PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

M'HAMED BOUGARA UNIVERSITY –BOUMERDES

FACULTY OF TECHNOLOGY

Electrical Systems Engineering Department

## Master's thesis

Presented by:

**Mr BOUKANDOURA Omar**

**Mr BOULAHIA Chemse Eddine**

Field of Study: Telecommunications

Major: Networks and Telecommunications

## TITLE:

# Image Encryption Algorithm Based on Rubik's Cube Principal

**Defended on 12 /07 /2023 in front of the jury composed of:**

| | | | | |
|---|---|---|---|---|
| Noureddine | MESSOUDI | MCA | University of Boumerdes | President |
| Samia | BELKACEM | MCA | University of Boumerdes | Supervisor |
| Karima | KESSAISSIA | MAA | University of Boumerdes | Examiner |

# *Acknowledgement*

*First and foremost, we would like to express our profound gratitude to Allah, The most Gracious and the Most Merciful for His blessings and guidance throughout this project. It is through His grace that we have accomplished this significant milestone, and we are humbled by His unwavering support.*

We express our deepest gratitude to Madame. BELKACEM, our esteemed supervisor, for her outstanding commitment, support, and unwavering faith in our capabilities, which have played a vital role in our achievements. We are profoundly appreciative of her invaluable mentorship, perceptive input, and constant inspiration, which have greatly influenced the development of this project into its current form.

We would also like to express our heartfelt appreciation to the University of Boumerdes, Faculty of Technology, and the team at the Incubator of Boumerdes. The nurturing atmosphere and abundant resources provided by our institution have played a pivotal role in fostering our development and education. We are sincerely thankful for the opportunities it has bestowed upon us in our respective field.

Finally, we would like to convey our profound appreciation to all those who have placed their trust in us. We are genuinely grateful to everyone who has offered assistance, kind words, or even the slightest gestures of support, as each contribution has played a significant role in our achievements.

# Dedication

*We dedicate this work, as a sign of respect, gratitude, and recognition to:*
*To our dearest parents, for their assignment, patience, prayers and their sacrifices that they endured. May Allah guard and protect them.*

*To our dear friends Les Zaki's, Aymen, Yacine, Les Oussama's and special dedication for Nessrine, Thanks for staying a true friend.*

*To all the people dear to my heart, for their help, their time, their encouragement, their assistance and support;*

*Not to mention all the teachers whether primary, middle, secondary or higher education;*

*Last but not least, we want to thank us, for believing in us, for doing all this hard work, we want to thank us for doing right more than wrong, we want to thank us for being us all the time*

