions. If Ω is a subset of Rn but is not a subset of Rn−1 , then n is the dimension of the problem. The aim may be to maximize or minimize the value of the objective function f . Solving the problem means finding the element(s) of Ω for which f attains its maximum (minimum) value, if they exist. In many lucky (or simplified) cases, f is a well-behaved mathematical function of n real arguments, and Ω is a sufficiently

regular subset of Rn so that the solution is found simply by equating to zero the derivatives off. But in the great majority of 204 F. [18]

I-6.4 Definition of some evolutionary algorithms

I-6.4.1 Genetic Algorithms

The genetic algorithm (GA) is one of the oldest and most known optimization techniques,

which are based on nature. In the GA, the search for solution space imitates the natural process which takes place in the environment, and the Darwinian theory of species evolution is taken into consideration. In GAs, we have a population of individuals; each, called a chromosome, represents a potential solution to the problem. The problem being solved is defined by the objective function. Depending on how ‘‘good’’ the given individual is fitted to the objective function, the value which represents its quality is attributed to it. This value is referred to as the fitness of the individual, and it is a main evaluating factor. Highly valued individuals have a better chance to be selected to the new generation of the population. In GAs, we have three operators:

- selection: a new population of individuals is created based on the fitness values of individuals from the previous generation.
- crossover: typically, parts of individuals are exchanged between two individuals selected to the crossover.
- mutation: the values of particular genes are changed randomly.
- Algorithm 1 presents the standard GA in the pseudo-code form (for more details see [64]) [7].

I-6.4.2 Genetic programming

Genetic programming (GP) is relatively new; it is a specialized form of a GA which operates on very specific types of solution, using modified genetic operators. The GP was developed by Koza as an attempt to find the way for the automatic generation of the program codes when the evaluation criteria for their proper operation is known. Because the searched solution is a program, the evolved potential solutions are coded in the form of trees instead of linear chromosomes (of bits or numbers) widespread in GAs. As GP differs from GA the used coding schema, Of course, the genetic operators are specialized for working on trees, e.g., crossover as exchanging the subtrees, mutation as a change of node or leaf. [7]

I-6.4.3 Differential evolution algorithm

Differential Evolution (DE) has been a competitive stochastic real parameter optimization algorithm since it was introduced in 1995. [48] is a type of evolutionary algorithm useful mainly for the function optimization in continuous search space. Although a version of DE algorithm for combinatorial problems has also been discussed the principal version of the DE algorithm was discussed by Storn and Price. The main advantages of DE over a traditional

GA are: It is easy to use, and it has efficient memory utilization, lower computational complexity (it scales better when handling large problems), and a lower computational effort (faster convergence) [7].

I-6.4.4 Evolution strategies

The evolution strategies (ESs) are different when compared to the GAs, mainly in the selection procedure. In the GA, the next generation is created from the parental population by choosing individuals depending on their fitness value, keeping a constant size of the population. In the ES, a temporary population is created; it has the different size than the parental population (depending on the assumed parameters k and l). In this step, the fitness values are not important. Individuals in the temporary population undergo crossover and mutations. From such populations, an assumed number of the best individuals are selected to the next generation of the population (in a deterministic way). ESs operate on the vectors of the floating point numbers, while the classical GA operates on binary vectors. . The primary types of ESs are ES(1 þ 1), ES(l þ kÞ, and ES(l; k) [7].

I-6.4.5 Evolutionary programming

Evolutionary programming (EP) was developed as a tool for discovering the grammar of the unknown language. However, EP became more popular when it was proposed as the numerical optimization technique. The EP is similar to the ES (l þ k), but with one essential difference. In EP, the new population of individuals is created by mutating every individual from the parental population, while in the ES (l þ k), every individual has the same probability to be selected to the temporary population on which the genetic operations are performed. In the EP, the mutation is based on the random perturbation of the values of the particular genes of the mutated individual. The newly created and the parental populations are the same sizes (l ¼ k). Finally, the new generation of the population is created using the ranking selection of the individuals from both, the parental and the mutated populations.[7]

I-6.4.6 Cartesian genetic programming

Cartesian genetic programming grew from a method of evolving digital circuits developed by Miller et al. in 1997. However, the term ‘Cartesian genetic programming’ first appeared in 1999 and was proposed as a general form of genetic programming in 2000 In CGP, programs are represented in the form of directed acyclic graphs. These graphs are represented as a two-dimensional grid of computational nodes. The genes that make up the genotype in CGP are integers that represent where a node gets its data, what operations the node performs on the data and where the output data required by the user is to be obtained. When the genotype is decoded, some nodes may be ignored.[30]

I-6.5 cryptography using algorithms evolutionary

I-6.5.1 Genetic algorithms in cryptography

While there have been many papers written on genetic algorithms and their application to various problems, there are relatively few papers that apply genetic algorithms to cryptanalysis. The earliest papers that use a genetic algorithm approach for a cryptanalysis problem were written in 1993, almost twenty years after the primary on genetic algorithms by John Holland. These papers focus on the simplest classical and modem ciphers, and are cited frequently by later works.

Figures 12 through 15 show which ciphers are attacked by which authors.

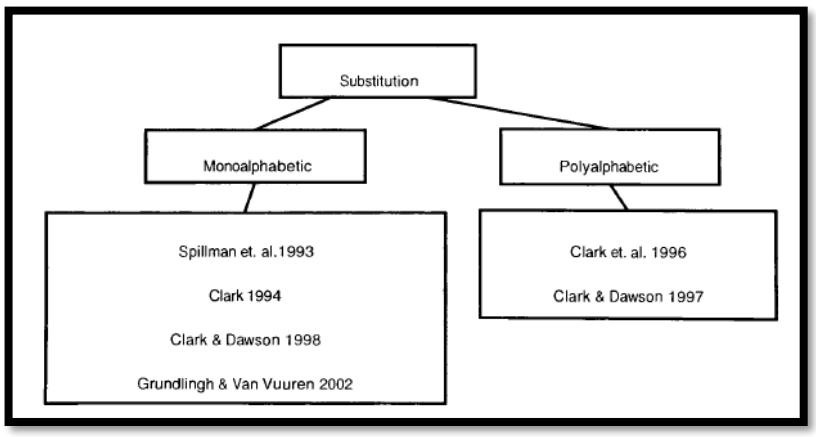


Figure I.13 Substitution Cipher Family and Attaking Papers.

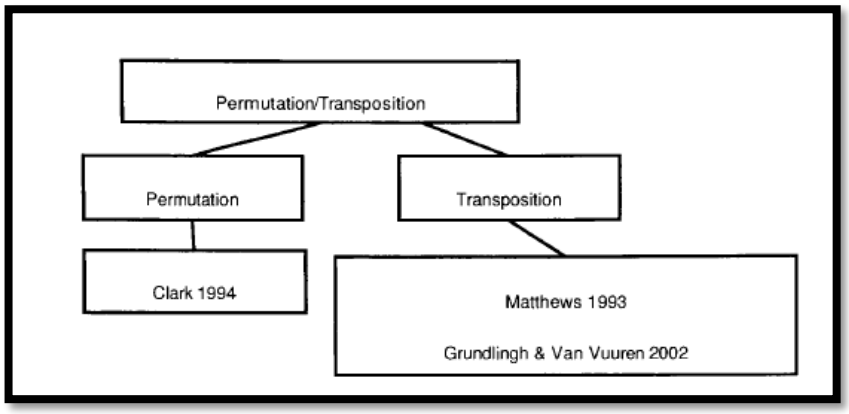


Figure I.14 Permutation Cipher Family an 1.

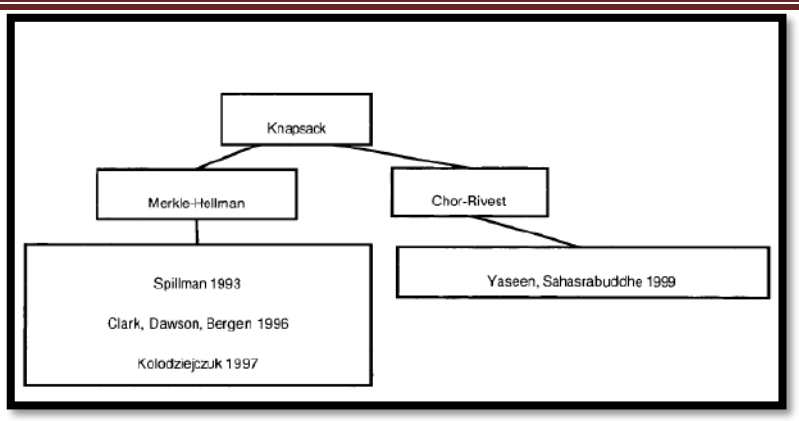


Figure I.15 Knapsack Cipher Family and Attacking Papers.

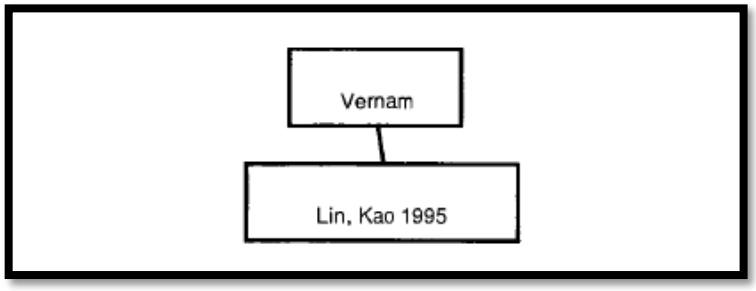


Figure I.16 Vernam Cipher Family and Attacking Papers.

A. Substitution : Spillman et al. - 1993

The first paper published in 1993 by Spillman, Janssen, Nelson and Kepner, focuses on the cryptanalysis of a simple substitution cipher using a genetic algorithm. This paper selects a genetic algorithm approach so that a directed random search of the key space can occur. The paper begins with a review of genetic algorithms, covering the genetic algorithm life cycle, and the systems needed to apply a genetic algorithm. The systems are considered to be a key representation, a mating scheme, and a mutation scheme. The selected systems are then discussed.[44]

B. Transposition : Matthews – 1993

The second paper published in 1993, by R. A. J. Matthews, uses an order-based genetic algorithm to attack a simple transposition cipher. The paper begins by discussing the archetypal genetic algorithm. This includes the typical genetic algorithm components, as well as the two characteristics a problem must have in order for it to be attackable by a genetic algorithm. The first characteristic is that a partially-correct chromosome must have a higher fitness than an incorrect chromosome.[44]

C. Merkle-Hellman Knapsack: Spillman - 1993

The third paper published in 1993, by R. Spillman applies a genetic algorithm approach to a Merkle-Hellman knapsack system. This paper considers the genetic algorithm as simply another possibility for the cryptanalysis of knapsack ciphers. A knapsack cipher is based on the NP-complete problem of knapsack packing, which is an application of the subset sum problem. The problem statement is: "Given n objects each with a known volume and a knapsack of fixed volume, is there a subset of the n objects which exactly fills the knapsack?". The idea behind the knapsack cipher is that finding the sum is hard, and this fact can be used to protect an encoding of information. There must also be a decoding algorithm that is relatively easy for the intended recipient to apply [44]

D. Substitution/Permutation : Clark - 1994

The only paper published in 1994, by Andrew Clark, includes the genetic algorithm as one of three optimization algorithms applied to cryptanalysis. The other two algorithms are tabu search and simulated annealing. The first section of the paper describes the two ciphers used - simple substitution and permutation. Next, suitability assessment is discussed. This section is where the fitness functions are developed.[44]

E. Vernam : Lin, Kao - 1995

This is the only paper published in 1995. By Feng-Tse Lin and Cheng-Yan Kao, is ciphertext-only attack on a Vernam cipher. The paper begins by introducing cryptography and cryptanalysis, including attack and cryptosystem types, as well as the Vernam cipher itself. The Vernam cipher is a one-time pad style system. Let M = m\,m2, ... ,mn denote a plaintext bit stream and K = ky, k2, . . . , kr a key bit stream. The Vernam cipher generates a ciphertext bit stream C = Ek(M) = cuc2, ,cn, where a = (m{ + ki) mod p and p is a base. Since this is a one-key (symmetric or private-key) cryptosystem,the decryption formula has the same key bit stream K, only M Dk(C) = my, m2, . . .,where m{ = (ct kt) mod p. The proposed approach is to determine K from an intercepted ciphertext C, and use it to break the cipher.[44]

F. Merkle-Hellman Knapsack: Clark, Dawson, Bergen - 1996

This paper, by Clark, Dawson, and Bergen is an extension of [54]. It contains a detailed analysis of the fitness function used in [54], as well as a modified version of the same fitness function. The paper begins by introducing the subset sum problem and previous work in genetic algorithm cryptanalysis. The fitness function from [54] is then discussed.[44]

G. Vigenere: Clark, Dawson, Nieuwland - 1996

This paper, by Clark, Dawson, and Nieuwland , is the first to use a parallel genetic al gorithm for cryptanalysis. It attacks a polyalphabetic substitution cipher. The first section introduces the simple and polyalphabetic substitution ciphers, while the second section in tro-

duces the genetic algorithm. The parallel approach chosen for this work is to have a number of serial genetic algorithms working on a separate part of the problem.[44]

H. Merkle-Hellman Knapsack: Kolodziejczyk - 1997

This paper published in 1997, is by Kolodziejczyk. It is an extension of [51]. It focuses on the Merkle-Hellman knapsack system, and the effect of initial parameters on the approach reported in [51]. The paper introduces the Merkle-Hellman knapsack, and then discusses the application of a genetic algorithm to the cryptanalysis thereof. Certain restrictions are placed on the encoding algorithm. These are:

- Only the ASCII code will be used in encryption
- The super increasing sequence will have 8 elements, ensuring that each character hasa unique encoding
- The plaintext is not more than 100 characters in length
- Due to these restrictions, the knapsack is a completely trivial problem.
- The genetic algorithm approach needs no discussion, as this application has no point.[44]

I. Substitution : Clark, Dawson - 1998

This is the only paper published in 1998. By Clark and Dawson compares three optimization algorithms applied to the cryptanalysis of a simple substitution cipher. These algorithms are simulated annealing, the genetic algorithm, and tabu search. Performance criteria, such as speed and efficiency, are investigated. The paper begins by introducing the application area and giving the basis for the current work. A simple substitution cipher is attacked by all three algorithms, with a new attack for tabu search, as compared to the attack done in [46]. Each of the attacks are compared on three criteria:

- The amount of known ciphertext available to the attack
- The number of keys considered before the correct solution was found
- The time required by the attack to determine the correct solution [44]

J. Chor-Rivest Knapsack: Yaseen, Sahasrabuddhe - 1999

This is the only paper from 1999. By Yaseen and Sahasrabuddhe is also the only paper that considers a genetic algorithm attack on the Chor-Rivest public key cryptosystem. The paper begins with a review of the Chor-Rivest scheme, followed by a review of genetic algorithms. The genetic algorithm attack is then discussed.[44]

K. Substitution/Transposition: Grundlingh, Van Vuuren- submitted 2002

This paper has not yet been published, but was submitted for publication in 2002. By Griindlingh and Van Vuuren combines operations research with cryptology and attacks two classical ciphers with a genetic algorithm approach. The two ciphers studied are substitution

and transposition. The paper begins with some historical background on the two fields, and then covers basic components of symmetric (private key) ciphers. The next two sections discuss genetic algorithms and letter frequencies in natural languages. The main difficulty in a genetic algorithm implementation is said to be the consideration of constraints.[44]

I-6.5.2 Related Work

- *Clark and Jacob* (*2000*) [25] experimented with two-stage optimization to generate Boolean functions. They use a combination of simulated annealing (SA) and hill climbing with a cost function motivated by Parseval theorem in order to find functions with high nonlinearity and low autocorrelation.
- *Clark et al. (2002)* [26] used simulated annealing to generate Boolean functions with cryptographically relevant properties. In their work, they consider balanced function with high nonlinearity and with the correlation immunity property less than or equal to two.
- *Kavut and Yucel (2003)* [52] developed improved cost functions for a search that combines SA and hill climbing. With that approach, the authors are able to find some functions of eight and nine inputs that have a combination of nonlinearity and autocorrelation values previously unattained. They also experiment with three-stage optimization method that combines SA and two hill climbing algorithms with different objectives.
- *Clark et al. (2003*) [24] experimented with SA in order to design Boolean functions by spectral inversion. They observe that many cryptographic properties of interest are defined in terms of Walsh-Hadamard transform values. Therefore, they work in the spectral domain where the cost function punishes those solutions that are not valid Boolean functions.
- *Burnett et al. (2004)* [35] presented two heuristic methods where the goal of the first method is to generate balanced Boolean functions with high nonlinearity and low autocorrelation. The second method aims at generating resilient functions with high nonlinearity
- *Millan et al. (2004)* [63] proposed a new adaptive strategy for local search algorithm for generation of Boolean functions with high nonlinearity. Additionally, they introduce the notion of the graph of affine equivalence classes of Boolean functions.
- *Burnett (2005)* [36] in her thesis used three heuristic techniques to evolve Boolean functions. The first method aimed to evolve balanced functions with high nonlinearity. The second method is used to find balanced Boolean functions with high nonlinearity that are correlation immune. The last method is used to find balanced functions with high nonlinearity and propagation characteristics different from zero. Furthermore, she experiments with the evolution of S-boxes.
- *Aguirre et al. (2007)* [21] used a multi-objective random bit climber to search for balanced Boolean functions of size up to eight inputs that have high nonlinearity. Results indicate that the multi-objective approach is highly efficient when generating Boolean functions that have high nonlinearity.
- *Izbenko et al. (2008)* [65] used a modified hill climbing algorithm to transform bent functions to balanced Boolean functions with high nonlinearity.

- *McLaughlin and Clark (2013)* [30] on the other hand used SA to generate Boolean functions that have optimal values of algebraic immunity, fast algebraic resistance and algebraic degree. In their work, they experiment with Boolean functions of sizes up to 16 inputs.
- *Picek et al. (2013)* [57] experimented with GA and GP to find Boolean functions that possess several optimal properties. As far as the authors know, this is the first application of GP to the evolution of cryptographically suitable Boolean functions.
- *Hrbacek and Dvorak (2014)* [48] experimented with CGP to evolve bent Boolean functions of size up to 16 inputs. The authors investigate several configurations of algorithms in order to speed up the evolution process. Since they do not limit the number of generations, they succeed in finding bent function in each run for sizes between 6 and 16 inputs. Several EAs are used by Picek et al. (2014) [59] to evolve Boolean functions that have better side-channel resistance. This paper presents the first application of optimization techniques to Boolean functions with improved side-channel resistance. With the goal of finding maximal nonlinearity values of Boolean functions
- *Picek et al. (2014)* [58] experimented with several EAs. Furthermore, they combine optimization techniques with algebraic constructions in order to improve the search. Although they are unable to find a balanced Boolean function with nonlinearity equal to 118, they present several possible avenues to follow when looking for highly nonlinear balanced Boolean functions.
- Picek et al. (2015) [55] compared the effectiveness of CGP and GP approach when looking for highly nonlinear balanced Boolean functions of eight inputs.
- *Picek et al. (2015)* [53] investigated several EAs in order to evolve Boolean functions with different values of the correlation immunity property. In the same paper, the authors also discuss the problem of finding correlation immune functions with minimal Hamming weight, but they experiment with only one size of Boolean functions. More extensive investigation on finding correlation immune Boolean functions with minimal Hamming weight and different sizes is conducted by Picek et al. (2015).[57]
- *Mariot and Leporati (2015) [*38] used Particle Swarm Optimization (PSO) to find Boolean functions with good trade-offs of cryptographic properties for dimensions up to 12. The same authors use GAs where the genotype consists of the Walsh-Hadamard values in order to evolve semibent (plateaued) Boolean functions (Mariot and Leporati, 2015).[39]
- *Picek et al. (2016)* [56] conducted a detailed analysis of the efficiency of a number of evolutionary algorithms and fitness functions for Boolean functions with 8 inputs.[37]

I-7 Conclusion

If you want to find information about “what is cryptography”, there are lot of data about this topic. But you try to understand the concept of advanced coding that uses arithmetic algorithms, this is not an easy task. Cryptography is not just one topic; It's a very broad topic that a lot of research has been done on, and this research is endless with new ideas popping up from time to time.

Security problem is not discovered today. It is a very old concept and because computer technology is changing very quickly, so the importance of security is required more than other technologies in the computer world. Crypto research is not the buzzword today; It is one of the oldest fields from about 4,000 years ago. People, especially the ancient Egyptian military, started working on it and today at the end of 2020, students, teachers and scholars are still doing research on it all over the world. Conclusion of the following key points related to coding.

- Cryptography has a very ancient history, 4000 years ago. People try to make safe connections with collaborators because they also keep their secrets from their enemies.
- The need for encryption is to secure reliable, authenticated information that is available whenever the user wants it.
- Whenever a secret technology is revealed, its secret will be dismantled at some point. So, the secure connection issue is never resolved unless you have to update yourself and your systems with the latest security tools.
- After the invention of computers and especially the computer network system, the importance of encryption increased more and more and hacking, viruses, cracking, etc. became a nuisance for computers.
- For many years, computers were used by the US military (computer networks), but now they are more common in everyday life. After the invention of the Internet, its use increased day by day. Many large institutions, especially banks, companies and financial sectors are associated with it.

CHAPTER 2: *DIRECT IMAGE PROJECTION USING GENETIC ALGORITHM (DIP-GA)*

II-1 Introduction

The increase of the defects, gaps and problems in coding processes and their complexity and the inability of traditional software solutions to comprehend them led the researchers to delve into finding efficient algorithms that help in finding suitable and ideal solutions to complex issues and in the speed of accessing, storing and retrieving solutions, then reaching a coherent structure for a smart software structure and one of these efficient algorithms.The genetic algorithm is one of the algorithms of computational development and it is one of the modern methods, as the importance of using this method has emerged in solving large-sized complex problems that have a huge number of alternative solutions, and the genetic algorithm is better than the traditional artificial intelligence techniques because the latter is sophisticated and tends to fall into regions with small local endings. In addition to that , the genetic algorithm does not fail to find a solution with a change in the entries. It is also better in the search process within the large, multi-space domain.

Shapes and the solution resulting from the application of the genetic application of the genetic algorithm is most often a close to one solution the ideal, and when applied, this method provides an intelligent search between a vast number of alternative plans [53].

In this chapter, we will explain the concept, working methods, and the advantages and disadvantages of all the algorithms we used in the image data file encrypting process. Among them is the previously applied algorithm represented by RC4 and the LSB algorithm that showed its weakness. In order to improve the LSB algorithm and solve its problems, we proposed an algorithm based on the projection method, and in order to improve its results, we presented the hybrid algorithm for the projection method and the genetic algorithm in two ways, the first without calculating the fitness and the second by calculating it.

II-2 Cyclic Redundancy Check

A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents. On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption. CRCs can be used for error correction (see bit filters). CRCs are so called because the check (data verification) value is a redundancy (it expands the message without adding information) and the algorithm is based on cyclic codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyse mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the check value has a fixed length, the function that generates it is occasionally used as a hash function.

The CRC was invented by W. Wesley Peterson in 1961; the 32-bit CRC function, used in Ethernet and many other standards, is the work of several researchers and was published in 1975.When to Use a CRC. The default CRC component can be used as a checksum to detect alteration of data during transmission or storage. CRCs are popular because they are simple to implement in binary hardware, are easy to analyze mathematically, and are particularly good at detecting common errors caused by noise in transmission channels [24].

Algorithm CRC32

Input : data: Byte

Output : crc32 : UInt32

Begin

crc32←0xFFFFFFFF;

for each byte **in** data **do**

nLookupIndex ← (crc32 xor byte) and 0xFF;

crc32 ← (crc32 shr 8) xor CRC_Table[n_Lookup_Index];

crc32 ← crc32 xor 0xFFFFFFFF;

End.

*Note***:** In all the encryption methods we implemented, we calculated a cyclic redundancy check 32(crc32) before starting the encryption process and accompanied us with the encoded image to allow the user to make sure that the image he decoded is the same as the original image.

In this section, we will re-apply previously known algorithms:

II-3.1 RC4 algorithm (Rivest Cipher 4" or "Ron's Code 4)

RC4 stream cipher is used to protect internet traffic as part of the SSL (Secure Socket Layer) and TLS (Transport Layer Security) protocols, and to protect wireless networks as part of the WEP (Wired Equivalent Privacy) and WPA (Wi-Fi Protected Access) protocols. This attack was described by Fluhrer, Mantin and Shamir.

II-3.1.1. Overview of RC4

In this paper, the RC4 encryption is characterize and executed. The RC4 is an abbreviation of "Rivest Cipher 4" or "Ron's Code 4". It uses a variable key length which can range between 1 to 256 bytes (8 to 2048 bits) and is utilized to instate a 256-byte state vector S. The key stream is totally independent of the used plaintext. It uses a variable length key from 1 to 256 bit to initialize a 256-bit state table. The state table is used for subsequent generation of pseudo-random bits that is XORed with plaintext to produce the cipher text as shown in by applying the same method we again decrypt the encrypted image. After the end of this step we again got the original image back In RC4 encryption algorithm, the encryption process including two Algorithms, Key Scheduling Algorithm (KSA) and Pseudo Random Generation Algorithm (PRGA) to produce the key stream of the stream cipher [44].

II-3.1.2 Encryption algorithm

We will mention the implementation stages, then show you the execution algorithm, and in the end we will show you diagram of encryption algorithm

A. RC4 Implementation

The steps for RC4 encryption algorithm are as follows :

1. Loading an image.

2. Determining the size of image (height and width).

3. We divide the pixels of the image into blocks and the length of the block is equal to the length of the image, meaning we take an entire line and consider it a block.

4. We change the pixel locations in every block in between.

5. We apply the encryption process using the RC4 algorithm for each block

6. after that We return the pixels to their location in the image.

B. Algorithm

The RC4 encryption algorithm are as follows

Algorithm RC4_Enc

Input: img: Image,key: String;

Output: img: Image;

Begin

//aij is the value of the pixel at the coordinates (i, j) in img.

m: Integer; m ← Height_img;

n: Integer;n ← Width_img;

i,j: Integer;

c,p: String;

c ← ””;

p ← ””;

K ← Initiali_key(key);

For i ← 0 to m do

For j ← n to 0 do

p ← p+ aij;

End for

For j ← 0 to n*24 do

c ← c+ p [j]⊕ K[j mod 256];

End for

End for

Return the pixels c in img;

End.

C. DIAGRAM

The RC4 encrypted diagram is as follows:

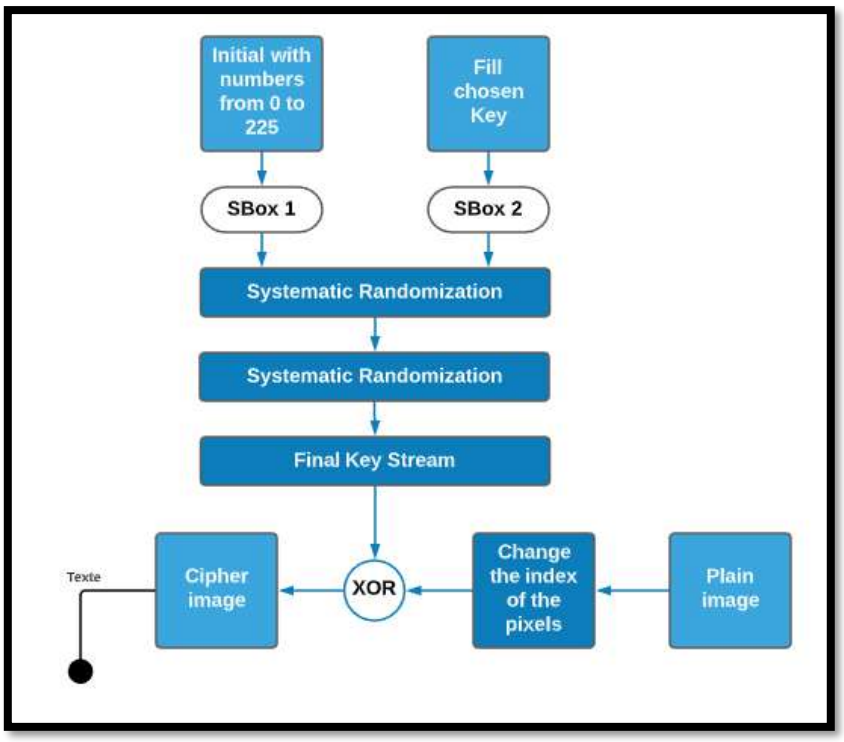


Figure II.1: Diagram of RC4 Encryption Algorithm.

II-3.1.3 Decryption Algorithm

We will mention the implementation stages, then show you the execution algorithm, and in the end we will show you diagram of decryption algorithm

A. Implementation steps

The steps for RC4 decryption algorithm are as follows :

1. Loading an image.

2. Determining the size of image (height and width).

3. We divide the pixels of the image into blocks and the length of the block is equal to the length of the image, meaning we take an entire line and consider it a block.

4. We apply the decryption process using the RC4 algorithm for each block

5. after that We return the pixels to their location in the block, then in the image.

B. Algorithm

The RC4 decryption algorithm are as follows

Algorithm RC4_Dec

Input: img: Image,key: String;

Output: img: Image;

Begin

//aij is the value of the pixel at the coordinates (i, j) in img.

m: Integer; m ← Height_img;

n: Integer;n ← Width_img;

i,j: Integer;

c,p,p1: String;

c ← ””; p ← ””; p1 ← ””;

K ← Initiali_key(key);

For i ← 0 to m do

For j ← 0 to n do

c ← c+ aij ;

End for

For j ← 0 to n*24 do

P1 = p1+ c [j]⊕ Kk[j mod 256];

End for

For j ← n to 0 do

p = p+p1[j];

End for

End for

Return the pixels p in img;

End.

C. Diagram

The steps for RC4 decryption diagram are as follows

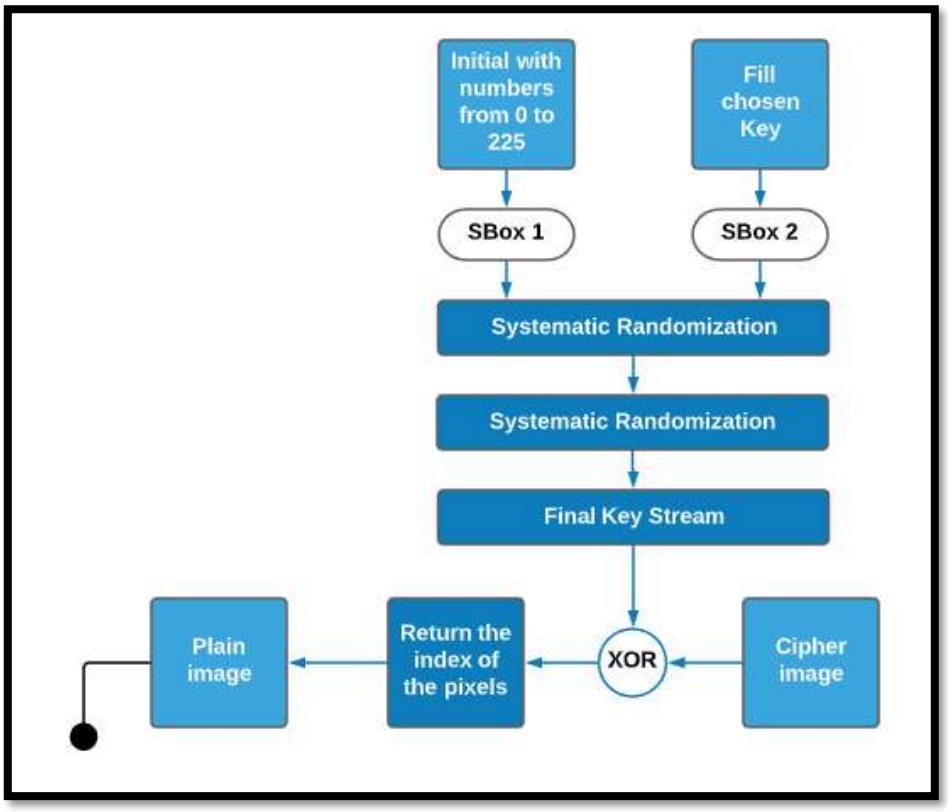


Figure II.2. Diagram of RC4 Decryption Algorithm.

II-3.1.4 RC4 Strengths

RC4 have many advantages over the other algorithms, in this section we will mention some of RC4 Strengths [80]:

1) The difficulty of knowing which location in the table is used to select each value in the sequence.

2) A particular RC4 key can be used only once.

3) Encryption is about 10 times faster than DES.

II-3.1.5 RC4 Weaknesses

Despite RC4 algorithms have many strengths points but also it has some of other weaknesses, we mention:

1) The RC4 algorithm is vulnerable to analytic attacks of the state table.

2) Weak Keys: these are keys identified by cryptanalysis that is able to find circumstances under which one or more generated bytes are strongly correlated with small subset of the key bytes. These keys can happen in one out of 256 keys generated [42].

II-3.2 Least Significant Bit Algorithm (LSB)

It is a method used in Steganography **b**ut we'll apply it to the encryption process and we will try to explain it

II-3.2.1 Overview of LSB algorithm

Least Significant Bit (LSB) insertion is one of the most popular techniques in the spatial-domain category. It is a common and simple approach to embed a secret message in a host image. The LSB technique works by using the least significant bits of each pixel in one image to hide the most significant bits of another changing the LSB of a pixel will cause some small changes in pixel intensity, however, these changes cannot be identified by the human eye. In the receiving stage, the data will be extracted and decompressed. The cipher text is then de-crypted to reveal the embedded message [6]

II-3.2.2-Encryption algorithm

We will mention the implementation stages, then show you the execution algorithm

A. Implementation steps

The steps for LSB encryption algorithm are as follows :

1. Loading an image.

2. Extracting the RGB components and store them it to table A.

3. Order the bits the pixels the table A according to the following condition, from the weakest byte to the strongest byte.

4. Hide the image bites with the bites that we arranged before, so that I start with the strong bites of the image until it runs out, and move to the weaker bits and so on until we are done.

5-We return each byte to its location in the pixel, then the image.

B. Algorithm

The LSB encryption algorithm are as follows

Algorithm LSB_Enc

Input: img: Image;

Output: img: Image;

Begin

//aij is the value of the pixel at the coordinates (i, j) in img.

m: Integer; m ← Height_img;

n: Integer;n ← Width_img;

```
i,j,ii,jj: Integer;i ← 0;j ← 0;
For jj ← 0 to 8 do
   For ii ← 0 to m*n do
      p← aij;
      a← p[f1]; i← i+1;
      Tb[ii] ← a+Tb[ii];
      p← aij;
      a← p[f1]; i← i+1;
      Tg[ii] ← a+Tg[ii];
      p← aij;
      a← p[f1]; i← i+1;
      Tr[ii] ← a+Tr[ii];
   End for
End for
For jj ← 0 to m*n do
   T[jj] ← Tr[jj]+ Tg[jj]+ Tb[jj];
End for
Return the pixels c in img;
End.
```

II-3.2.3. Decryption Algorithm

We will mention the implementation stages, then show you the execution algorithm.

A. Implementation steps

The steps for LSB decryption algorithm are as follows

1. Loading an image.

2. Extracting the RGB components and store them it to table A.

3. Inverting the crypto process, we return each bite to its pixel location and then in the image.

B. algorithm

The decryption algorithms of LSB method is the inverse of the encryption LSB method that is presented on the last section.

C. Diagram

The diagram for LSB encryption and decryption algorithm are as follows

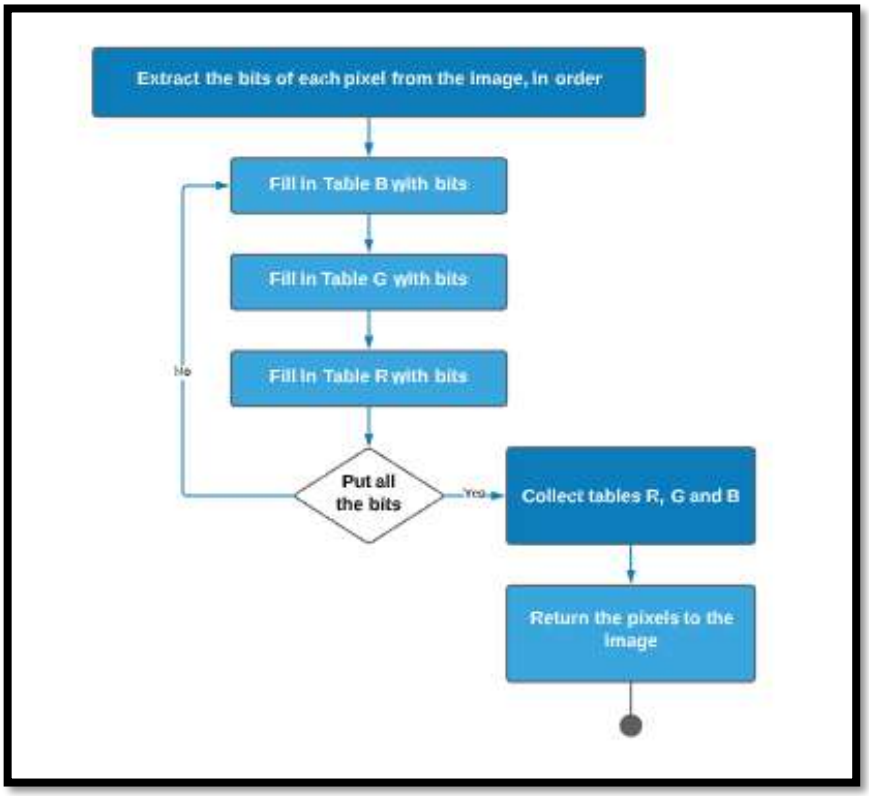


Figure II.3. Diagram of Algorithm LSB.

II-3.2.4 LSB Strengths

LSB have many advantages , in this section we will mention some of LSB Strengths ,

1. Its advantages are that it achieves total image distortion in most cases.

2. Simplicity in its implementation

3. The histogram distribution of the encoded image is significantly different from the histograms of the original images. So, if any statistical attackers did attack, they would not be able to breach the suggested security.

II-3.2.5 LSB Weaknesses

Despite LSB algorithms have many strengths points but also it has of other weaknesses, we mention:

1. We noticed that the method of working with LSB is simple. If any programmer discovered the used encryption method, he will easily decode it.

2. Although LSB embedding methods mask data in a way that humans are not aware of, such schemas can easily be destroyed by the opponent such as using lossy compression or filtering algorithms. Any process that modifies the values of some pixels, directly or indirectly, may degrade the quality of the original object.

II-4 *First contribution: Direct Image Projection (DIP)*

It is a new technique, and for the first time it is used in the coding process, we will try to explain a method and then implement it with mentioning the algorithms

II-4.1 Encryption Using the Direct Image Projection Method

In this part we will first explain the concept of the method, then mention the stages of its implementation with the algorithm,

II-4.1.1 DIP Formulation

In our algorithm, we have proposed a new idea of substitution of pixels in the image, the purpose of which is to hide them. It is based on the projection of each pixel of the original image onto another virtual image which appears as a rotated image. This rotation is explained by a virtual observer which is supposed to revolve around the center of the original image. Thus, the relationship between these two images is ensured by the fact that the plane of the virtual image is perpendicular to the viewer's direction. Thus, and because each pixel has a ray of light, the position of intersection of its direction with that of the virtual image is used as the projection position. However, this projection has a disadvantage that some parts of the original image disappear, while others are projected in the same positions. The reason is that the positions of the projected pixels should be rounded to the nearest integers.[6]

II-4.1.2 Direct Projection Positions

It is assumed that the original image is viewed from a position on a direction perpendicular to its center, while its projection is viewed from a tilted position. First, the different positions of the viewer are assumed to be located on a circle around the center of the image. The radius of this circle is what defines the distance from the center of the image. The reference position is then that where the direction of the observer is perpendicular to the plane of the original image. Consequently, any other position is defined by its angle of inclination relative to the reference position. .[6]

The projection process is based on the intersection of the direction of the light ray towards the viewer of each pixel of the original image and the virtual image whose plane is perpendicular to the viewer's current direction. The point of intersection is the position to use to paste the original pixel there. Also, it is obvious that the new position is slightly different from the original one. Thus, when all the pixels are projected and depending on the tilt, the resulting image is either extended or condensed compared to the original (Figure II.4). .[6]

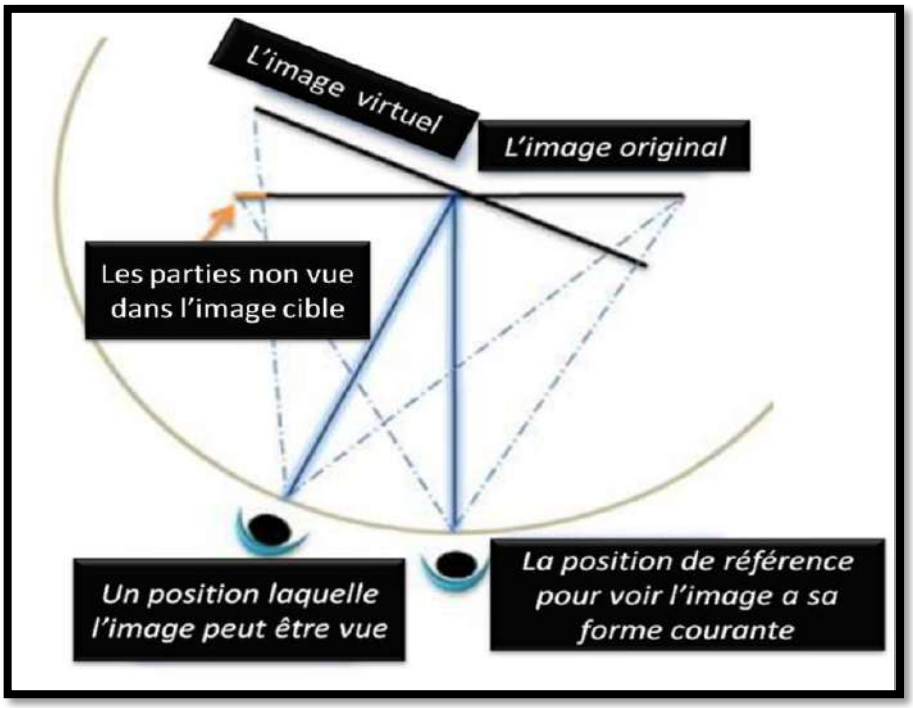


Figure II.4. System for projecting an image.