# Table of content

# *List of abbreviations*

**ADC**: Analog-to-Digital Converter

**AES**: Advanced Encryption Standard.

**CBC**: Cipher Block Chaining.

**CFB**: Cipher Feedback.

**CMYK**: Cyan, Magenta, Yellow, Key (Black)

**CPU**: Central Processing Unit

**CT**: Computed Tomography

**CTR**: Counter Mode

**DES**: Data Encryption Standard.

**DIP**: Digital Image Processing

**ECB**: Electronic Codebook

**EEG**: Electroencephalography

**GCD**: Greatest Common Divisor

**GIF**: Graphics Interchange Format

**HDR**: High Dynamic Range

**IBM**: International Business Machines Corporation.

**IDEA**: International Data Encryption Algorithm.

**IEEE**: Institute of Electrical and Electronics Engineers.

**ISCR**: Input Signal-to-Change Ratio

**IV**: Initialization Vector.

**LAN**: Local Area Network

**LFSR**: Linear Feedback Shift Register.

**MD2-4-5**: Message Digest algorithms 2, 4, and 5.

**NASA**: National Aeronautics and Space Administration

**NPCR**: Normalized Pixel Change Ratio

**OFB**: Output Feedback.

**PET**: Positron Emission Tomography

**PNG**: Portable Network Graphics

**PRNG**: Pseudorandom Number Generator.

**PRP**: Pseudorandom Permutation

**PSNR**: Peak Signal-to-Noise Ratio

**RC5**: Rivest Cipher 5.

**RGB**: Red, Green, Blue

**RGB**: Red, Green, Blue

**RSA**: Rivest, Shamir, Adleman

**SHA**: Secure Hash Algorithm.

**TIFF**: Tagged Image File Format

**UACI**: Unified Average Change Intensity

**UV**: Ultraviolet

**XOR**: Exclusive OR. It is a logical operation that outputs true (1) only when the number of true inputs is odd.

# *List of Figures*

# *List of tables*

## ملخص

في عصر التقدم الرقمي السريع، أدى الاستخدام الواسع النطاق للصور التي تحتوي على معلومات حساسة إلى ظهور متطلبات ماسة لاتخاذ تدابير أمنية صارمة وتقنيات تشفير متقدمة .في هذه أطروحة يستكشف التطبيق المبتكر لمبدأ مكعب روبيك في تشفير الصور .مستوحاة من اللغز الشهير، توفر خوارزمية Rubik's Cube إطارا فريدا لمعالجة البيانات وتحويلها .من خلال الاستفادة من هذا المبدأ، يتم اقتراح طريقة تشفير صور قوية وآمنة للغاية .يتضمن النهج المقترح تطبيق تحويلات مكعب روبيك على قيم البكسل ، وخلط بيانات الصورة من خلال التناوب والتباديل .تقدم هذه العملية التعقيد، مما يجعل فك التشفير غير المصرح به بدون المفتاح الصحيح مستحيلا تقريبا .الميزة الرئيسية لهذه الطريقة هي قابليتها للانعكاس، مما يتيح إعادة بناء الصورة الأصلية بأمانة أثناء فك التشفير .يضيف دمج مبدأ مكعب روبيك طبقة إضافية من التعقيد وعدم الخطية، مما يعزز أمان الصورة المشفرة.



**الكلمات المفتاحية :تشفير الصور، التشفير، مكعب روبيك، الأمان، الصور الرقمية.**

*Abstract*

In the era of rapid digital advancement, the pervasive use of images containing sensitive information has given rise to a critical requirement for stringent security measures and advanced encryption techniques. In This master's thesis the application of the Rubik's Cube principle in image encryption is explored. The Rubik's Cube algorithm provides a unique framework for data manipulation and transformation. By leveraging this principle, a robust and highly secure image encryption method is proposed. The proposed approach involves applying Rubik's Cube transformations to pixel values, shuffling the image data through rotations and permutations. This process introduces complexity, making unauthorized decryption without the correct key virtually impossible. A key advantage of this method is its reversibility, enabling faithful reconstruction of the original image during decryption. The incorporation of the Rubik's Cube principle adds an additional layer of complexity and nonlinearity, enhancing the security of the encrypted image.

**Keywords: Image encryption, Cryptography, Rubik's Cube, Security, Digital images.**

*Résumé*

À l'ère des progrès numériques rapides, l'utilisation généralisée d'images contenant des informations sensibles a donné lieu à des exigences urgentes de mesures de sécurité strictes et de techniques de cryptage avancées. Dans ce mémoire, l'application du principe du Rubik's cube dans le cryptage d'images est testé. L'algorithme Rubik's Cube fournit un cadre unique pour le traitement et la transformation des données. Profitant de ce principe, une méthode de cryptage d'image très forte et sécurisée est utilisée. L'approche proposée consiste à appliquer les transformations du Rubik cube aux valeurs de pixels, en mélangeant les données d'images par rotation et permutations. Ce processus introduit de la complexité, rendant le décryptage non autorisé sans la bonne clé presque impossible. Le principal avantage de cette méthode est sa réversibilité, qui permet de reconstruire fidèlement l'image originale lors du décryptage. L'intégration du principe du Rubik's Cube ajoute une couche supplémentaire de complexité et de non-linéarité, améliorant ainsi la sécurité de l'image cryptée.

**Mots clés : Cryptage d'images, cryptographie, Cube de Rubik's, Securité, image numérique**

# *General Introduction*

Security has always been a paramount concern for mankind, and with the advent of computers and telecommunications, the complexity of security issues has intensified. Today, the digital landscape presents a myriad of challenges, including computer viruses, unauthorized access, non-identification, and information forgery. In response to these evolving threats, innovative solutions are required to protect sensitive data. The field of encryption, which plays a crucial role in ensuring the confidentiality, integrity, and authenticity of information, has witnessed significant advancements. *(B.Imane, 2018)*

The digital age has seen a massive proliferation of images, making image encryption an essential aspect of secure data transmission and storage. However, existing encryption techniques have their limitations, necessitating the exploration of novel approaches to enhance security. *(A.Said, 2008)*

In this context, our final project master's thesis aims to protect digital image by encryption process using Rubik's Cube method, and testing their performances.