Consider the pixel located at Po from the center O of the image and seen from the reference position Vo, When the observer moves around the center of the image along the circle having its radius equal to | OVo | →, the position that the pixel Po will take in the target image becomes Pt. In other words, Pt is the position where the pixel Po is supposed to be seen in the target image when tilting the position of the observer. So the objective is to determine the value of OPt.

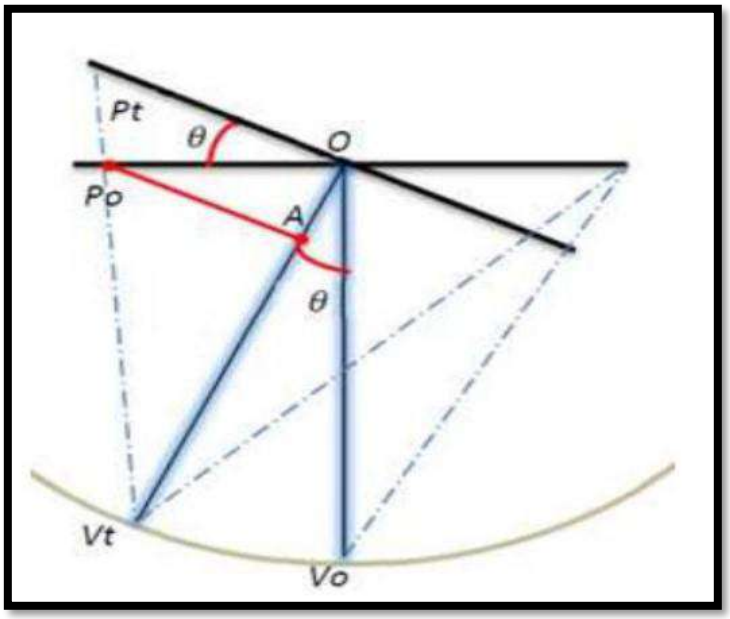


Figure II.5. Estimation of the position of the projection of a pixel.

Let θ be the angle between the directions of the reference position and the new position. This angle is the same between the plane of the original image and the tilted one. Let A be the perpendicular projection of Po on the new direction | OVt | →, such that ΔPoAO is a right triangle and OPoA = θ. It is easy to see that APo = OPo * cosθ. If we take ΔOPtVt into account, using this expression and based on the truth of the following expression:

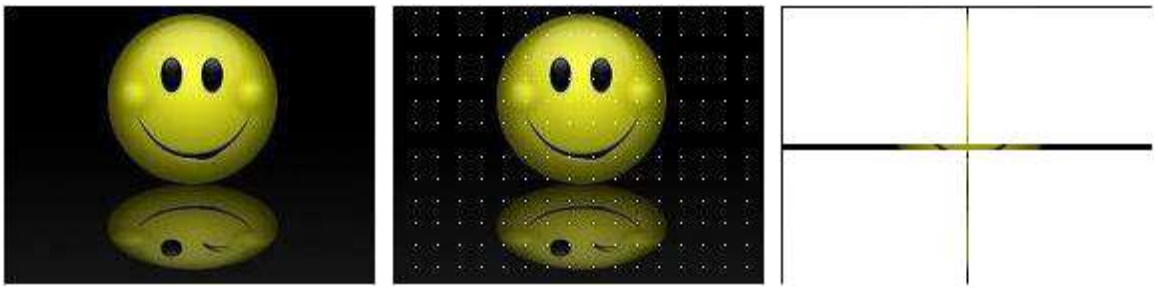
$$OPtAPo = OVtAVt \tag{1}$$

The value of OPT can be easily calculated from the resulting expression (AVt = OVt – OA) :

$$OPt = OVt * OPo * \cos\theta OVt - OPo * \sin\theta \tag{2}$$

According to figure (II.5), formula (2) is only true for a part which is half of the image located on the same side of the viewer's position with respect to the original. In fact, the other part of the image is projected onto the target image using different formulas.

As the expected results in the substitution algorithm depend neither on the complexity of the formulas, nor on their number, only formula (2) was used. Thus, it is used simultaneously on both sides of the image. Also, instead of using each pixel alone, the construction of the new virtual image uses a whole line or column.



(a) (b) (c)

Figure II.6. (a) original image, (b) and (c) positions projected from (a).

When constructing the positions of the pixels two problems arise. The first is that of the pixels having their projection positions outside the limits of the virtual image. For example, in Figure II.6 in image (c) we see positions or holes that are not white. This means that they are not affected by the projection.

The second problem arises and is due to the fact that each estimated position of OPt is rounded to the integer closest to its value, which makes it possible to obtain several rows or columns of pixels having the same projection position. This takes place in the case of using large angles which condense a whole part of the original image in a reduced part of that projected. So, for a given position, we only keep a single row point or column point and the others are ignored.

The projection algorithm repeats the same operation on both sides of the image. This means that the projection is applied in the other side as if the virtual observer is moved to the left side and then to the right side.

The idea of our algorithm does not require the recovery of lost positions, so these two problems do not need to be corrected. But despite all this, it's very easy to spot the difference in using small or large angles.

Algorithm DIP-projection

Input: ((height/ width) r, (angle) θ, (distance) d)

Output: opt

Begin

d=d*r;

For i ← 0 to r do

T ← d * i cos(θ) / d - (i * sin(θ));

If (T > 0 and T < r and opt [a-1] != T) then

Opt[a]=T;

a=a+1;

End if

End for

End.

II-4.1.3 Encryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for DIP encryption algorithm

A. Implementation steps

The steps for DIP encryption algorithm are as follows:

1. Loading an image.

2. Determining the size of image (height and width).

3. We enter the angle of rotation and the distance you want.

4. We extract the location that we will hide in it using the OPT function.

4.1. first, we use OPT (width, angel, distance) that allowed us to discover the hiding location according to the width and we put them on a table (I).

4.2. second, we use OPT (height, angel, distance) that allowed us to discover the hiding location according to the height and we put them on a table (J).

4.3. Be matrix of the pixels that are in the locations you specified OPT function Matrix (J [], I []).

5. The encryption process use the LSB method we explained before but the process of concealment is in the locations that we have defined Matrix (J [], I []).

6. although using this method the image will not encrypted completely because there is no place to hide the remaining bits and to solve this problem we hide these points in the locations that are not used by OPT function.

7. After hiding all the image, we return each pixel to its location in the image.

B. Algorithm

The DIP encryption algorithm are as follows:

Algorithm Proj_Enc

Input: img: Image,d1: Real,d2: Real,o1: Real,o2: Real;

Output: img: Image;

Begin

//aij is the value of the pixel at the coordinates (i, j) in img.

m: Integer; m ← Height_img;

n: Integer;n ← Width_img;

J ← OPT(d1,o1,m);

I ← OPT(d2,o2,n);

ij ← J_length * I_length;

i,j,ii,jj: Integer;i ← 0;j ← 0;

For jj ← 0 to 8 do

For ii ← 0 to ij do

p← aij;

a← p[f1]; i← i+1;

Tb[ii] ← a+Tb[ii];

p← aij;

a← p[f1]; i← i+1;

Tg[ii] ← a+Tg[ii];

p← aij;

a← p[f1]; i← i+1;

Tr[ii] ← a+Tr[ii];

End for

End for

For jj ← 0 to m*n do

T[jj] ← Tr[jj]+ Tg[jj]+ Tb[jj];

End for

We put the rest of the bits in the table T1 in order;

Table T values are placed in the pixels of coordinates (*i*, *j*) in the image corresponding to

Tables I and J ;

Table T1 values in the remaining pixels;

End.

C. Diagram

The diagram for DIP encryption algorithm are as follows:

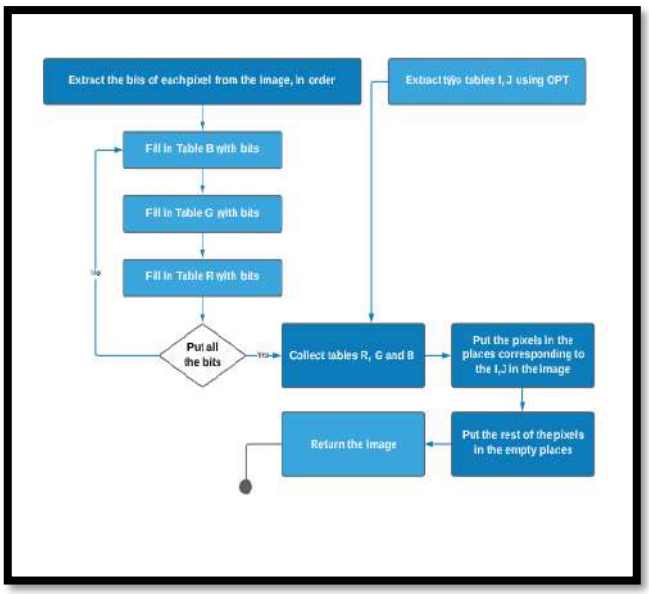


Figure II.7. Diagram of Algorithm encryption by DIP

II-4.1.4 Decryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for DIP decryption algorithm

A. Implementation steps

The steps for DIP decryption algorithm are as follows:

1. Loading an image.

2. Determining the size of image (height and width).

3. We enter the angle of rotation and the distance you want.

4. we extract the location that we will hide in it using the OPT function.

4.1. First, we use OPT (width, angel, distance) that allowed us to discover the hiding location according to the width and we put them on a table (I).

4.2. Second, we use OPT (height, angel, distance) that allowed us to discover the hiding location according to the height and we put them on a table (J).

4.3. Be matrix of the pixels that are in the locations you specified OPT function Matrix (J [], I []).

5. The decryption process use the LSB method we explained before but the process of concealment is in the locations that we have defined Matrix (J [], I []).

6-Retrieve the bytes from the locations that are not used by OPT function.

7. After we restore each byte to its original location in the pixels, we return each pixel to its location in the image.

B. Algorithm

As for Decryption we do the reverse operation;

B. Diagram

C. The diagram for DIP decryption algorithm are as follows:

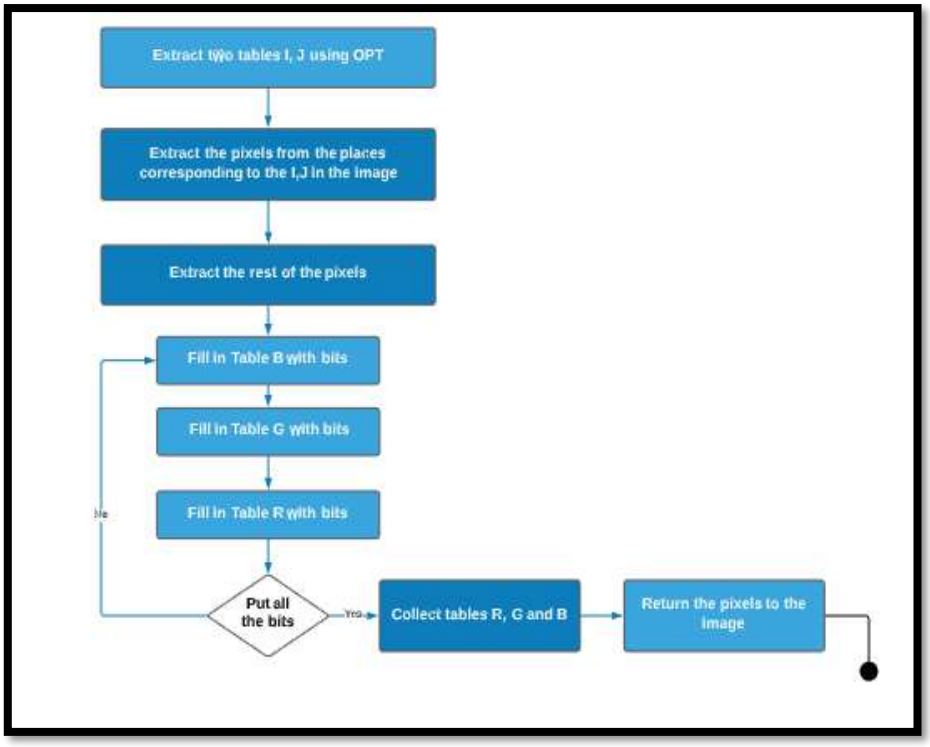


Figure II.8. Diagram of Algorithm by projection method.

II-4.1.5 DIP Strengths

DIP have many advantages over the other algorithms, in this section we will mention some of DIP Strengths :

1. One of its advantages is that it achieves complete image distortion in all cases

2. Simplicity in its implementation

3. The histogram distribution of the encoded image is significantly different from the histograms of the original images. So, if any statistical attackers did attack, they would not be able to breach the suggested security.

II-4.1.6 DIP weaknesses

Despite DIP algorithms have many strengths points but also it has some of other weaknesses, we mention

1. If the angle of rotation used and the distance are not appropriate, then its result is the same as the result of encryption using the LSB algorithm.

2. Any process that modifies the values of some pixels, either directly or indirectly, may result in degrading of the quality of the original object, less robust, the hidden data can be lost with image manipulation (image compression, crop, resize etc.).

II-4.2 *Second Contribution: Genetic Algorithm (GA)*

In our work we have implemented the basis of crossover and shifting features at the pixel level. We did not implement them at the bit level, so we did not show the best results for them.

II-4.2.1 Applications of Genetic Algorithms

There are many applications of genetic algorithms in different fields. These applications are concerned with problems which are hard to be solved but have easily verifiable solutions. Another common trait, which belongs to these applications, is the equation style of fitness function. Cryptography and cryptanalysis could be considered to meet these criteria. However, cryptology is not closely related to the typical GA application areas and, subsequently, fitness equations are difficult to generate. This makes the use of a genetic al15gorithms approach to cryptology rather unusual. Genetic algorithms find application in bioinformatics,

phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics, pharmacometrics, and other fields such as image enhancement, image processing, and image cryptography. [63]

II-4.2.2. Information Security

The application of genetic algorithms (GAs) to the field of cryptology is rather unique. Few works exist on this topic. Genetic algorithms are evolutionary algorithms based on the notion of natural selection. The genetic algorithms have been proven to be reliable and powerful optimization technique in a wide variety of applications. It can be applied to both texts and images. Genetic algorithms are secure since they do not directly utilize the natural numbers. The results obtained for generating keys using genetic algorithms should be good in terms of coefficient of autocorrelation. Generally, genetic algorithms have two basic functions, namely crossover and mutation [63].

II-4.2.3. Using Crossover and Mutation Operators of GAs

In this approach, a new image security algorithm has been proposed using the concept of genetic algorithms based on cross-over and mutation features. This algorithm provides safer image encryption, minimal data loss, maximum speed and maximum distortion in encryption, while adding an additional degree of protection [63].

II-4.2.4 Encryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for GAs encryption algorithm

A. Implementation steps

The steps for GAs encryption algorithm are as follows

1.Loading an image.

2. Determining the size of image (height and width).

3. Dividing the image into the chromosomes, but the length of the chromosome is according to the following conditions:

3-1 If the image size is less than (500*500) dividing the pixels into a set of chromosomes, each block size is (16 pixel).

3-2 If the image size is larger than (500) dividing the pixels into a set of chromosomes, each chromosomes size is (64 pixel).

4. Extraction and storage of RGB components. Chromosome table and consider it as the primary chromosome table.

5. Doing crossover operator is like the first type (Single Point Crossover), We choose as two specific symbols and Crossover proceeds in there are two possibilities:

5.1. If the session number is a pair: We merge the first half of the first chromosome with the second text of the second chromosome, and the first half of the second chromosome with the second half of the first chromosome.

5.2. If the session number is impair, then we combine the second third of the first chromosome with the other two thirds of the second chromosome. The second third of the second chromosome with the other two thirds of the first chromosome

6. We save new children in a new table.

7. Doing mutation operator:

The mutation currency is applied to 3 bits in a chromosome(children)

Bit 1: P=4 and B=12

Bit 2: P=3 and B=6

Bit 2: P=2 and B=18

P: pixel on which we will be booming

B: The bit of the mutation, if it is equal to 1, put 0 and vice versa

8. Getting the encrypted chromosomes

9. Repeating the 5th, 6th and 7th steps for all chromosomes to get the encrypted image

10. We repeat the process as desired by the user.

B. Algorithm

The GA encryption algorithm are as follows

Algorithm GA_Enc

Input: img: Image;

Output: img: Image;

Begin

For i ← 0 to 100 do

pop ← New Population(img);

Crossover (pop);

Selection (pop);

Mutation (pop);

End for

Return_image(pop);

End.

C. Diagram

The diagram for GA encryption algorithm are as follows:

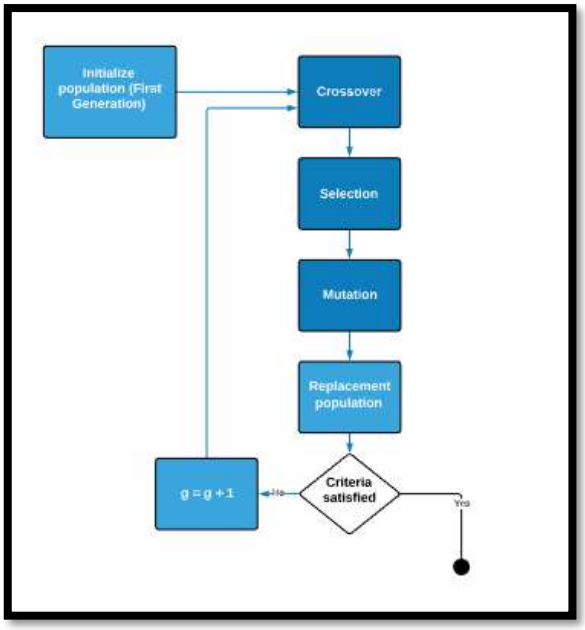


Figure II.9. Diagram of Encryption Algorithm by genetic algorithm.

II-4.2.5 Decryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for GA decryption algorithm

A. Implementation steps

The steps for GA decryption algorithm are as follows:

1. Loading an image.

2. Determining the size of image (height and width).

3. Dividing the image into the chromosomes, but the length of the chromosome is according to the following conditions:

3.1. If the image size is less than (500) dividing the pixels into a set of chromosomes, each block size is (16 pixel).

3.2. If the image size is larger than (500) dividing the pixels into a set of chromosomes, each chromosomes size is (64 pixel)

4. Extraction and storage of RGB components. Chromosome table and consider it as the primary chromosome table.

5. We reverse the mutation process that we explained before.

6. We reverse the crossover process that we explained before.

B. Algorithm

The GA decryption algorithm are as follows:

Algorithm GA_Dec

Input: img: Image;

Output: img: Image;

Begin

For i ← 100 to 0 do

pop ← New Population(img);

Mutation (pop);

Crossover (pop);

Selection (pop);

End for

Return_image(pop);

End.

C. Diagram

The diagram for GA decryption algorithm are as follows:

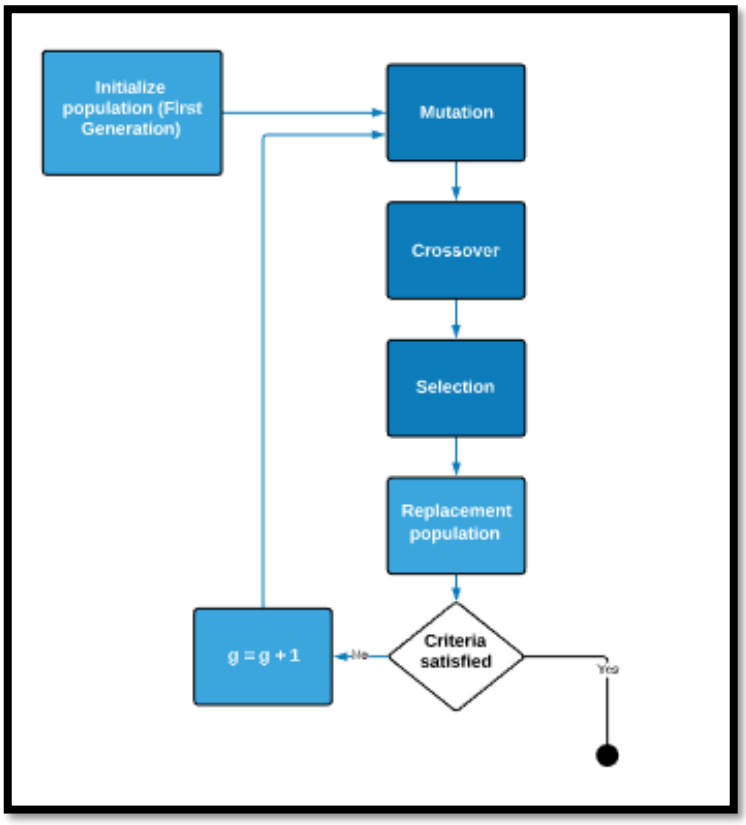


Figure II.10. Diagram of Decryption Algorithm by genetic algorithm.

II-4.2.4. Advantages of GA

There are a number of advantages of Genetic Algorithms. Some of them are as under:

1-The main advantage of the GA lies in its parallelism. Most of the search techniques start from one point and continue until with a single point in each iteration until a final solution is reached.

Therefore a problem of local maxima may exist in them, while the starting solution space in GA is having multiple points in search space and hence the problem of local maxima generally does not exist.

2- The GA is much easier to implement as compared to other techniques as it requires no knowledge or gradient information about the response surface.

3- GA use simple operations, but are able to solve problems which are found to be computationally prohibitive by traditional algorithmic and numerical techniques.

4- Solution time with GA is highly predictable – it is determined by the size of the population, time taken to decode and evaluate a solution and the number of generations of population.

.[63]

II-4.2.5.Limitations of GA based systems

Although there are a number of advantages of GAs, yet there are some limitations as well.

Some of which are described below:

1- One of the biggest problems in implementing is identification of the fitness function.

As the optimal solution heavily depends of the fitness function, therefore it must be determined accurately. There are no standard techniques available to define a fitness function and it is the sole responsibility of the user to define it.

2- Sometimes premature convergence can occur and therefore the diversity in the population is lost, which is one of the major objectives of GA.

3-Another problem is related with the choosing of various parameters like the size of the population, mutation rate, crossover rate, the selection method and its strength.

4-GA themselves are blind to the optimization process, as they only look at the fitness value of each chromosome rather than knowing what the fitness value actually means.[63]

II.4.3. *Third Contribution:* DIP-GA Without Using the Fitness Function

We proposed this algorithm, in order to improve DIP and make it less prone to fracture, we created a hybridization algorithm between DIP and the genetic algorithm.

II.4.3.1. Using Crossover and Mutation Operators of GAs and the projection method

The encryption process was completed after going through two stages, the first is encoding the image using the projection method, then we apply the second encryption process using the genetic algorithm

II-4.3.2 Encryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for DIP-GA encryption algorithm

A. Implementation steps

The steps for DIP-GA encryption algorithm are as follows

1 We are working on image encrypted using the projection method.

2. Determining the size of image (height and width).

3. We divide the pixels you return to us

Matrix (OPT (width, angel, distance), OPT (height, angel, distance)).

to the chromosomes, but the length of the chromosome is according to the following conditions:

3-1 If the image size is less than (500*500) dividing the pixels into a set of chromosomes, each block size is (16 pixel).

3-1 If the image size is larger than (500*500) dividing the pixels into a set of chromosomes, each chromosomes size is (256 pixel)

4. Extraction and storage of RGB components. Chromosome table and consider it as the primary chromosome table.

5. Doing crossover operator is like the first type(Single Point Crossover).

We choose as two specific symbols and Crossover proceeds in there are two possibilities:

5.1. If the session number is a pair : We merge the first half of the first chromosome with the second text of the second chromosome, and the first half of the second chromosome with the second half of the first chromosome.

5.2. If the session number is impair, then we combine the second third of the first chromosome with the other two thirds of the second chromosome. The second third of the second chromosome with the other two thirds of the first chromosome

6. We save new children in a new table..

7. Doing mutation operator

The mutation currency is applied to 3 bit in a chromosome

Bit 1: P=4 and B=12

Bit 2: P=3 and B=6

Bit 2: P=2 and B=18

P:pixel On which we will be booming

B: The bit of the mutation, if it is equal to 1, put 0 and vice versa

8. Getting the encrypted chromosomes

9. Repeating the 5th, 6th and 7t h steps for all chromosomes to get the encrypted image

11. We repeat the process as desired by the user.

B. Algorithm

The DIP-GA encryption algorithm are as follows

Algorithm GA with Projection without fitness_Enc

Input: img: Image,d1: Real,d2: Real,o1: Real,o2: Real;

Output: img: Image;

Begin

Encrypt_projection(img,d1,d2,o1,o2);

For i ← 0 to 100 do

pop ← New Population(img);

Crossover (pop);

Selection (pop);

Mutation (pop);

End for

Return_image(pop);

End.

C. Diagram

The diagram for DIP-GA encryption algorithm are as follows

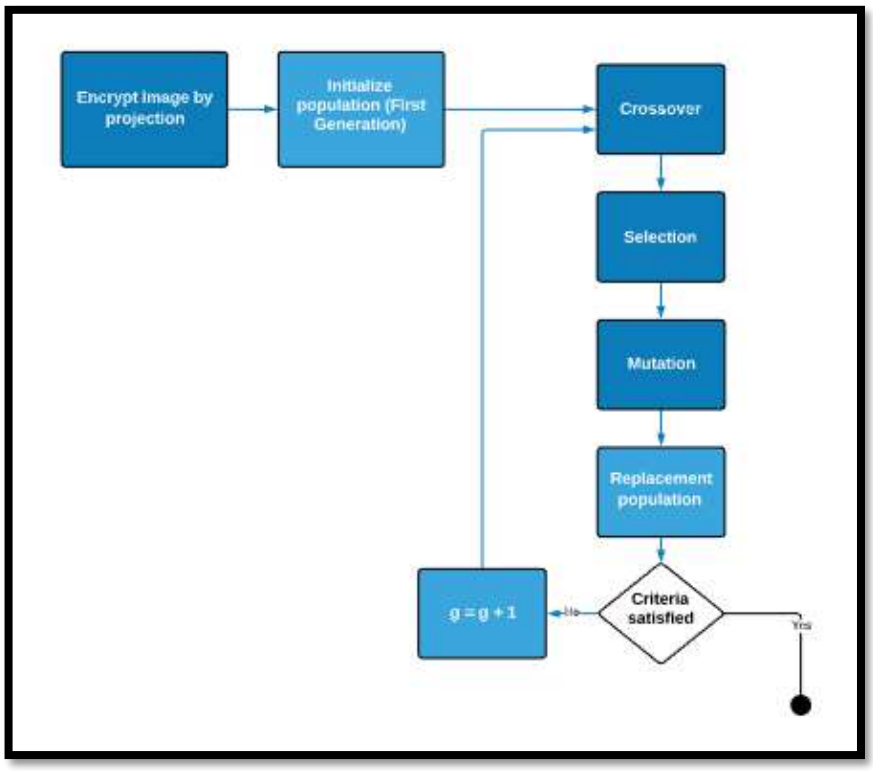


Figure II.11. Diagram of Encryption Algorithm DIP-GA No fitness.

II-4.2.3 Decryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for DIP-GA decryption algorithm

A. Implementation steps

The steps for DIP-GA decryption algorithm are as follows:

1. Loading an image.

2. Determining the size of image (height and width).

3. We divide the pixels you return to us

Matrix (OPT (width ,angel , distance) , OPT (height,angel , distance)).

to the chromosomes, but the length of the chromosome is according to the following conditions:

3-1 If the image size is less than (500*500) dividing the pixels into a set of chromosomes, each block size is (16 pixel).

3-2 If the image size is larger than (500*500) dividing the pixels into a set of chromosomes, each chromosomes size is (64 pixel)

4. Extraction and storage of RGB components. Chromosome table and consider it as the primary chromosome table.

5- We reverse the mutation process that we explained before.

6- We reverse the crossover process that we explained before.

B. Algorithm

The DIP-GA decryption algorithm are as follows

Algorithm GA with Projection without fitness_Dec

Input: img: Image,d1: Real,d2: Real,o1: Real,o2: Real;

Output: img: Image;

Begin

For i ← 100 to 0 do

pop ← New Population(img);

Mutation (pop);

Crossover (pop);

Selection (pop);

End for

Return_image(pop);

Decrypt_projection(img,d1,d2,o1,o2);

End.

C. Diagram

The diagram for DIP-GA encryption algorithm are as follows:

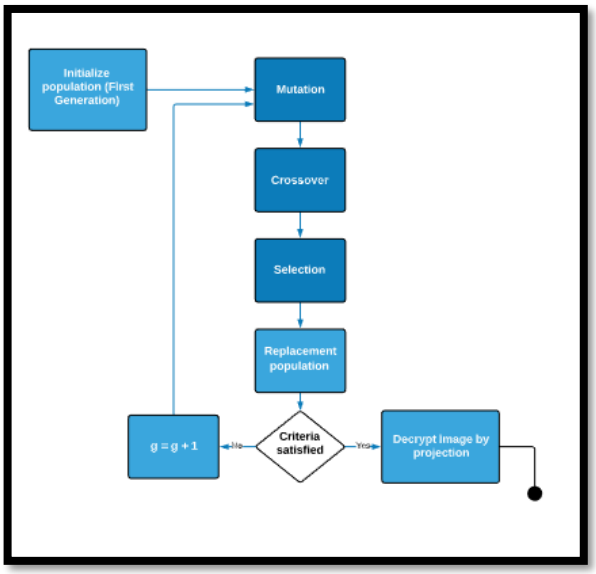


Figure II.12. Diagram of the decryption algorithm DIP-GA No fitness.

II.4.4 *Fourth Contribution:* DIP-GA with The Fitness Function

This algorithm is the same as the others but the generation selection process new determined after fitness calculation

II.4.4.1. Using Crossover and Mutation Operators of Gas and the projection method

The encryption process was completed after going through two stages, the first is encoding the image using the projection method, then we apply the second encryption process using the genetic algorithm

II.4.4.2 Encryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for DIP-GA with The Fitness encryption algorithm

A. Implementation steps

The steps for DIP-GA with The Fitness encryption algorithm are as follows:

1 We are working on image encrypted using the projection method.

2. Determining the size of image (height and width).

3. We divide the pixels you return to us

Matrix (OPT (width, angel, distance), OPT (height, angel, distance)).

to the chromosomes, but the length of the chromosome is according to the following conditions:

3-1 If the image size is less than (500*500) dividing the pixels into a set of chromosomes, each block size is (16 pixel).

3-2 If the image size is larger than (500*500) dividing the pixels into a set of chromosomes, each chromosomes size is (64 pixel)

4. Extraction and storage of RGB components. Chromosome table and consider it as the primary chromosome table.

5. Doing crossover operator is like the first type (Single Point Crossover).

We choose as two specific symbols and Crossover proceeds in there are two possibilities:

5.1. If the session number is a pair: We merge the first half of the first chromosome with the second text of the second chromosome, and the first half of the second chromosome with the second half of the first chromosome.

5.2. If the session number is impair, then we combine the second third of the first chromosome with the other two thirds of the second chromosome. The second third of the second chromosome with the other two thirds of the first chromosome

6. We save new children in a new table.

7 .The test of the new generation goes through two stages.

7.1.We calculate the fitness for both parents and children according to the equation

Fit=α * PSNR(A,B)+ β* MSE(A,B) / Fit.bas and (α+ β=1)

A:is image with parents chromosomes

B: is image with childrens chromosomes

-Represent α and β the importance of the two equations to us, if it were PSNR Most important from MSE, we give to α Value greater than β If not, we do the opposite.

In our case all PSNR and MSE they are equally important so we give them the same value

(α=0.5) + (β =0.5) =1

-Fit.bas is the smallest value of fitness in the previous generation of parents.

7.2. We choose the most suitable chromosomes, either for parents or children, after calculating fitness. The son cannot be tested with the father because if we do this, it causes us to lose information.

If fitness parents < fitness children We choose the parents

Else We choose the children

8. Doing mutation operator

The mutation currency is applied to 3 bits in a chromosome

Bit 1: P=4 and B=12

Bit 2: P=3 and B=6

Bit 2: P=2 and B=18

P: pixel on which we will be booming

B: The bit of the mutation, if it is equal to 1, put 0 and vice versa

9. Getting the encrypted chromosomes

10. Repeating the 5th, 6th, 7th and 8th steps for all chromosomes to get the encrypted image

11. We repeat the process as desired by the user.

B. Algorithm

The DIP-GA with The Fitness encryption algorithm are as follows:

Algorithm GA with Projection_Enc

Input: img: Image,d1: Real,d2: Real,o1: Real,o2: Real;

Output: img: Image;

Begin

Encrypt_projection(img,d1,d2,o1,o2);

For i ← 0 to 100 do

pop ← New Population(img);

Crossover (pop);

For i= 0 to pop_lenght/2 do

fitness = Fitness(img.chromosome1,img.chromosome2);

If (fitness > fitness_best) then Select two sons chromosomes and add them to an img;

Else Select two Parents chromosomes and add them to an img;

End if

End for

Selection (pop);

Mutation (pop);

End for

Return_image(pop);

End.

C. Diagram

The diagram for DIP-GA with The Fitness encryption algorithm are as follows:

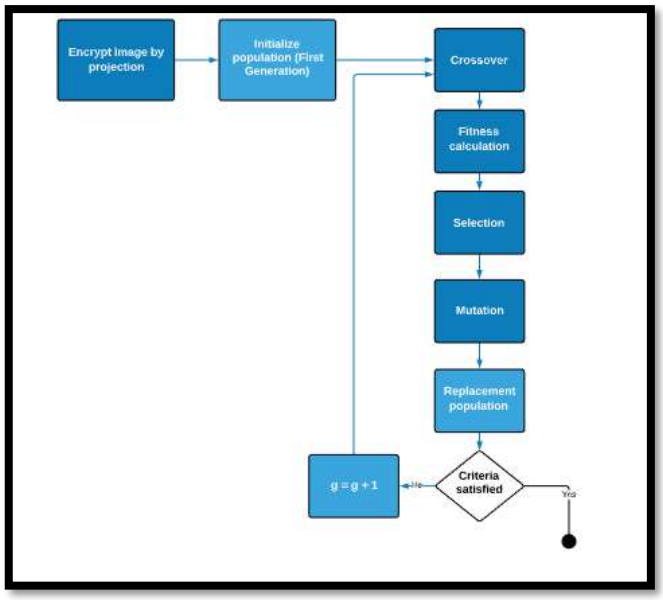


Figure II.13. diagram of the encryption algorithm by hybrid projection method and GA with computation of fitness.

II.4.4.2 Decryption algorithm

We will mention the implementation stages, then show you the execution algorithm In the last We will show The diagram for DIP-GA with The Fitness decryption algorithm

II.4.2.1 Implementation steps

The steps for DIP-GA with The Fitness decryption algorithm are as follows

1. Loading an image.

2. Determining the size of image (height and width).

3. We divide the pixels

Matrix (OPT (width, angel, distance), OPT (height, angel, distance)).

to the chromosomes, but the length of the chromosome is according to the following conditions:

3-1 If the image size is less than (500*500) dividing the pixels into a set of chromosomes, each block size is (16 pixel).

3-2 If the image size is larger than (500*500) dividing the pixels into a set of chromosomes, each chromosomes size is (64 pixel)

4. Extraction and storage of RGB components. Chromosome table and consider it as the primary chromosome table.

5. We reverse the mutation process that we explained before.

6. We reverse the crossover process that we explained before.

7. We reverse the Selection process that we explained before.

10. Repeating the 5th, 6th , 7th and 8th steps for all chromosomes to get the de

crypted image

11. We repeat the process as desired by the user.

II.4.4.2.2 Algorithm

The DIP-GA with The Fitness decryption algorithm are as follows

Algorithm GA with Projection_Dec

Input: img: Image,d1: Real,d2: Real,o1: Real,o2: Real;

Output: img: Image;

Begin

For i ← 100 to 0 do

pop ← New Population(img);

Mutation (pop);

Crossover (pop);

Selection (pop);

End for

Return_image(pop);

Decrypt_projection(img,d1,d2,o1,o2);

End.

II.4.4.2.3 diagram

The diagram for DIP-GA with The Fitness decryption algorithm are as follows

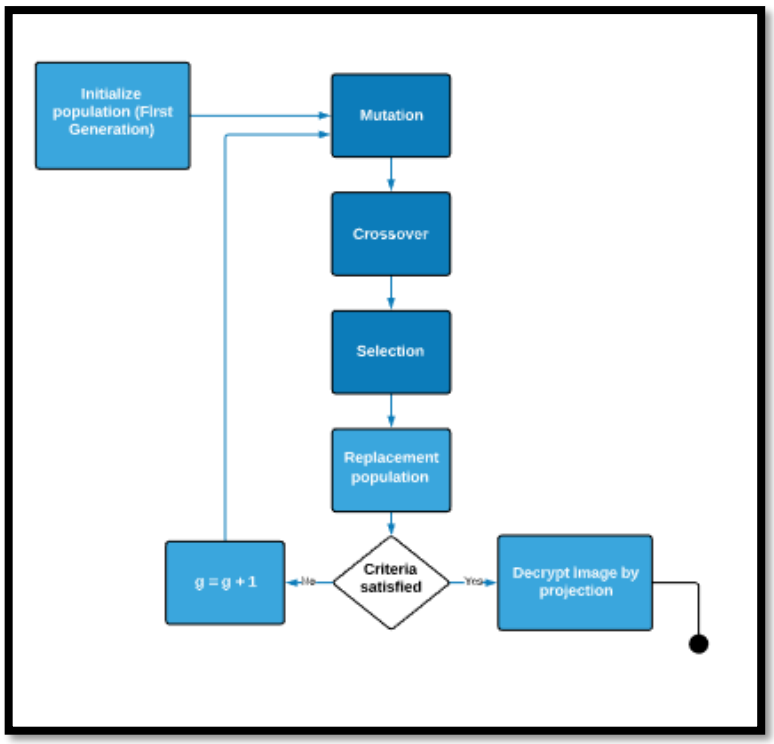


Figure II.14. Diagram of the encryption algorithm by hybrid projection method and GA with computation of fitness.

II.5 Conclusion

In this chapter, we presented several algorithms for encoding images, some of which are previously in place, the most important of which are the ones that we have proposed as new and effective algorithms in the encryption process, in addition to the stages of implementation of these algorithms and their strengths and weaknesses.

We are now ready to move to the next step, which is to achieve practical work By showing the results of encryption and subjecting them to a number of tests to demonstrate its effectiveness

CHAPTER III: THE RESULTS AND ANALYSIS DIRECT IMAGE PROJECTION USING GENETIC ALGORITHM (DIP-GA)

III-1.introduction

This chapter is devoted to the practical implementation of the proposed algorithms that we previously programmed. We will first see the results of implementing the algorithms that we discussed in the second chapter, and we will also extract the graph of all the encoded images with those algorithms and compare it with the graph of the original images, and then analyze (the encoded files)) With it and we subject it to many image quality tests, and in the end we compare the results of the tests for each algorithm and conclude which is the best in all security aspects.

III-2 Definition of image quality

Historically, image quality is described in terms of the visibility of the distortions in an image, such as colour shifts, blurriness, Gaussian noise and blockiness. The most common way of modeling an image quality metric is therefore by quantification of the visibility of these distortions. For example, Just Noticeable Difference (JND) model by Sarnoff predicts subjective rating of an image by examining the visibility of distortions. In , Janssen has proposed a new philosophy for image quality. He regards image as carriers of visual information instead of two-dimensional signals, regards visual-cognitive processing as information processing rather than signal processing and regards image quality as image adequacy in the visual interaction process instead of visibility of distortions. Though this concept is interesting and seem applicable, most of the works are still reported based on the former concept, due to its simplicity and good performance .[34]

III-2.1The investigated metrics are as follow:

Attention: In all of our phrases denote the following symbols

M: height the image.

N: width the image.

a_{ij} : Pixel, which is located in the i j site for the first image.

b_{ij} : Pixel, which is located in the i j site for the second image.