To this end, our master's thesis is structured into three chapters, each focusing on a critical aspect of image encryption.

Chapter 1 serves as an introduction to our thesis which talks about image processing, emphasizing its significance in the context of encryption. A deep exploration of image representation, compression, and manipulation techniques is undertaken to comprehend the intricacies involved in handling images as data. Additionally, an analysis of the vulnerabilities and challenges specific to image encryption sets the stage for the subsequent chapter.

Chapter 2 give details on image encryption concept, providing a comprehensive overview of the fundamental principles and techniques employed in securing data. By exploring encryption algorithms, cryptographic protocols, and key management systems, this chapter establishes the theoretical foundation necessary for understanding the intricacies of image encryption.

Chapter 3 forms the core of our research, presenting our proposed method of image

encryption based on the principles of the Rubik's Cube. We adapted the Rubik's Cube mechanics and applied it to the digital domain to create a robust and efficient encryption algorithm. This chapter unveils the inner workings of our method, highlighting its strength, security, and computational efficiency.

Throughout our master’s thesis, we will engage in precising analysis and evaluation. We will conduct practical experiments, subjecting encrypted images to different threats, and measuring the resilience of our approach. The results obtained will provide empirical evidence to validate the security and practicality of our proposed image encryption method.

At last, our final project thesis will be closed by a general conclusion.

# *Chapter 1: Image processing*

## 1. Introduction

Image processing is one of the engineering and computer science that are always-evolving fields. Researchers are continuously attempting to develop image processing methods that offer more features, better accuracy, and faster processing speeds.

In this chapter we are going to study images processing through the response to these questions: What are images? what can be understood from an image? How much information can be retrieved from an image? And what sort of applications can be developed from the available information?

## 2. What are images?

In the broadest possible sense, images are pictures, a way of recording and presenting information visually. Pictures are important to us because they can be an extraordinarily effective medium for the storage and communication of information. Consider the familiar example of the photograph. We use photography in everyday life to create a permanent record of our visual experiences, and to help us share those experiences with others. In showing someone a photograph, we avoid the need for a lengthy, tedious and, in all likelihood, ambiguous verbal description of what was seen. This emphasizes the point that humans are primarily visual creatures. We rely on our eyes for most of the information we receive concerning our surroundings, and our brains are particularly adept at visual data processing. *(A. Baskar, 2023)*

Photography is the imaging technique with which we are most familiar, simply because the information it records is similar to that which we receive using our eyes. Both human vision and photography require a light source to illuminate a scene. The light interacts with the objects in the scene and some of it reaches the observer, whereupon it is detected by the eyes or by a camera. Information about the objects in the scene is recorded as variations in the intensity and color of the detected light. A key point is that, although a scene is (typically) three-dimensional, the image of that scene is always two-dimensional. *(A. Baskar, 2023)*

## 3. Types of Images

Images' types are another essential component in the field of image processing. They are as follows:

### 3.1. Binary image

A binary will contain only two colors: white and black. Black is represented by 0 and white is represented by 1. Each pixel in this type of image will have either a value of 0 or 1, representing black or white, respectively (Figure 1.1). In the binary image, each pixel needs only 1 bit of storage space. Be it white or black, what we need is just 1 bit to store that pixel. This is an important aspect to be remembered and this will help in distinguishing the binary image from the black-and-white image. *(A. Baskar, 2023)*

### 3.2. Black-and-white image

Most beginners are confused about what a binary image is and what a black-and-white image is. There is a very fundamental difference that differentiates the two. When it comes to black-and-white images, each pixel needs 8 bits of storage space. Each of these pixels can have a 0 or 1. Again, 0 represents black and 1 represents white. Multiple 0s and 1s are in an image, but the storage requirement for the pixels is much higher. This gives smoothness and enriched quality to the image. *(A. Baskar, 2023)*

*Figure 1. 1 Black & White image*

### 3.3. Grayscale image

The next type of image to be discussed is the grayscale image. It is a special image that has a range of shades from black to white, i.e., the shades should be between white and black. Often

people regard this as no color and they refer to the shades of white and black. The most commonly used format is the 8-bit format and it accommodates 256 different shades. The range of these shades is from 0 (black) to 255 (white) and in-between are the different shades. *(A. Baskar, 2023)*