III-2.1.1 Mean square error (MSE)

Mean Squared Error (MSE): One obvious way of measuring this similarity is to compute an error signal by subtracting the test signal from the reference, and then computing the average

energy of the error signal. The mean-squared-error (MSE) is the simplest, and the most widely used, full-reference image quality[14]

measurement.MSE stands for the mean squared difference between the original image and the projected image. It is calculated by formula 1. a_{ij} is the value of the pixel at the coordinates (i, j) in the original image. b_{ij} is the pixel's value at the same coordinates in the corresponding generated image [22].

$$MSE = \left(\frac{1}{M \times N}\right) \sum_{i=1}^{M} \sum_{j=1}^{N} \left(a_{ij} - b_{ij}\right)^{2} \quad (1)$$

III-2.1.2.Peak Square Noise Ratio (PSNR)

The PSNR is inversely proportional to the Mean Squared Error. When comparing the two images, PSNR is calculated by taking the Mean Squared Error (MSE) between the pixel intensities and taking the ratio of the maximum possible intensity to the result of the calculation.[22]

PSNR, given by Eq. 2, is a classical quality index defined as the ratio between the maximum possible pixel value (e.g., equals to 255 in case of RGB color images) and the mean square error. It is given by:

$$\boldsymbol{PNSR = \frac{10log_{10}255^{2}}{MSE}} \qquad (\mathbf{2})$$

PSNR of RGB color images can be calculated by evaluating then summing up MSEs of all channels. That's to say, the peak value $\frac{255^2}{MSE}$ is replaced with $\frac{\mathbf{255^2 \times 3}}{\sum_{i \in \{R,G,B\}} \boldsymbol{MSE_i}}$**.** PSNR is more consistent with the presence of error compared to the SNR.

$$\boldsymbol{PNSR = \frac{10log_{10}\, 255^{2} \times 3}{\sum_{i \in \{R,G,B\}} MSE_i}} \qquad (\mathbf{3})$$

III-2.1.3 Average Difference (AD)

The average difference refers to the pixel difference between the original image and its corresponding degraded image. This measure is applicable to any image processing applications

where we find the average difference between two images. Larger value of the AD, specifies the poor quality of the image .[22]

AD is the difference average between the original and the cover image. AD is given by the Eq. 4 :

$$AD = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (a_{ij} - b_{ij})}{MN} \qquad (4)$$

III-2.1.4 Maximum Difference (MD)

It is the absolute difference between original and Degraded image. Higher the value of Maximum Difference indicates that the image is poor quality [22].

MD is given by Eq. 5, is the maximum difference among pixels of the original image and their corresponding ones in the cover image .

$$MD = MAX|a_{ij} - b_{ij}| \qquad (5)$$

III-2.1.5 Peak Mean Square Error (PMSE)

It stands for the mean square error (MSE) based on the square of the maximum value among original image pixels:

$$PMSE = \frac{1}{MN} \times \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (a_{ij} - b_{ij})^2}{[MAX(a_{ij})]^2} \qquad (6)$$

III-2.1.6 Normalized Cross-Correlation (NCC)

Normalized Cross-Correlation (NCC): The closeness between two digital images can also be quantified in terms of correlation function. Normalized Cross-Correlation (NCC) measures the similarity between two images and is given by the equation.[22]

NCC is one of the methods used for template matching. The process used for finding incidences of cover images and original images.

$$NCC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(a_{ij}\times b_{ij})}{\sum_{i=1}^{M}\sum_{j=1}^{N}(a_{ij}{}^{2})} \qquad (7)$$

III-2.1.7 Structural Content (SC)

Structural Content (SC): SC is also correlation based measure and measures the similarity between two images. [22]

SC can be used to determine the nearest of the original image from the cover image. SC is calculated as follows:

$$SC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(a_{ij}{}^{2})}{\sum_{i=1}^{M}\sum_{j=1}^{N}(b_{ij}{}^{2})} \qquad (8)$$

III-2.1.8 Laplacian Mean Square Error(LMSE)

(LMSE). The experimental result shows that laplacian error map localizes the error in a better way compared to structural similarity index (SSIM) map.[22]

LMSE quantifies the quality of image reconstruction. It is calculated as follows:

$$LMSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(O(a_{ij}) - O(b_{ij})\right)}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(O(a_{ij})\right)^{2}} \qquad (9)$$

.where $O(a_{ij}) = a_{i+1j} + a_{i-1j} + a_{ij+1} + a_{ij-1} - 4a_{ij}$.

III-2.1.9 Normalized Absolute Error (NAE)

The normalized absolute error is a measure of how far is the reconstructed image from theo-riginal image, with the value of zero being the perfect fit. A large value of Normalised abso-lute error indicates a poor quality image and a small value gives a good quality image [46].

NAE is a measure of how far is the stego image from the original cover image with the value of zero being the perfect fit. Big value of NAE indicates poor quality of the resulting image after embedding.

$$NAE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij} - b_{ij}|}{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|} \qquad (10)$$

III-2.1.10 the (NPCR) and (UACI)

The NPCR and UACI are two most significant quantities that quantify the strength ofencryption algorithms. NPCR is the measure of absolute number of pixels change rate and UACIcomputes average difference of color intensities between two images when the change in one image is subtle. The NPCR and UACI values can be evaluated by Eqns. (11) and (12), where *T* denotes the largest supported gray-value compatible with image format, |· | denotes the absolute value function.

$$\mathrm{NPCR} = \mathrm{N}\left(a_{ij}, b_{ij}\right) = \sum_{\mathrm{ij}} \frac{\mathrm{D(i,j)}}{\mathrm{N}\times\mathrm{M}} \times 100\% \qquad (11)$$

$$\mathrm{NPCR} = \mathrm{U}\left(a_{ij}, b_{ij}\right) = \frac{1}{\mathrm{N}\times\mathrm{M}} \sum_{\mathrm{ij}} \frac{|a_{ij} - b_{ij}|}{\mathrm{T}} \times 100\% \qquad (12)$$

$$\mathrm{D(i,j)} = \begin{cases} 0 & \text{if } a_{ij} = b_{ij} \\ 1 & \text{if } a_{ij} \neq b_{ij} \end{cases}$$

III-3 The results and Analysis

This section presents several experiments to test the effect of the proposed algorithms on image coding. The proposed methods are implemented and applied in four different types and sizes to check their adaptability, quality, safety and speed. Results are presented in this section

to confirm the characteristics of the proposed methods based on genetic algorithms and to compare them with the image encrypted quality.

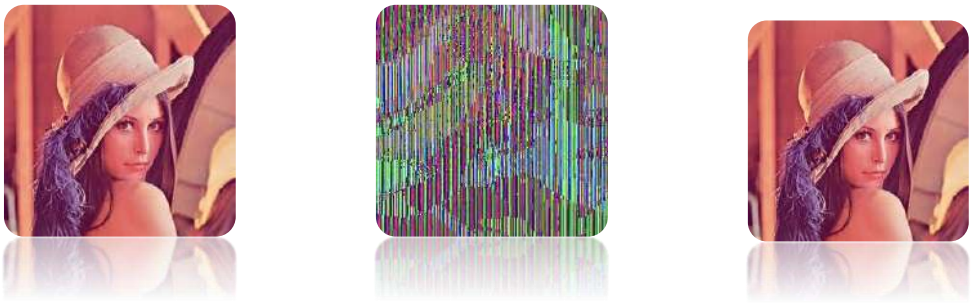
The security analysis performed in these methods is visual analysis, graph analysis, correlation coefficient. All these experiments demonstrate that the methods proposed in this chapter achieve all results completely and correctly. In this subsection, discussions and analyzes of cryptographic algorithms based on genetic algorithm and others are presented. The following discussions and analyzes are conducted on the original, encrypted and decrypted images.

III-3.1. "Rivest Cipher 4" or "Ron'S Code 4" algorithm (RC4)

We will first experiment with encryption and decryption using several image scales, then we will perform a visual analysis of the encrypting process, and at the end we will display the Histogram results and conduct an analysis.

III-3.1.1 Implement the encryption process

The proposed algorithm are implemented and applied to four different images in types and sizes to verify their adaptability, quality, security and speed.



(a) original image (b) Encrypted image (c) decrypted image

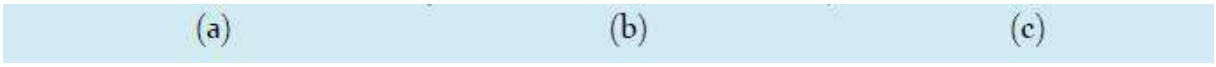


Figure III.1: Encryption and decryption of Image 'Lena' by RC4 algorithm

(a) original image (b) Encrypted image (c) decrypted image

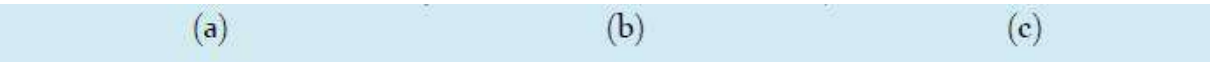
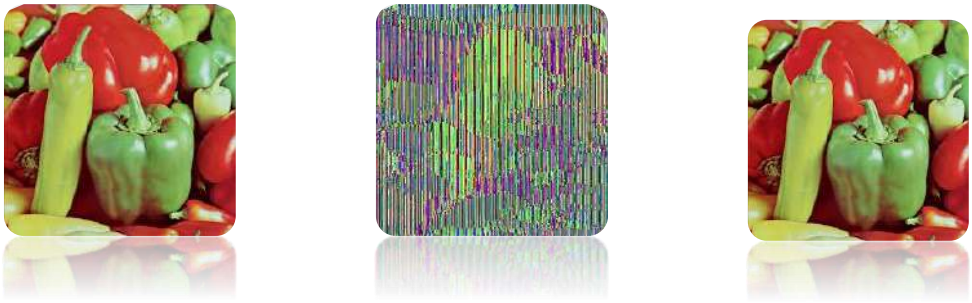


Figure III.2:Encryption and decryption of Image 'baboon' by RC4 algorithm



(a) original image (b) Encrypted image (c) decrypted image

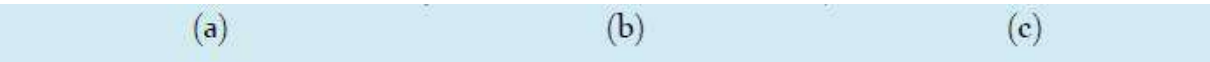
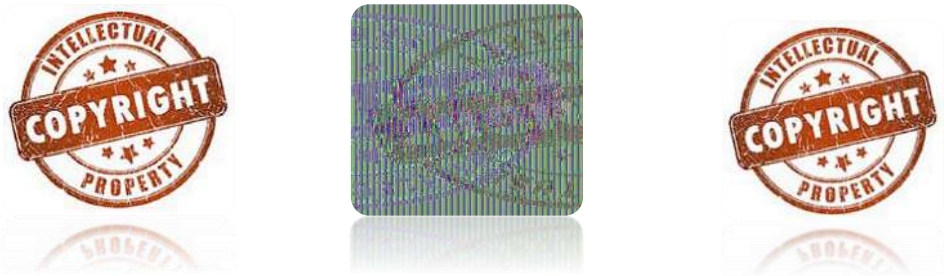


Figure III.3:Encryption and decryption of Image 'pepper' by RC4 algorithm



(a) original image (b) Encrypted image (c) decrypted image

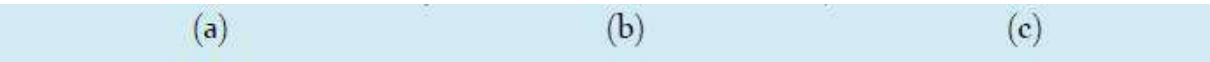


Figure III.4: Encryption and decryption of Image 'copyright' by RC4 algorithm

III-3.1.2 Visual Analysis

The purpose of the visual test is to highlight the similarities between a regular image and its symbols. Figures III.1- III.2 - III.3 - III.4 show that the encrypted image is not completely clear, but it does not display the original image in all encrypted images. The visual test was

performed on different images, which differed in sizes and shapes, and showed the same results.

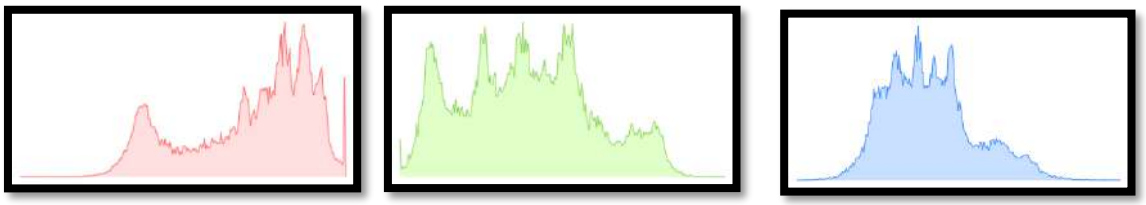
III-3.1.3.Statistical Analysis

Statistical analysis has been performed on the proposed image encryption approaches, demonstrating its superior confusion and diffusion properties, which strongly resist statistical attacks. This is shown by the test on the histograms of the enciphered images, and on the correlations of the adjacent pixels in the ciphered image.

III-3.1.4 Histogram Analysis.

Histogram analysis gives the idea of statistical analysis attackers. Statistical analysis has been performed on the proposed approach, demonstrating its superior confusion and diffusion properties, which strongly resist statistical attacks. The proposed approaches give the original image and cipher image histogram [2-3]. Figures III.5 – III.6 show Histogram of the regular "Lena" image and the encoded image using RC4 algorithm While comparing the two, the histogram of the encrypted image is uniformly distributed and is relatively different from the histograms of the original images. The histogram of the fully encrypted image has increased until it climaxes and is lowered, as for the original image it is irregular

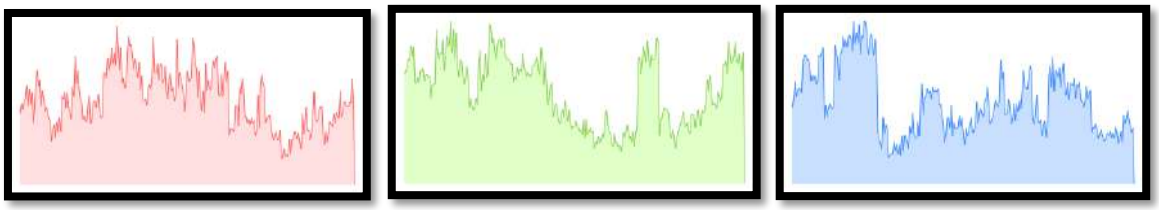
(a) original image



b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.5:Original Image of Len and Its RGB Histogram

Encrypted image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.6:Encrypted Image by RC4 algorithm of lena and Its RGB Histogram

III-3.2 LSB algorithm(Least Significant Bit)

We will first experiment with encryption and decryption using several image scales, then we will perform a visual analysis of the encrypting process, and at the end we will display the Histogram results and conduct an analysis

III-3.2.1 Implement the encryption process

The proposed algorithm are implemented and applied to four different images in types and sizes to verify their adaptability, quality, security and speed.

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.7:Encryption and of Image 'Lena' by LSB algorithm

(a) original image (b) Encrypted image (c) decrypted image

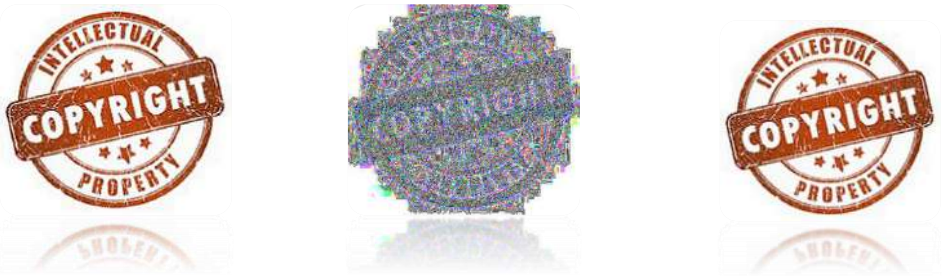
(a) (b) (c)

Figure III.8:Encryption and decryption of Image 'baboon' by LSB algorithm

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.9: Encryption and decryption of Image 'pepper' by LSB algorithm



(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.10:Encryption and decryption of Image 'copyright' by LSB algorithm

III-3.2.2 Visual Analysis.

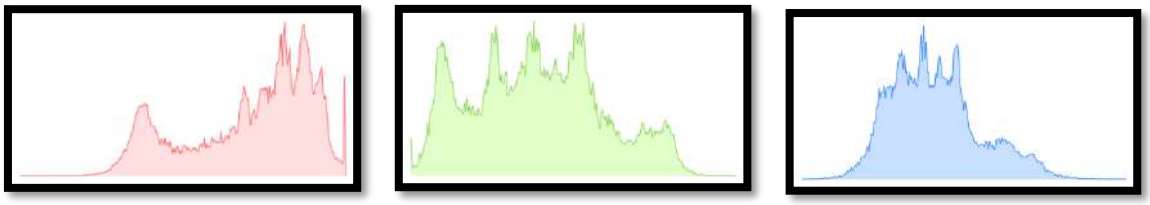
The purpose of the visual test is to highlight the similarities between a regular image and its symbols. Figures III.7 – III.8 - III.9 show that the encoded images do not have any features of normal images. A visual test was performed on different images, which differ in sizes and shapes, and showed no perceptual similarity for most tests, but for the Figures III.10 logo image, he is not completely confused and there is a perceptual similarity between the original image and the encrypted image.

III-3.2.3 Histogram Analysis

The histogram analysis is used to illustrate the confusion and diffusion properties in the encrypted image for testing purposes Histogram of the regular "Lena" image and the encoded image using the LSB algorithm are shown in figures

III.11- III.12 While comparing the two, the histogram of the encrypted image is fairly uniform and is significantly different from that of the original image. The encrypted images transmitted are not affected by any attacker.

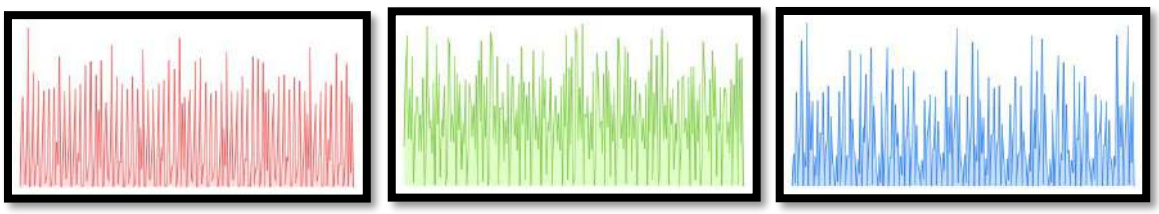
(a)original image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.11:Original Image of Lena and Its RGB Histogram

(a)Encrypted image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.12: Encrypted Image by LSB algorithm of Lena and Its RGB Histogra

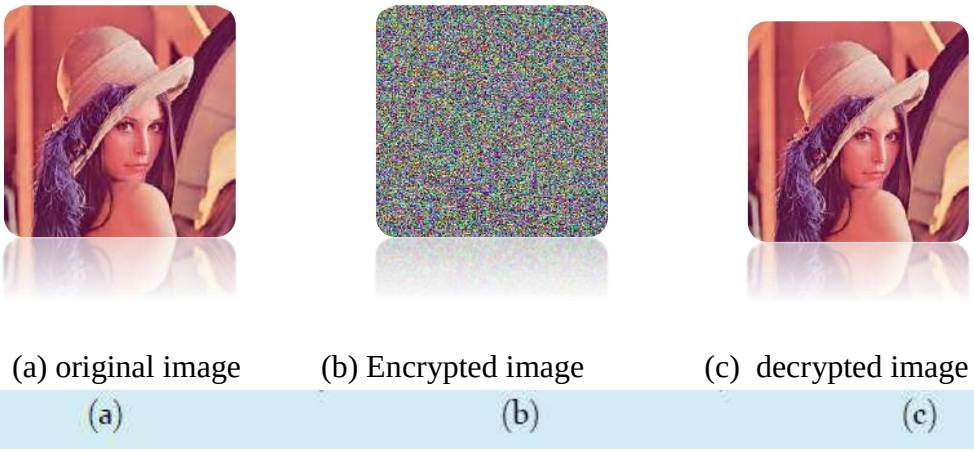
III-3.3 First contribution: Direct Image Projection (DIP)

We will first experiment with encryption and decryption using sdifferent images in types and sizes , then we will perform a visual analysis of the encrypting process, and at the end we will display the Histogram results and conduct an analysis

III-3.3.1 Implement the encryption process

The proposed algorithm are implemented and applied to four different images in types and sizes to verify their adaptability, quality, security and speed.