*Figure 1. 2 Grayscale color scheme*

## 3.4. Color Image

Each pixel in a color image has color information. Each pixel in a color image is composed of three channels, and they are most commonly regarded as R, G, and B, representing the colors red, green, and blue, respectively. In this type of image, one should again visualize the image as a matrix. Each box (pixel) in the matrix is composed of three components: R, G, and B channels. Each of these channels needs 8 bits for storage, hence each pixel is 24 bits. Hence, it should be understood that R, G, and B together constitute a pixel in a color image. The shade of each of the pixels vary based on the intensity of R or G or B. Each of these R or G or B channels individually has 256 shades in it. With all these shades we could produce a beautiful color image with all good colors captured. *(A. Baskar, 2023)*

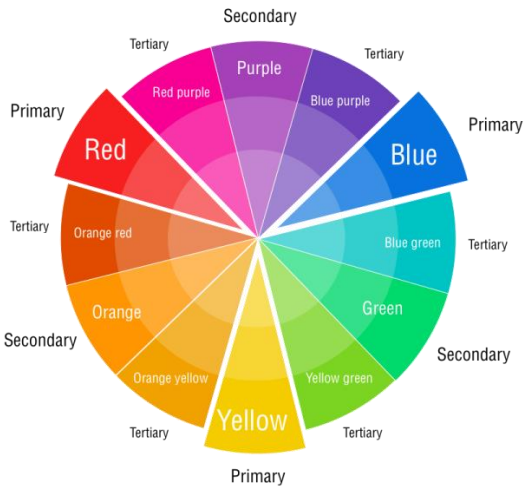



*Figure 1. 3 Color wheel*

## 4. File Formats

Pixels in a digital image can be stored in a variety of formats. The file format that should be used is crucial since we will be changing his contents. Some file formats are lossy, while others are lossless. *(S. Montabone,2010)*

Lossy formats compress the original image, removing certain details of it so that it still looks similar to the human eye but with a smaller file size. This means that some data is lost in the process. The nice thing about them is that the file sizes can get really small. The bad thing is that, in some cases, the image loses too much information and it does not look as good as the original. Also, every time you save in a lossy format, the process is repeated, degrading your image even more.

Lossless formats, on the other hand, encode the original image in a way so that it doesn't lose any details of the original image. They can still compress images, but not as much as the lossy formats. The nice thing about these formats is that they always preserve the full quality of your original image. The bad thing is that they may take more space in your disk than the lossy ones.

These are the most common raster image file formats.

### 4.1. JPG format

JPG, or JPEG (for Joint Photographic Experts Group), is the most common file format in digital photography. The JPEG format is optimized for photographs and similar continuous tone images that contain a large number of colors, and can achieve astonishing compression ratios, even while maintaining a high image quality. The JPEG compression technique analyzes images, removes data that is difficult for the human eye to distinguish, and stores the resulting data as a 24-bit color image. The level of compression used in the JPEG conversion is definable, and photographs that have been saved with a JPEG compression level of up to 15 are often difficult to distinguish from their source images, even at high magnification. *(C.Relf, 2003)*

### 4.2. GIF format

GIF (CompuServe Graphics Interchange File) images use a similar LZW compression algorithm to that used within TIFF images, except the bytes are reversed and the string table is upside-down. All GIF files have a color palette, and some can be interlaced so that raster lines can appear as every four lines, then every eight lines, then every other line. This makes the use of GIFs in a Web page very attractive when slow download speeds may impede the viewing of

an image. Compression using the GIF format creates a color table of 256 colors; therefore, if the image has fewer than 256 colors, GIF can render the image exactly. If the image contains more than 256 colors, the GIF algorithm approximates the colors in the image with the limited palette of 256 colors available. Conversely, if a source image contains less than 256 colors, its color table is expanded to cover the full 256 graduations, possibly resulting in a larger file size. The GIF compression process also uses repeating pixel colors to further compress the image. *(C.Relf, 2003)*

## 4.3. PNG format

The Portable Network Graphics file format is also a lossless storage format that analyzes patterns within the image to compress the file. PNG is an excellent replacement for GIF images, and unlike GIF, is patent-free. PNG images can be indexed color, true color or grayscale, with color depths from 1 to 16 bits, and can support progressive display, so they are particularly suited to Web pages. Most image formats contain a header, a small section at the start of the file where information regarding the image, the application that created it, and other nonimage data is stored. The header that exists in a PNG file is programmatically editable using code developed by a National Instruments' Applications Engineer. You are able to store multiple strings (with a maximum length of 64 characters) at an unlimited number of indices. Using this technique, it is possible to “hide” any type of information in a PNG image, including text, movies and other images (this is the method used to store region of interest (ROI) information in a PNG file). *(C.Relf, 2003)*

## 4.4. TIFF format

TIFF was developed by Microsoft and Aldus in 1986. TIFF is a trademark that was originally registered to Aldus, which subsequently merged with Adobe Systems (San Jose, Calif). Adobe now controls the TIFF specifications copyright. TIFF was created primarily by imaging developers of input and output devices such as printers, monitors, and scanners; as a result, it is specifically designed to be compatible with different image processing devices.

The word “Tagged” in “TIFF” refers to this format’s complicated file structure. The initial header of the file data is followed by “chunks” of data called “tags,” which convey the image information to the program displaying the file. The actual TIFF specifications define over 70 different tag types. This level of complexity allows great Rexibility between viewers; however, pro- grams that interpret TIFF images must contain all the different data for the tags. Although many programs simplify this by implementing only certain tags, skipping some tags could

theoretically affect image quality, and private tags may limit the use of TIFF files for some applications. *(R.Wiggins, 2001)*

### 4.5. OpenEXR format

This format is an open standard for describing high dynamic range images (HDR). The file extension that it uses is EXR. You can use it with lossy or lossless compression. It was created and actively used by Industrial Light & Magic on motion pictures. *(S. Montabone,2010)*

## 5. Image properties

### 5.1. Resolution

The resolution of an image is the number of pixels that are used to represent the image. For example, if the resolution of an image is $2{,}816 \times 2{,}112$, it means that the image has a width of 2,816 pixels and a height of 2,112 pixels. You can also say that the resolution of that image is 6 megapixels (MP) because $2{,}816 \times 2{,}1122$ is 5,947,392, which is roughly 6 million pixels, or 6MP. *(S. Montabone,2010)*

In general, the more pixels used to represent an image, the better it will look. This is true up to a limit where no gain in quality is perceived by the human eye. This limit depends on the use of the image. Today, there are two main uses for digital photographs: displaying them on a computer screen (or a similar device) and printing them on paper (or a similar surface).


*Figure 1. 4 Resolution comparison*

## 5.2. Aspect Ratio

A digital camera's sensors are nowadays measured in MP. This represents the total number of pixels of the resulting image (width x height) but gives no information about its aspect ratio (width to height). In general, point-and-shoot digital cameras have an aspect ratio of 4:3, which is the standard definition in television and monitors (as opposed to high-definition, which has an aspect ratio of 16:9). On the other hand, digital single lens reflex cameras (DSLRs) usually have an aspect ratio of 3:2, which is the same as the 35mm film format. You can easily check the aspect ratio of your camera just by looking at the size of any image that you have taken with it and calculate the width-to-height ratio (take its width and divide it by its height). For example, if the result is 1.3333…, you know that the image has a 4:3 aspect ratio. On the other hand, if the result is 1.5, then you know that your camera produces images with a 3:2 aspect ratio. *(S. Montabone,2010)*

Aspect ratio is particularly important when printing a digital image. There are different standard sizes in print and in digital images, so part of your image may need to be taken away to fit the print standard. *(S. Montabone,2010)*

## 5.3. Color

Color is very important to digital images. In a digital environment, a specific number of bits are used to represent information. This means that we cannot store an infinite number of colors. We need to allocate a certain number of bits to be used to represent color. This number is called color depth and for understanding what those numbers represent, you need to learn about color spaces. Those are the two subjects that I will show you now.*(S. Montabone,2010)*

## 5.4. Color Depth

Color depth is the number of bits that are used to represent the color of each pixel in a digital image. The more bits used; the more unique colors can be used in the image. When more unique colors are used, the image looks more natural to the human eye up to a limit where no difference is noted when increasing the color depth. *(S. Montabone,2010)*

If we use a color depth of only 1 bit, the resulting image will only have 2 possible colors. These two colors can actually be any of the available colors that the computer can reproduce, but usually black and white are used in this scenario. If we increase the color depth to 8 bits, you are able to use up to 256 (28) different colors, which is normally used for representing a

grayscale image or a reduced set of colors. *(S. Montabone,2010)*

Using 24 bits gives you more than 16 million unique colors. When increasing the color depth to more than 24 bits, most humans cannot tell the difference from the original with 24 bits; that is why this color depth is often called TrueColor. *(S. Montabone,2010)*