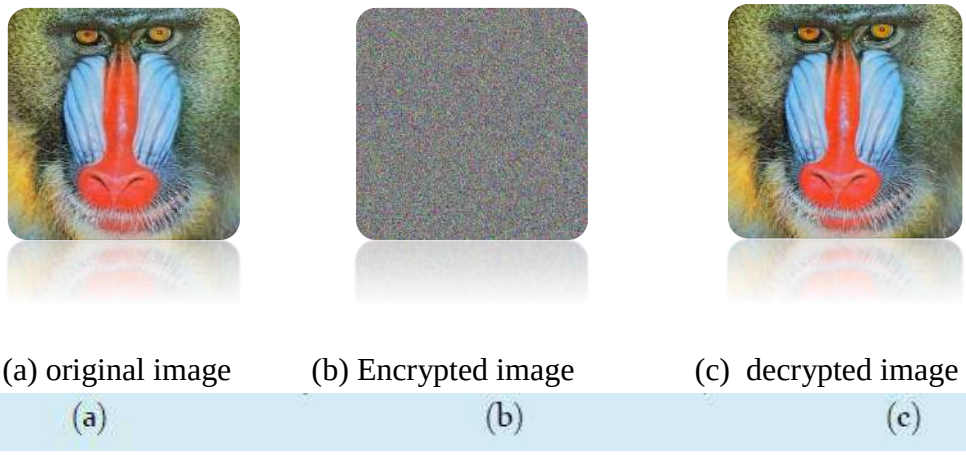
The distance and angle have respectively been set to 1.5 and 0.52 rad for both vertical and horizontal projection.



(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.13:Encryption and decryption of Image 'Lena' by DIP



(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.14: Encryption and decryption of Image 'baboon' by DIP

(a) original image (b) Encrypted image (c) image decrypted

(a) (b) (c)

Figure III.15: Encryption and decryption of Image 'pepper' by DIP

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.16: Encryption and decryption of Image 'copyright by DIP

III-3.3.2 Visual Analysis.

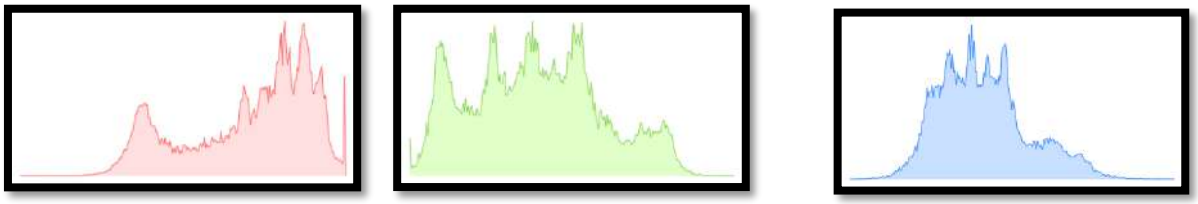
The purpose of visual testing is to highlight the presence of the similarities between plain image and its cipher. Figures III.13- III.14- III.15- III.16 shows that the encrypted images do not contain any features of the plain images. After the visual testing had been performed on some images, which have different sizes and formats, it showcased that there is no perceptual similarity.

III-3.3.3 Histogram Analysis

The histogram analysis is used to illustrate the confusion and diffusion properties in the encrypted image for testing purposes. The histogram of the plain image 'Lena' and that of the encrypted image by DIP method are shown in figures III.17– III.18 While comparing the two, The histogram of the encrypted image is uniformly distributed and significantly different from the respective histograms of the original images. The cipher image histogram is hori-

zontal, so if any statistical attackers attack, they will not be able to break the proposed security..

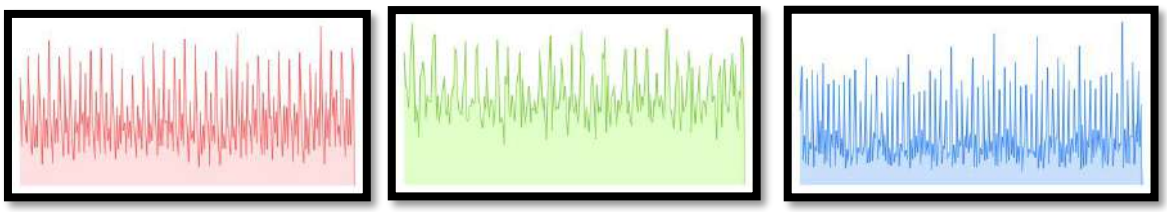
(a)original image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.17: Origina Image of Lena and Its RGB Histogram

(a) Encrypted image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.18: Encrypted Image by DIP of Lena and Its RGB Histogram

III-3.3.4 Direct Image Projection Analysis:

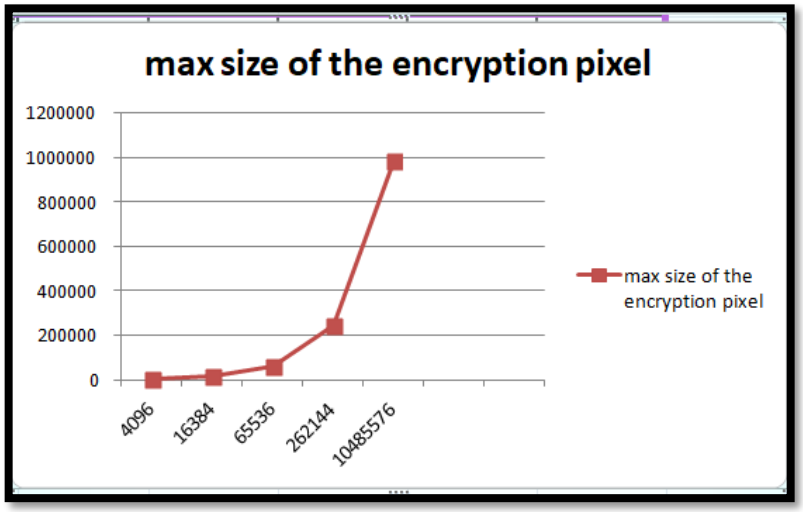
you will make experimentation and a discuss of :

A . The relationship of the encryption pixel ratio to the size of the image

The relationship between the size of the image and the corresponding max size of the encryption pixel The distance and angle have respectively been set to 2.0 and 30o= 0.16 rad for both vertical and horizontal projection.

Image size	**64x64**	**128x128**	**256x256**	**512x512**	**1024x1024**
max size of theencryption pixel	60x60	122x122	248x248	496x496	994x994
Available proportion	87.9%	90.8%	93.8%	93.8%	94.2%

Table III.1:the change the the encryption pixel values to changing the size of the image



FigureIII.19: the change the the encryption pixel values to changing the size of the image

The FigureIII.19 represents change the the encryption pixel values to changing the size of the image. Search The distance and angle were set respectively 2.0 and 30 degrees = 0.16 rad for

both the vertical and horizontal projection. Where we note that the relationship between the image size and the size of the the encryption pixel in general is a direct relationship Search as the image size increases, the size of the encryption pixel increases and vice versa.

B. The relationship of the encryption pixel ratio to the Angle

The affect of choosing the rotation angle on the hiding available proportion. The image size and distance are 250x250 and 1.5 respectively in all cases.

Angle (rad)	**0**	**0.26**	**0.52**	**0.78**	**1.04**	**1.30**	**1.57**
max size of the encryption pixel	250x250	236x236	232x232	224x224	178x178	99x99	1x1
Available proportion	100%	89.11%	86.12%	80.28%	50.69%	16%	0.0144%

Table III.2 the change of the encryption pixel values in relation terms of change in angle values

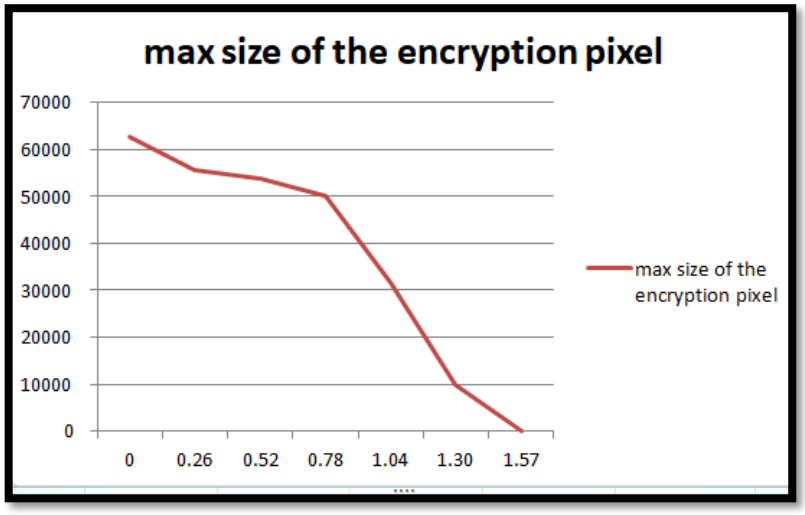


Figure III.20: Curve the change of the encryption pixel values in relation terms of change in angle values

The Figure III.20 represents the encryption pixel values in relation terms of change in angle values of image size and distance of 250 × 250 and 1.5, respectively in all cases. Where we

notice that the relationship between the size of the encryption pixel and the value of the angle in general is an inverse relationship, look for the lower the angle value, the larger the size of the encryption pixel and vice versa

C. The relationship of the encryption pixel ratio to the distance

The affect of choosing the distance on the hiding available proportion. The image size and angle are 250x250 and 0.5 respectively in all cases.

Distance	**0.4**	**0.8**	**1.2**	**1.6**	**2.0**	**2.4**	**2.8**
max size of the encryption pixel	166 x166	206 x206	226 x226	236 x236	240 x240	240x240	238x238
Available proportion	44.1%	67.9%	81.7%	89.11%	92.16%	92.16%	90.63%

Table III.3:the change the encryption pixel values in relation terms of to changing in distance values

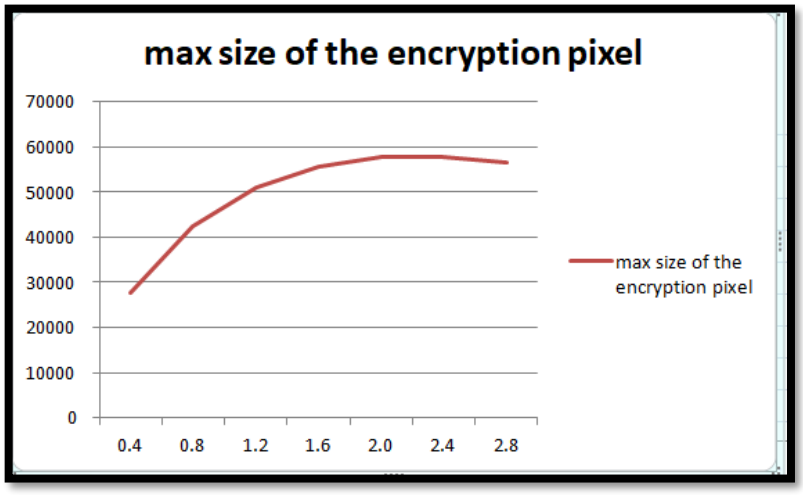


Figure III.21:Curve of change of coding pixel values with respect to change in distance values

The Figure III.21 represents change the encryption pixel values in relation terms of to changing in distance values. Search Image size and angle are set to 20 and 0.16 respectively. Where we note that the relationship between the size of the image and the size of the encryption pixel in general is a positive relationship. The larger the distance, the larger the encryption pixel size and vice versa

III-3.4 Second Contribution: Genetic Algorithm (GA)

We will first experiment with encryption and decryption using sdifferent images in types and sizes , then we will perform a visual analysis of the encrypting process, and at the end we will display the Histogram results and conduct an analysis

III-3.4.1 Implement the encryption process

The proposed algorithm are implemented and applied to four different images in types and sizes to verify their adaptability, quality, security and speed. In all of the tests, the genetic algorithm was repeated 1,000 times

(a) original image (b) Encrypted image (c) decrypted image

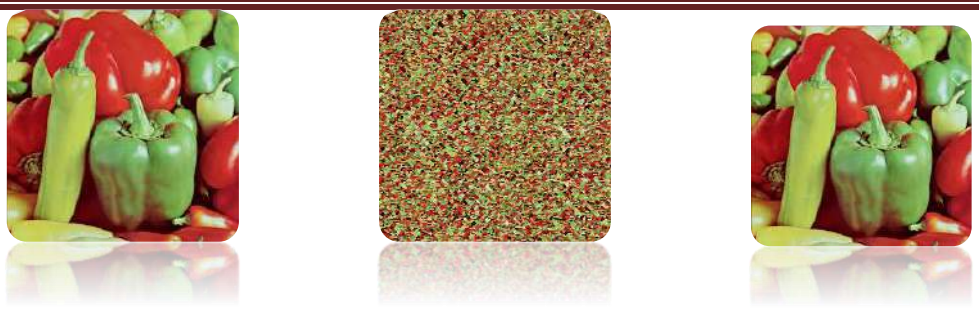
(a) (b) (c)

Figure III.22 :Encryption and decryption of Image 'Lena' by genetic algorithm

(a) original image (b) Encrypted image (c) decrypted imag

(a) (b) (c)

Figure III.23:Encryption and decryption of Image 'baboon' by genetic algorithm



(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.24:Encryption and decryption of Image 'pepper' by genetic algorithm

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.25:Encryption and decryption of Image 'copyright' by genetic algorithm

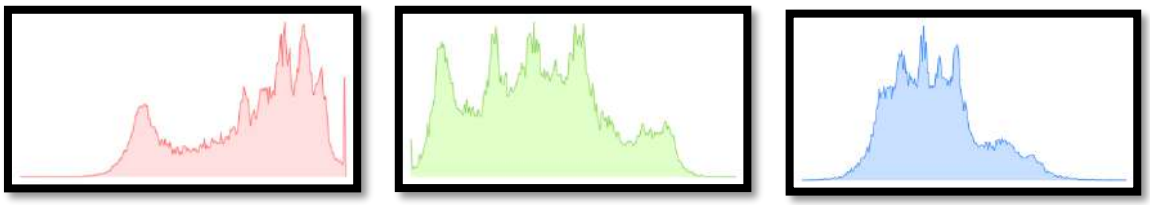
III-3.4.2 Visual Analysis

The purpose of visual testing is to highlight the presence of the similarities between plain image and its cipher. Figures III..22- III..23- III.24 - III.25 shows that the encrypted images do not contain any features of the plain images. After the visual testing had been performed on some images, which have different sizes and formats, it showcased that there is no perceptuasimilarity.

III-3.4.3 Histogram Analysis

Histogram analysis is used to demonstrate the characteristics of the disturbance and propagation in the coded image for test purposes. The histogram of the normal "Lena" image and the encrypted image by GA image are shown in Figures III.26 to III.27 When comparing the two, the histogram of the encoded image for the red color is completely different from the graph of the original image. As for the blue and green colors, the difference was not clear.

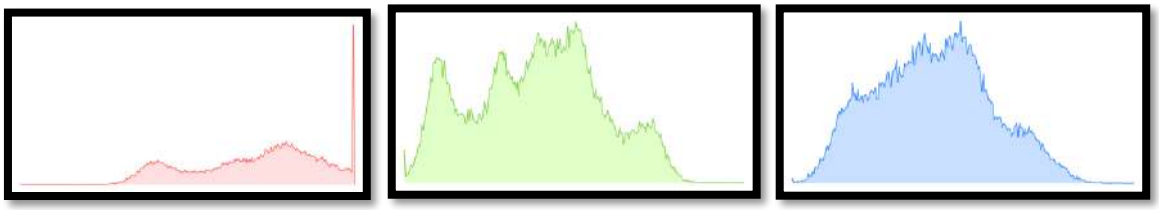
(a) original image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.26: Original Image of Lena and Its RGB Histogram

(a) Encrypted image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

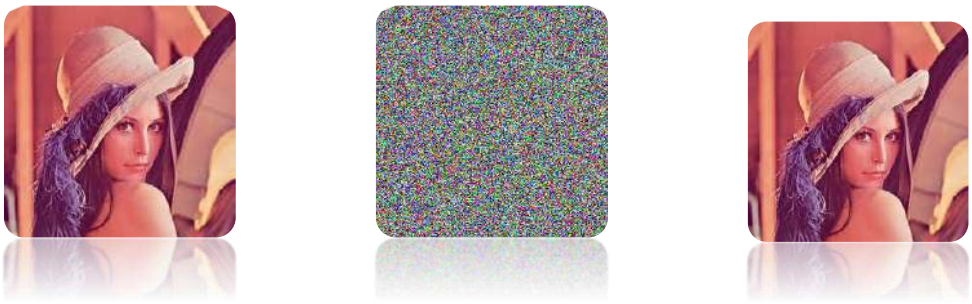
Figure III.27: Encrypted Image by genetic algorithm of Lena and Its RGB Histogram

III-3.5 *Third Contribution:* DIP-GA Without Using the Fitness Function.

We will first experiment with encryption and decryption using sdifferent images in types and sizes , then we will perform a visual analysis of the encrypting process, and at the end we will display the Histogram results and conduct an analysis

III-3.5.1 Implement the encryption process

The distance and angle have respectively been set to 2.0 and 30o= 0.16 rad for both vertical and horizontal projection. And The genetic algorithm was repeated 50 times.



(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.28: Encryption and decryption of Image 'Lena' by DIP-GA Without Using the Fitness Function

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.29:Encryption and decryption of Image 'baboon' Lena' by DIP-GA Without Using the Fitness Function

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.23: Encryption and decryption of Image 'pepper' Lena' by DIP-GA Without Using the Fitness Function

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.31: Encryption and decryption of Image 'copyright' by DIP-GA Without Using the Fitness Function

III-3.5.2 Visual Analysis

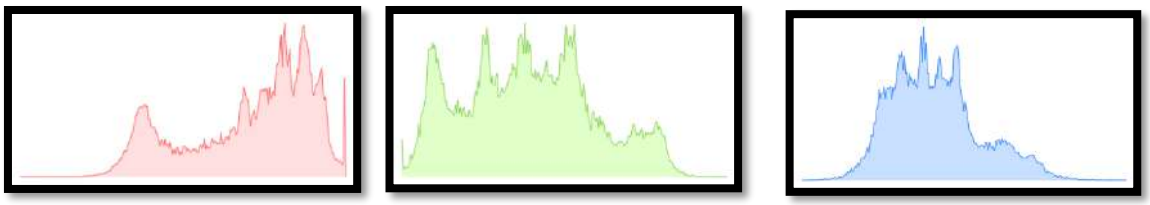
The purpose of visual testing is to highlight the presence of the similarities between plain image and its cipher. Figures III.28- III.29- III.30- III.31 shows that the encrypted images do not contain any features of the plain images. After the visual testing had been performed on some images, which have different sizes and formats, it showcased that there is no perceptual similarity.

III-3.5.3 Histogram Analysis

The histogram analysis is used to illustrate the confusion and diffusion properties in the encrypted image for testing purposes. The histogram of the plain image 'Lena' and that of the encrypted image by DIP-GA Without Using the Fitness Function are shown in figures III.32 - III.33 Encryption and Decryption Image

. While comparing the two, the histogram of the encrypted image is fairly uniform and is significantly different from that of the original image. The encrypted images transmitted are not affected by any attacker.

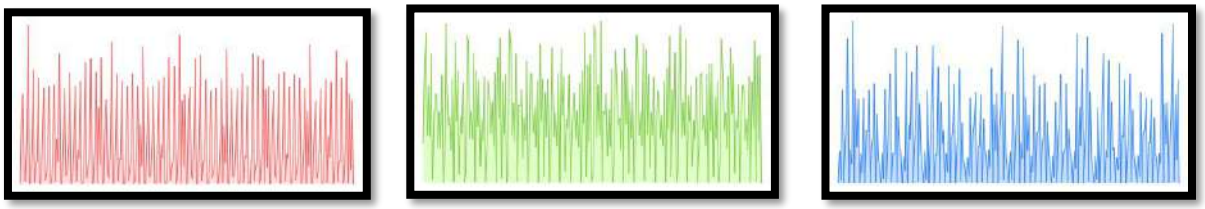
(b) original image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.32: Original Image of Lena and Its RGB Histogram

(b) Encrypted image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.33: Encrypted Image by DIP-GA Without Using the Fitness Function of Lena and Its RGB Histogram

III-3.6 Fourth Contribution: DIP-GA with The Fitness Function.

We will first experiment with encryption and decryption using sdifferent images in types and sizes , then we will perform a visual analysis of the encrypting process, and at the end we will display the Histogram results and conduct an analysis

III -3.6.1 Implement the encryption process

The distance and angle have respectively been set to 2.0 and 30o= 0.16 rad for both vertical and horizontal projection. And The genetic algorithm was repeated 50 times.

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.31: Encryption and decryption of Image 'Lena' by DIP-GA with The Fitness Function.

(a) original image (b) Encrypted image (c) decrypted image

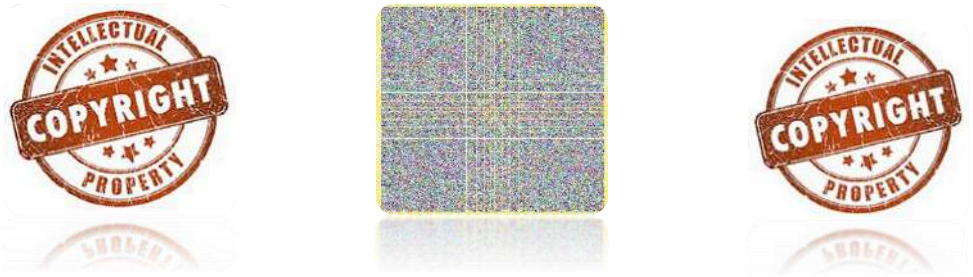
(a) (b) (c)

Figure III.32: Encryption and decryption of Image 'baboon' Lena' by DIP-GA with The Fitness Function..

(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.33: Encryption and decryption of Image 'pepper' Lena' by DIP-GA with The Fitness Function.



(a) original image (b) Encrypted image (c) decrypted image

(a) (b) (c)

Figure III.34: Encryption and decryption of Image 'copyright' by DIP-GA with The Fitness Function.

III-3.6.2 Visual Analysis

The purpose of visual testing is to highlight the presence of the similarities between plain image and its cipher. Figures **III.31- III.32 - III.33- III.34** shows that the encrypted images do not contain any features of the plain images. After the visual testing had been performed on some images, which have different sizes and formats, it showcased that there is no perceptual similarity.

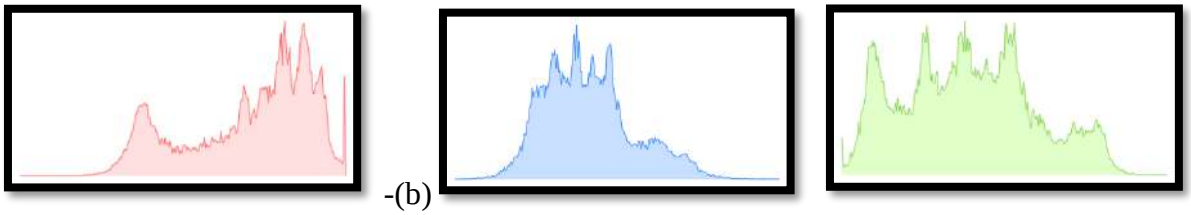
III-3.6.3 Histogram Analysis

The histogram analysis is used to illustrate the confusion and diffusion properties in the encrypted image for testing purposes. The histogram of the plain image 'Lena' and that of the encrypted image by DIP-GA with The Fitness Function. are shown in figures III.35 - III.36 Encryption and Decryption Image ,While comparing the two, the histogram of the encrypted

image is fairly uniform and is significantly different from that of the original Image.The encrypted images transmitted are not affected by any attacker.

.

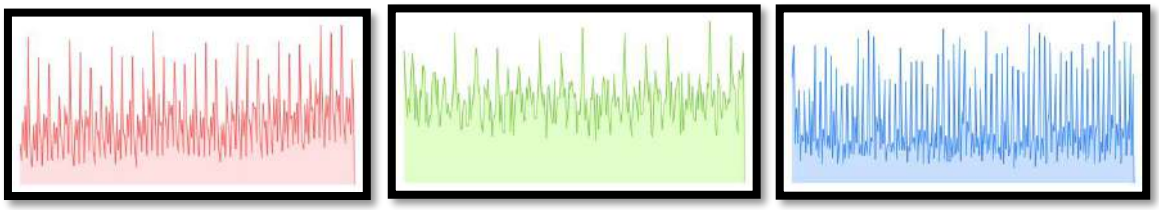
(a) original image



-(b) Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.35:Original Image of Lena and Its RGB Histogram

(a) Encrypted image



(b)- Histogram of red Chanel (c)- Histogram of green Chanel (d)- Histogram of blue Chanel

Figure III.36: Encrypted Image by DIP-GA with The Fitness Function of Lena and Its RGB Histogram

III-4 Image Quality metrics results of tests

Quality metrics results yielded by the proposed method compared to RC4 ,LSB,DIP , GA

DIP-GA without using fitness and DIP-GA wth using fitness

Method	Image Size	*MSE*	*PNSR*	*AD*	*MD*	*PMSE*	*NK*	*SC*	*LMSE*	*NAE*	*UACI*	*NPCR*
RC4	8	3455.5	0.00201	-6.752	205.0	0.06100	0.95604	0.87806	2.22	0.403	32.3	99.18
	16	3857.5	0.00185	-11.50	187.0	0.06533	0.96948	0.83081	-5.516	0.445	35.87	99.36
	32	3583.9	0.00195	-8.383	190.0	0.06327	0.95917	0.86305	-9.93	0.42	33.52	99.48
	64	3711.7	0.00188	-11.94	195.0	0.05799	0.97343	0.83288	2.2704	0.437	35.44	99.44
	128	3508.3	0.00199	-10.64	200.0	0.05941	0.9730	0.84832	-7.728	0.4126	33.34	99.33
	256	3619.6	0.00189	-13.36	216.0	0.05791	0.98366	0.822	2.365	0.4328	35.56	99.32
Conventional LSB	8	3534.24	0.0019	3.7085	210.0	0.0556	0.8800	1.0477	1.2929	0.3764	32.53	99.33
	16	3514.13	0.002	4.0471	182.0	0.054	0.8792	1.05427	-4.709	0.372	32.37	98.63
	32	3598.02	0.00199	3.6295	187.0	0.06	0.8772	1.0494	-1.739	0.3803	31.97	98.72
	64	3580.3	0.00199	2.3701	196.0	0.0577	0.8852	1.0332	-2.553	0.3794	32.45	98.97
	128	3528.05	0.00201	1.0821	192.0	0.056	0.8921	1.0219	8.412	0.3755	32.46	98.99
	256	3493.08	0.00202	1.5930	196.0	0.055	0.88922	1.0306	-6.336	0.3734	32.39	98.99
Projection based LSB	8	4677	0.0017	-0.606	224.0	0.0652	0.8701	1.0592	-3.676	0.388	34.8	99.58
	16	4985	0.0016	-0.442	228.0	0.0610	0.868	1.0609	-1.852	0.3882	34.46	99.6

	32	5175	0.0016	-2.095	238.0	0.062	0.866	1.0425	1.807	0.390	35.90	99.62
	64	5491.	0.0015	-2.807	218.0	0.058	0.8795	1.0234	1.509	0.3820	36.5	99.67
	128	5879.	0.0014	-2.527	225.0	0.055	0.892	1.037	5.6877	0.375	37.2	99.75
	256	4677	0.0017	-0.606	224.0	0.054	0.887	1.06	-3.677	0.3728	37.75	99.72

GA	8	3748.45	0.0040	0.0	198.0	0.060	0.89199	1.0	-3.15	0.3920	21.61	96.32
	16	3663.12	0.0042	0.0	213.0	0.056	0.89558	1.0	7.7951	0.3825	21.28	95.87
	32	3721.21	0.0041	0	190.0	0.058	0.89232	1.0	-5.255	0.3919	21.34	96.78
	64	3545.75	0.0043	0.0	195.0	0.055	0.89904	1.0	-4.630	0.3788	20.82	96.68
	128	3411.16	0.0045	0.0	172.0	0.065	0.90404	1.0	4.0584	0.3720	20.65	99.12
	256	3270.86	0.0046	0.0	171.0	0.063	0.90744	1.0	3.3627	0.3598	20.06	95.37
DIP-GA Without Using the Fitness	8	3802	0.0018	5.3076	224.0	0.059	0.86340	1.07	4.4265	0.3858	32.8	99.61
	16	4830.	0.0017	-0.606	224.0	0.074	0.84775	1.04	2.7120	0.4575	34.63	99.60
	32	5029	0.0016	-0.920	226.0	0.077	0.83378	1.06	-5.921	0.4690	36.12	99.66
	64	5143	0.0016	-2.095	238.0	0.079	0.83384	1.05	9.4897	0.4740	36.52	99.63

	128	5490	0.0015	-2.808	218.0	0.084	0.82154	1.07	1.5145	0.4873	37.28	99.74
	256	5868	0.0014	-2.527	225.0	0.090	0.802	1.10	-1.561	0.5055	37.77	99.72
DIP-GA with The Fitness Function	8	3605	0.0018	0.4607	197.0	0.059	0.895	1.00	-9.339	0.3849	34.92	99.60
	16	4639	0.0017	-2.493	224.0	0.071	0.8650	1.01	2.7234	0.4550	35.13	99.62
	32	4803	0.0016	-2.508	225.0	0.073	0.8513	1.03	-2.577	0.4645	36.08	99.67
	64	5072	0.0016	-2.529	238.0	0.077	0.840	1.04	9.8699	0.4735	36.23	99.682
	128	5291	0.0015	-2.521	216.0	0.081	0.829	1.06	-5.5	0.4788	36.91	99.72
	256	5643	0.0015	2.2818	223.0	0.086	0.793	1.16	-7.039	0.4762	35.39	99.73

Tableau III.4 : representing the results of quality measures tests

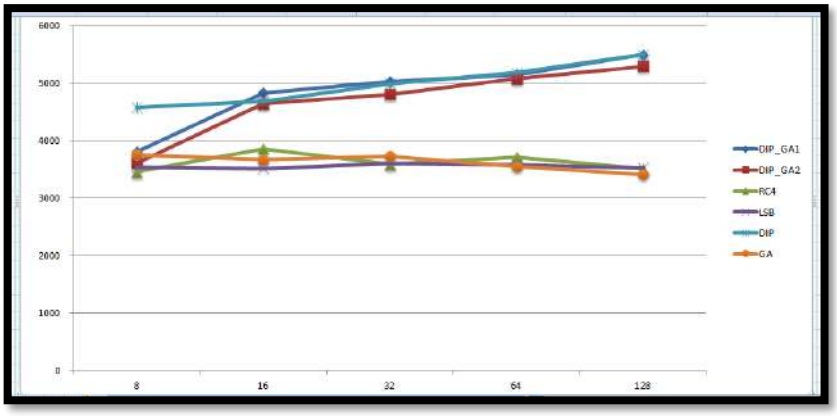


Figure III.37: Curve representing the results of MSE tests as a function of image size

From the Figure III.37: Curve representing the results of MSE tests as a function of image size is clear that the Mean Squared Error (MSE) value is very great values for all experi-

ments. And he scored the most value in algorithm DIP_GA with fitness and DIP_GA without fitness and DIP Which indicates that it is more effective.

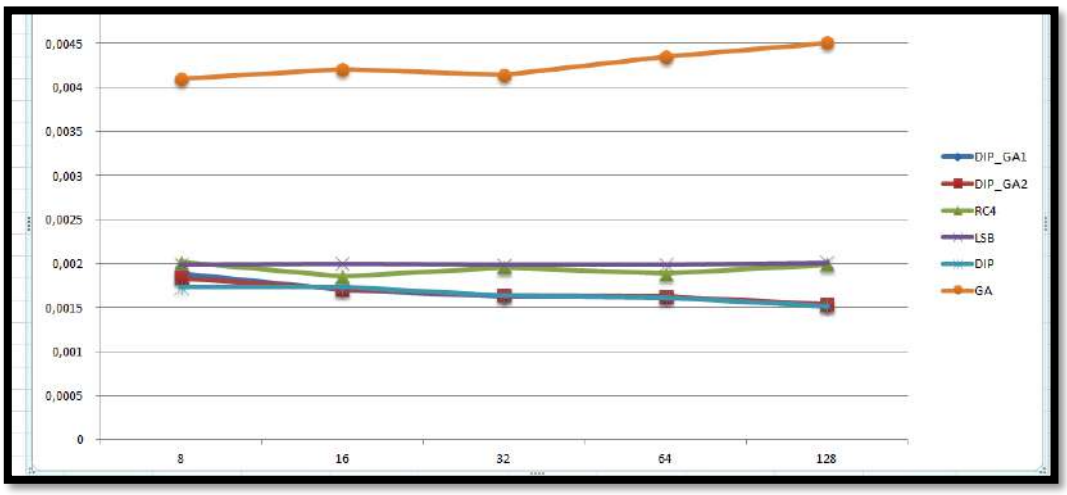


Figure III.38:Curve representing the results of PSNR tests as a function of image size

From the Figure III.38: Curve representing the results of PNSR tests as a function of image size is clear that the Peak Signal to Noise Ratio

(PNSR) value range between 0.0015 dB to 0.004dB That is, it is value is very low for all experiments low value implies good performance in cryptograph. The lowest value was scored in the algorithm DIP_GA with fitness and DIP_GA without fitness and DIP indicating that it is more effective.

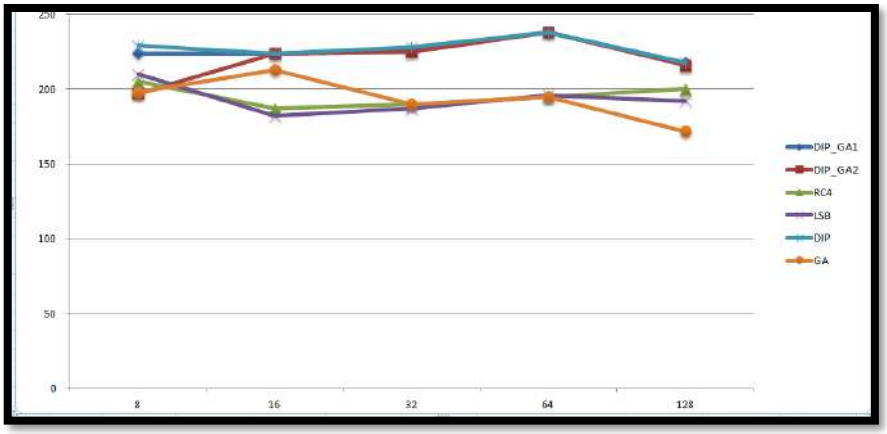


Figure III.39: curve represents the change in MD values in terms of image resizing

The curve represents the change in MD values in terms of image resizing. The maximum difference is used to measure the cover quality of the original image of the encoded im-

age. Note that all values are high. A high value means good performance in the encryption process

And he scored the most value in algorithm DIP_GA with fitness and DIP_GA without fitness and DIP Which indicates that it is more effective.

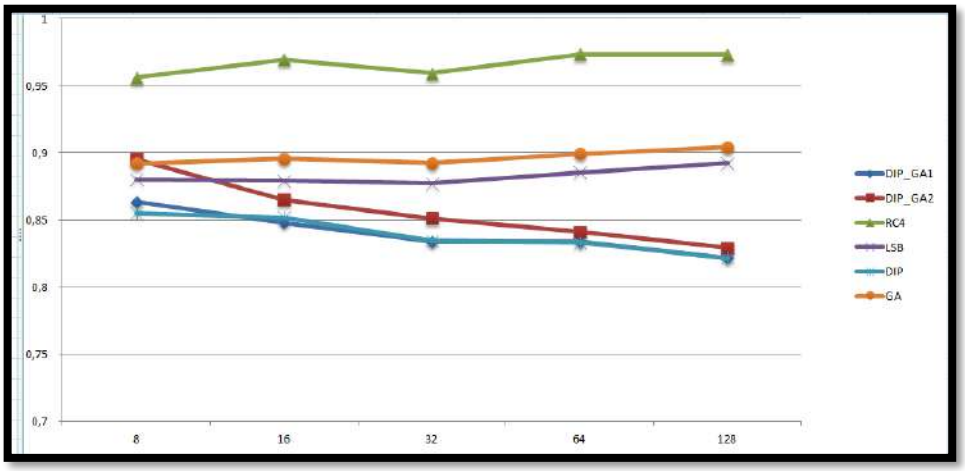


Figure III.40:Curve the change in NK values in terms of image resizing

he curve represents the change in NK values in terms of image resizing. We note that all normal NK values in the proposed algorithms are less than 1, which means that the similarity between the original and the encoded image is low, and we note that the lowest value was recorded using the DIP -GA 2, DIP-GA 1 and DIP algorithms, which means that it is more effective in Obfuscation.

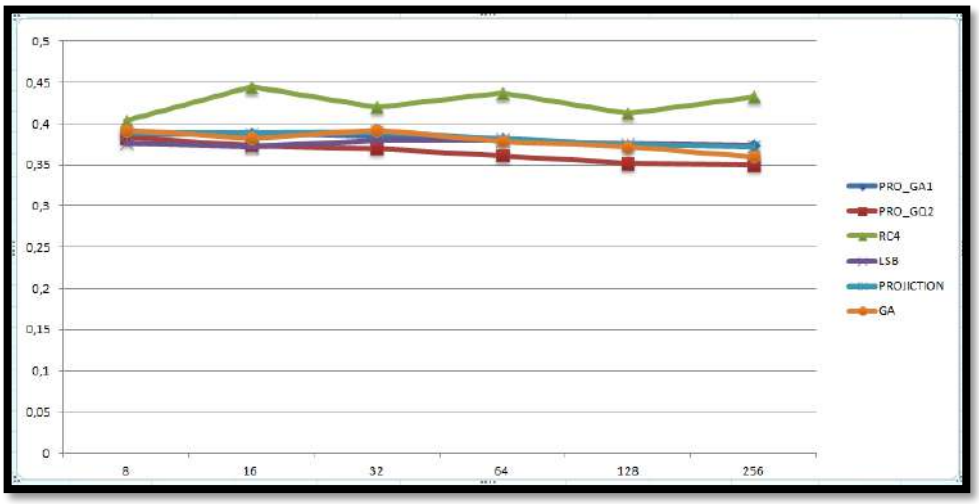


Figure III.41: Curve representing the change in NAE values in terms of image resizing

The curve represents the change in NAE values in terms of image resizing note that all normal absolute error (NAE) values range from 0.35 dB to 0.45 dB That is, the results are high,

and this indicates that the results are good for all the proposed algorithms.

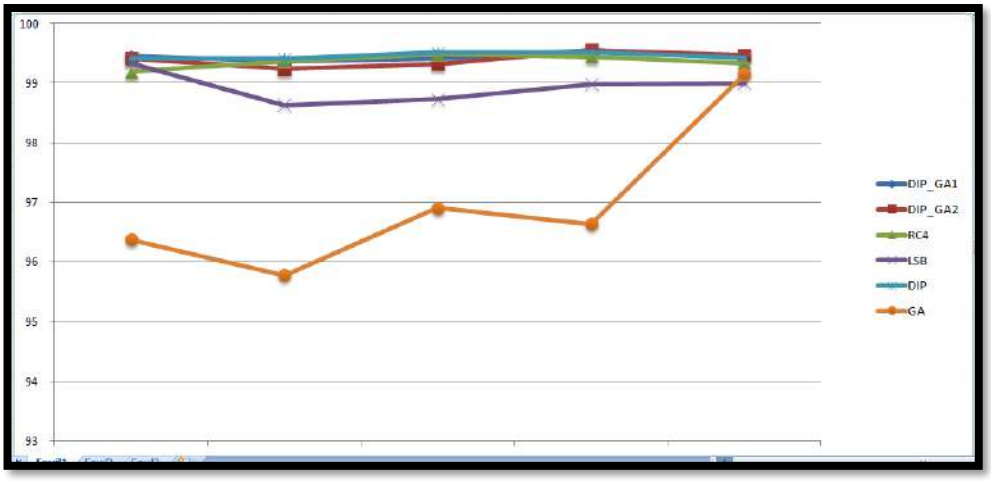


Figure III.42: Curve representing the change in PCNR values in terms of image resizing

The curve represents the change in **PCNR** values in terms of image resizing. We note that all normal **PCNR** values in the proposed algorithms are between 95 and 99, which means that the similarity between the original image and the encoded image is low, and we note that near-cortical values were recorded in DIP, RC4, DIP -GA 2, and DIP-GA 1, which It means that it is more effective in jamming than the other.

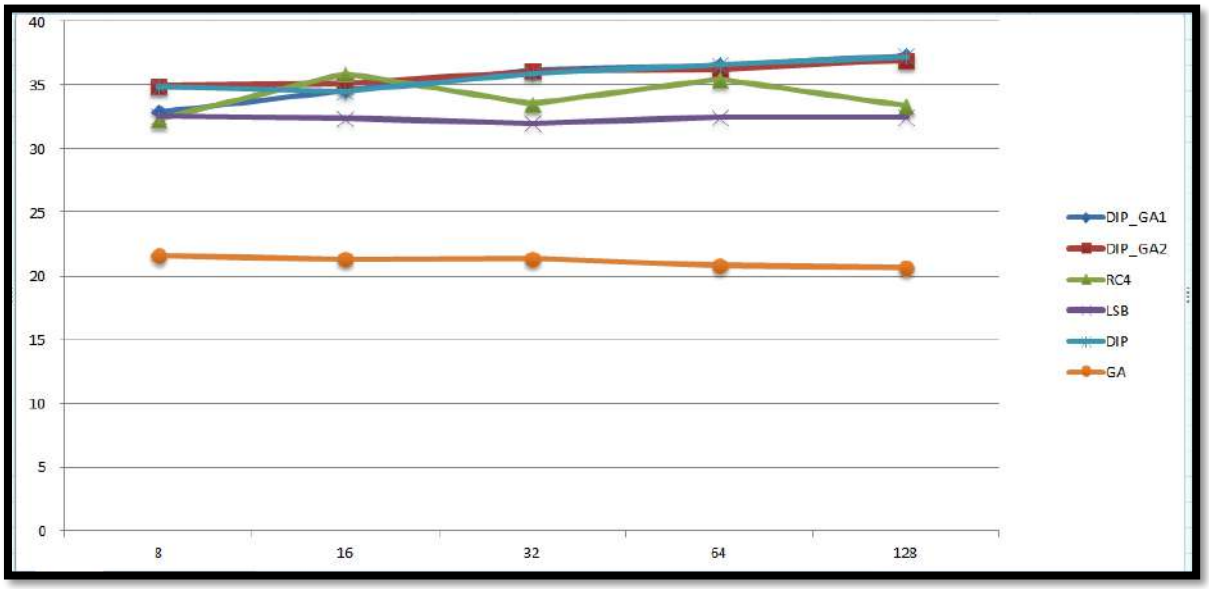


Figure III.43:Curve representing the results of UACI in terms of the change in image size

The curve represents the change in UACI values in terms of the change in image size we observed that the mean values of difference (UACI) between the two pixel values specified for the original image and the encoded image in the DIP, DIP-GA1 and DIP-GA 2 algorithms are greater than 35 and this indicates, our coding system is very sensitive to small changes in the plain text.In GA it was less than 23, which indicates that the algorithm does not change the

house order .The price for the RC4 L, SB is between 34 and 32, which is close to the ideal value

III-5. Average time of Encryption and Decryption

The executing time of encryption and decryption process by proposed approaches based on genetic algorithms has been implemented and conducted using NetBeans IDE 8.2(JDK 9) in Intel Core i7 (6th Gen) 6600U / 2.6 GHz with Windows-10 operating system. In this case, four separate images having the different type's formats of sizes 52*52pixels, 147*147pixels and 208*208pixels type BMP and 220*220 pixels, 310*282 pixels type JPG have been used to measure the encryption time for each image by using the suggested approaches. The encryption time obtained using these images are given in tables **III.6**

Image /Approach	**RC4**	**LSB**	**DIP**	**GA**	**DIP-GA Without Fitness**	**DIP-GA with The Fitness**
8BMP	676	158	145	388	436	5096
64BMP	2918	555	388	920	1126	54931
128BMP	6706	784	680	1399	1762	209270
9 JPG	8025	932	720	1516	2000	2444236
21JPG	18528	1330	1217	1648	2812	1221676

Tableau III.5 Average time to encryption and decryption images using Applicable algorithms

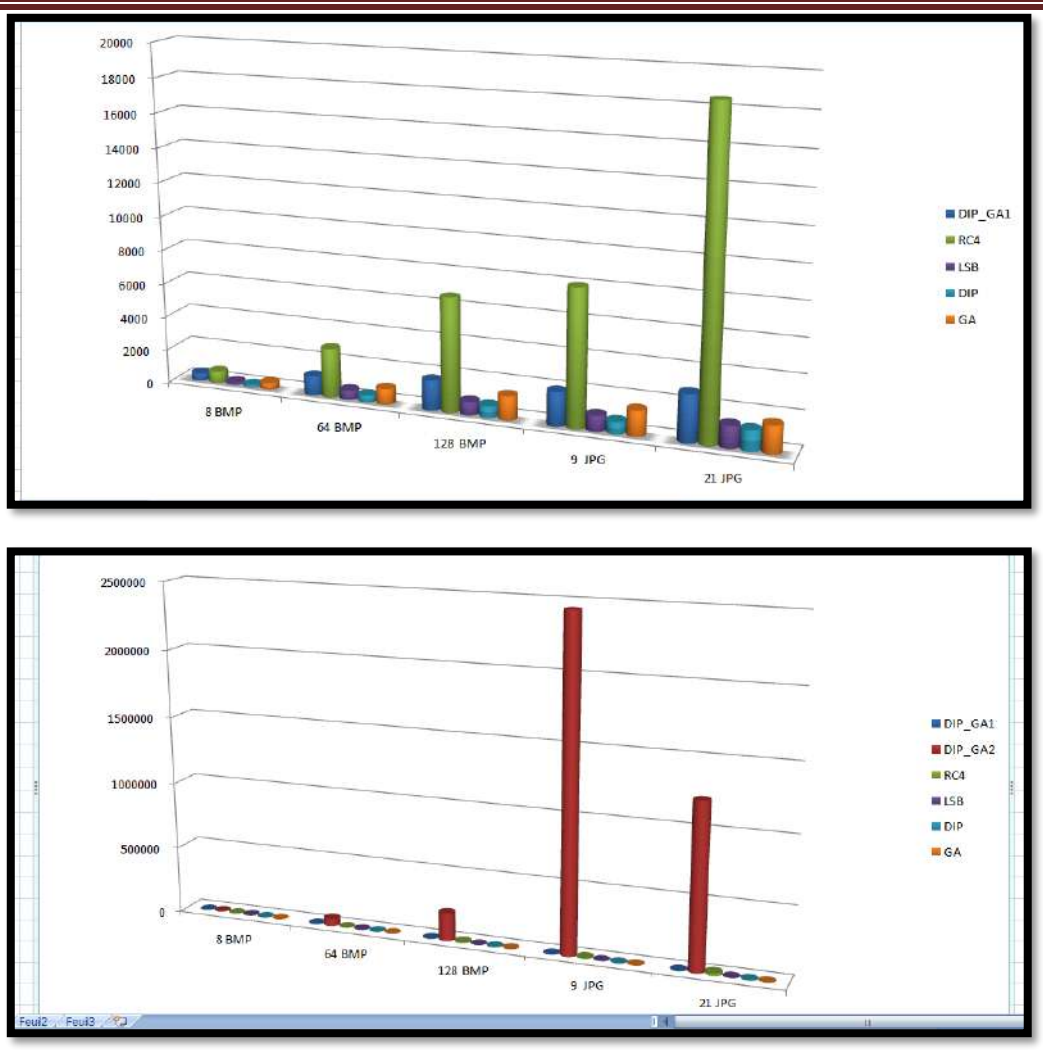


Figure4.12 Average time of encryption and decryption images by using genetic algorithm

Note that the DIP-GA algorithm takes more time to execute then RC4 The DIP is the one that needs the least time to be executed

6. conclusion

In this chapter, we presented good results with the algorithms that we can explain in the second episode of watching long series and engagement

The largest percentage of the watermark, the image value, the distance and the larger the image size, the minimum data loss, and the maximum distortion, from the medium watermarks, it is difficult to discover the points we have hidden in, and draw the image retrieval.

We proved from the graph results that not all algorithms can retrieve the encrypted images

In this chapter, we presented good results with the algorithms that we can explain in the second episode of watching long series and engagement

The largest percentage of the watermark, the image value, the distance and the larger the image size, the minimum data loss, and the maximum distortion, from the medium watermarks, it is difficult to discover the points we have hidden in, and draw the image retrieval.

We proved from the graph results that not all algorithms can retrieve the encrypted images.

From the calculations of various hidden measures such as (mean square error) MSE, (peak

Signal to Noise Ratio (PSNR) NK (mean difference) AD,

(Structural Content) SC, (Maximum Difference) MD and (Standard Absolute Error) NAE, PMSE and LMSE

Conclusion that a file

Perform the DIP-GA scheme with the fitness function and GA without using the LSB projection-based fitness function

We concluded that the traditional DIP-GA with fitness function is the best of all in terms of performance

GENERAL

CONCLUSION

This work has given us the opportunity to discover a new domain and a new way to encrypt to achieve our goals.The study of this project consisted of three chapters, After we presented in the first chapter a briefly described of the basic components of encryption and evolutionary algorithms, The process of implementing the proposed algorithms is addressed in second chapter, And also after the detail in third Chapter, devoted to compare those algorithms, We have led to several conclusions.

In short, it is important to mention that we were interested in reaching satisfactory results here.By successfully using all the algorithms proposed in Chapter second, We concluded that the DIP projection-based algorithm proved effective in encrypting images. We noticed that the best key to achieving optimal results is to export 50% of the encryption points using a specific angle and distance depending on the image dimensions, We also concluded using tests and analysis that all proposed algorithms proved their strength in performance, Especially the DIP algorithm with the genetic algorithm, where the DIP algorithm has improved significantly by getting the best values, For example, PNSR≈0.0015, MSE≈5642, NPCR≈99.74, and this in calculating fitness, But the problem of time remains, so that it takes more time than its predecessors, So it turns out that if you are looking for excellent performance in less time, you'll need to DIP with GA without calculating fitness.

We have achieved the established goals since the beginning of our project,We have succeeded in developed algorithms that provide the best innovative, efficient encryption and safer solution.

REFERENCES

1. A. Clark: Modern optimisation algorithms for cryptanalysis. In Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, p. 258-262.
2. A. Hoffmann, Artificial and Natural Computation, University of New South Wales, Sydney, New South Wales, Australia, in 2001. [img]
3. A. Naveed, Y. Saleem, A. Nisar, A. Rafiq: PERFORMANCE EVALUATION AND WATERMARK SECURITY ASSESSMENT OF DIGITAL WATERMARKING TECHNIQUES, ISSN. 1013-5316, March 2015.
4. A.N. Soo, S. Ghazali, A. Rozniza, A. Andrew: The use of least significant bit (LSB) and knight tour algorithm for image steganography of cover image, International Journal of Electrical and Computer Engineering (IJECE), ISSN: 2088-8708, Vol 9, No 6, December 2019, p. 5218-5226.
5. A. Slowik, H. Kwasnicka: Evolutionary algorithms and their applications to engineering problems, in 16 March 2020.
6. A. Srna, BSc: Selective Encryption Methods for Securing Multi-Resolution Smart Meter Data, Thesis at Salzburg University of Applied Sciences, Bowling Green, Ohio, July 2013, p18
7. B. Delman: Genetic Algorithms in Cryptography, A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, New York July 2004.
8. B. Schneier : Cryptographie appliquée: protocoles, algorithmes et codes source en C, Vuibert informatique, ISBN. 2841800369, 9782841800360, 1996.
9. B. Wyseur: White-Box Cryptography, Ph.D graduation presentation, Heverlee, March 5, 2009, p4.
10. C. Paar, J. Pelzl: Understanding Cryptography, A Textbook for Students and Practitioners, ISBN 978-3-642-44649-8, in 2010.
11. C.P. Pfleeger, S.L. Pfleeger, J. Margulies : Security in Computing, in janv 2015.
12. C. Sasi varnan, A. Jagan, J. Kaur, D. Jyoti, D.S. Rao: Image Quality Assessment Techniques pn Spatial Domain, ISSN : 0976-8491, Vol. 2, Issue 3, September 2011.
13. D. Bhavana: Design and Implementation of a Customized Encryption Algorithm for Authentication and Secure Communication between Devices, The University of Toledo, in August 2017, p4. D. Chaum, R.L. Rivest, AT. Sherman: ADVANCES IN CRYPTOLOGY, Proceedings of Crypto 82, New York in August 1983.
14. D. Kahn : THE Codebreakers, The Story of Secret Writing, 1301 Avenue of the Americas, New York, Feb 1973.
15. F. Battaglia: EVOLUTIONARY COMPUTATION METHODS AND THEIR APPLICATIONS IN STATISTICS, Department of Statistics, Sapienza University, Rome, Italy, , in 2009.

16. F.G. Mohammadi, M.H. Amini, H.R. Arabnia: Evolutionary Computation, Optimization and Learning Algorithms for Data Science, arXiv:1908.08006, vol 1, 16 Aug 2019.
17. F. IDIRI: Algorithmes de chiffrement asymétrique à base de factorisation : mise en œuvre de RSA, Rabin et Paillier, Universit´e A/Mira de Bejaia, in 2015-2016.
18. From Dan's Tool, MD5 Hash Generator, 2019, https://www.md5hashgenerator.com.
19. F. Shameem, M. Seshashayee: Experimental Study of Image Segmentation Using K-Means Clustering Algorithm with Image Quality Metrics, International Journal of Engineering Science Invention (IJESI), ISSN. 2319 – 6734, Vol 7, Issue 10, Oct 2018, PP 37-43.
20. H. Aguirre, H. Okazaki, Y. Fuwa: An Evolutionary Multiobjective Approach to Design Highly Non-linear Boolean Functions, In Proceedings of the Genetic and Evolutionary Computation Conference GECCO'07, in 2007, p. 749–756.
21. https://fr.wikipedia.org/wiki/Contr%C3%B4le_de_redondance_cyclique, 26 mars 2020.
22. I. Salnikov: Ciphers from ancient times to the present day, in Feb 2013, http://historyofciphers.blogspot.com.
23. J.A. Clark, J. Jacob, S. Maitra, P. Stanica: Almost Boolean functions, the design of Boolean functions by spectral inversion, in 2003, vol 3, p. 2173–2180.
24. J. Clark, J. Jacob: Two-Stage Optimisation in the Design of Boolean Functions, Information Security and Privacy, vol 1841, of Lecture Notes in Computer Science, Springer Berlin Heidelberg in 2000, p. 242–254.
25. J.A. Clark, J.L. Jacob, S. Stepney, S. Maitra, W. Millan: Evolving Boolean Functions Satisfying Multiple Criteria, In Progress in Cryptology - INDOCRYPT 2002, p. 246–259.
26. J. Edney, W.A. Arbaugh, W. Arbaugh: Real 802.11 Security, Wi-Fi Protected Access and 802.11i, Addison Wesley 2003. J.F. Miller: Cartesian Genetic Programming, University of New York, June 2003.
27. J. Fran¸ C. Blanchette: Cryptographic culture and evidence law in the age of electronic documents, MIT Press, in 2012.
28. J. McLaughlin, J.A. Clark: Evolving balanced Boolean functions with optimal resistance to algebraic and fast algebraic attacks, maximal algebraic degree and very high nonlinearity. Cryptology ePrint Archive, Report in 2013.
29. K. David: The Codebreakers, ISBN 978-0-684-83130-5, in 1967, p6.
30. K.Thung: A survey of image quality measures, Conference: Technical Postgraduates (TECHPOS), 2009 International Conference, in January 2010.
31. K. Vivek, S.A. Vivek: ACM Ubiquity, Elliptic Curve Cryptography, Vol 9, Issue 20, in May 2008, p1
32. K. Vivek, S.A.Vivek: Elliptic Curve Cryptography, National Informatics Centre Government Of India, ACM Ubiquity, Vol 9, Issue 20, May 20 – 26,in 2008.

33. L. Burnett, W. Millan, E. Dawson, A. Clark: Simpler methods for generating better Boolean functions with good cryptographic properties, Australasian Journal of Combinatorics 29, in 2004, p. 231–247.
34. L.D. Burnett: Heuristic Optimization of Boolean Functions and Substitution Boxes for Cryptography, PhD thesis, Queensland University of Technology, in 2005.
35. L. Mariot, A. Leporati: A Genetic Algorithm for Evolving Plateaued Cryptographic Boolean Functions, In Theory and Practice of Natural Computing - Fourth International Conference, TPNC 2015, Mieres, Spain, December 15-16, 2015, p. 33–45.
36. L. Mariot, A. Leporati: Heuristic Search by Particle Swarm Optimization of Boolean Functions for Cryptographic Applications, In Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, p. 1425–1426.
37. L.R. Knudsen, M.J.B. Robshaw: The Block Cipher Companion, ISSN 1619-7100, 2011, p.152.
38. L. Stosic, M. Bogdanovic: RC4 stream cipher and possible attacks on WEP, Article in International Journal of Advanced Computer Science and Applications, in March 2012.
39. M. Dacier : Vers une évaluation quantitative de la sécurité informatique, Réseaux et télécommunications [cs.NI], Institut National Polytechnique de Toulouse - INPT, 1994, Français.
40. M. Hattim: An efficient image cryptography using hash-LSB steganography with RC4 and pixel shuffling encryption algorithms, Conference: 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), March 2017.
41. M. Rouse: RSA algorithm (Rivest-Shamir-Adleman), Accessed May 12, 2017, http://searchsecurity.techtarget.com/definition/RSA
42. M. Wan Azani, H. Yazid, J. Mastura, M. Zainal, A.S. Abdul-Nasir, M. Noratikah: A Review of Image Quality Assessment (IQA): SNR, GCF, AD, NAE, PSNR, ME, Journal of Advanced Research in Computing and Applications, ISSN: 2462-1927, June 2017.
43. M.W. Munir: Cryptography, Technical Report, Feb 2005
44. P.N. Suganthan: Differential Evolution Algorithm, Recent Advances, School of Electrical and Electronic Engineering, Nanyang Technological University.
45. R. Das: Performance Analysis of IoT Security scheme employing an Integrated Approach of Cryptography and Steganography, December 15, 2017.
46. R. Hrbacek, V. Dvorak: Bent Function Synthesis by Means of Cartesian Genetic Programming, In Bartz-Beielstein, Parallel Problem Solving from Nature - PPSN XIII, vol 8672, in 2014, pages 414–423.
47. R. Spillman: Cryptanalysis of knapsack ciphers using genetic algorithms, October 1993, p.367-377.

48. R. Van Rijswijk, M. Oostdijk: Applications of Modern cryptography Technologies, applications and choices (2010), https://vdocuments.mx/applications-of-modern-cryptography.html, post on Feb 2017.
49. S.A. Vanstone, P.C. Van Oorschot, A.J. Menezes, Handbook of Applied Cryptography, CRC Press, in 1996.
50. S. Kavut, M. Yucel: Improved Cost Function in the Design of Boolean Functions Satisfying Multiple Criteria, Progress in Cryptology - INDOCRYPT 2003, Springer Berlin Heidelberg, vol 2904, p.121–134.
51. S. Picek, C. Carlet, D. Jakobovic, J.F. Miller, L. Batina: Correlation Immunity of Boolean Functions, An Evolutionary Algorithms Perspective, In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, p. 1095–1102.
52. S. Picek, C. Carlet, S. Guilley, J. Miller, D. Jakobovic: Evolutionary Algorithms for Boolean Functions in Diverse Domains of Cryptography.
53. S. Picek, D. Jakobovic, J.F. Miller, E. Marchiori, L. Batina: Evolutionary Methods for the Construction of Cryptographic Boolean Functions, In Genetic Programming - 18th European Conference, EuroGP 2015, Copenhagen, Denmark, April 8-10, 2015, p. 192–204.
54. S. Picek, D. Jakobovic, J.F. Miller, L. Batina, M. Cupic: Cryptographic Boolean functions, One output, many design criteria, in 2016, p. 635–653.
55. S. Picek, D. Jakobovic, M. Golub: Evolving Cryptographically Sound Boolean Functions, p. 191–192, New York, NY, USA, ACM, in 2013.
56. S. Picek, E. Marchiori, L. Batina, D. Jakobovic: Combining Evolutionary Computation and Algebraic Constructions to Find Cryptography-Relevant Boolean Functions, In Parallel Problem Solving from Nature - PPSN XIII - 13th International Conference, Ljubljana, Slovenia, in September 2014, p. 822–831
57. S. Picek, L. Batina, D. Jakobovic: Evolving DPA-Resistant Boolean Functions, In Parallel Problem Solving from Nature - PPSN XIII - 13th International Conference, Ljubljana, Slovenia, in September 2014, p. 812–821.
58. S. Picek, S. Guilley, C. Carlet, D. Jakobovic, J.F. Miller: Evolutionary Approach for Finding Correlation Immune Boolean Functions of Order t with Minimal Hamming Weight, In Theory and Practice of Natural Computing - Fourth International Conference, TPNC 2015, Mieres, Spain, December 15-16, 2015, p. 71–82.
59. S.R. TEERTH : Encryption and Decryption Image Using Multiobjective Soft Computing Algorithm, Thesis in MARATHWADA UNIVERSITY, NANDED, 2015.
60. T. Jawahar, K. Nagesh: DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis, Department of Computer Science, Himachal Pradesh University, Shimla, INDIA, ISSN 2250-2459, Vol 1, Issue 2, in December 2011. p2.
61. W. Millan, J. Fuller, E. Dawson: New concepts in evolutionary search for Boolean functions in cryptology, Computational Intelligence, in 2004, p. 463–474

62. W. Terence: The Code for Gold, Edgar Allan Poe and Cryptography, Representations, University of California Press (1994), 35–57.
63. Y. Izbenko, V. Kovtun, A. Kuznetsov: The design of boolean functions by modified hill climbing method, Cryptology ePrint Archive, Report in 2008.