## 6. Image Processing

Image processing is a general term for the wide range of techniques that exist for manipulating and modifying images in various ways. Photographers and physicists can perform certain image processing operations using chemicals or optical equipment; however, we concern ourselves solely with digital image processing, i.e., that which is performed on digital images using computers. We will consider not only how digital images may be manipulated and enhanced, but also how they may be acquired, stored and represented in computer memory. *(A. Baskar, 2023)*

Digital imaging actually predates modern computer technology; newspaper pictures were digitized for transatlantic transmission via submarine cable in the early 1920s. However, true digital image processing (DIP) was not possible until the advent of large-scale digital computing hardware. The early motivation for the development of DIP techniques came from the space programmed in 1964, NASA's Jet Propulsion Laboratory used computers to correct distortions in images of the lunar surface obtained by the Ranger 7 probe. Now, more than three decades later, DIP finds applications in areas as diverse as medicine, military reconnaissance and desktop publishing. *(A. Baskar, 2023)*

### 6.1. Steps in Digital Image Processing

The first step in digital image processing is image acquisition. We shall start with that.

#### 6.1.1. Image acquisition

The first and foremost step is where to acquire the image. To make it a bit more technical, this stage is where the image is made available after a bit of preprocessing. Most of the time, the acquired image will be digital when it comes out the camera; if not, it should be converted to a digital image with the help of an analog-to-digital converter (ADC). *(B. Jähne,2002)*

#### 6.1.2. Image enhancement

Image enhancement refers to the process of highlighting certain information of an image, as well as weakening or removing any unnecessary information according to specific needs.

For example, eliminating noise, revealing blurred details, and adjusting levels to highlight features of an image. *(dynamsoft.com,2023)*

Image enhancement techniques can be divided into two broad categories:

1) **Spatial domain:** Enhancement of the image space that divides an image into uniform pixels according to the spatial coordinates with a particular resolution. The spatial domain methods perform operations on pixels directly.

2) **Frequency domain**: Enhancement obtained by applying the Fourier Transform to the spatial domain. In the frequency domain, pixels are operated in groups as well as indirectly.

Why Perform Enhancement?

The basic goal of image enhancement is to process the image so that we can view and assess the visual information it contains with greater clarity. Image enhancement, therefore, is rather subjective because it depends strongly on the specific information the user is hoping to extract from the image. *(C,Solomon,2011)*

The primary condition for image enhancement is that the information that you want to extract, emphasize or restore must exist in the image. Fundamentally, and the desired information must not be totally swamped by noise within the image. Perhaps the most accurate and general statement we can make about the goal of image enhancement is simply that the processed image should be more suitable than the original one for the required task or purpose. This makes the evaluation of image enhancement, by its nature, rather subjective and, hence, it is difficult to quantify its performance apart from its specific domain of application. *(C,Solomon,2011)*

1) Enhancement Via Image Filtering

The main goal of image enhancement is to process an image in some way so as to render it more visually acceptable or pleasing. The removal of noise, the sharpening of image edges and the ‘soft focus’ (blurring) effect so often favored in romantic photographs are all examples of popular enhancement techniques. These and other enhancement operations can be achieved through the process of spatial domain filtering. The term spatial domain is arguably somewhat

spurious, but is used to distinguish this procedure from frequency domain procedures. Thus, spatial domain filtering simply indicates that the filtering process takes place directly on the actual pixels of the image itself. *(C,Solomon,2011)*

## 6.2. Image Restoration

Restoration is more connected to working with the appearance of the image, meaning this step helps in improving the appearance of the image. It is actually a step that can undo the defects in the image (degraded image). Degradation may also be in the form of, for example, noise or blurring, and restoration helps in restoring the image with better quality. *(W.Pratt, 2023)*

It may be viewed as an estimation process in which operations are performed on an observed or measured image field to estimate the ideal image field that would be observed if no image degradation were present in an imaging system. Mathematical models are described in the first section of this chapter for image degradation in general classes of imaging systems. These models are then utilized in the following sections as a basis for the development of image restoration techniques. *(W.Pratt, 2023)*

### 6.2.1. Image Restoration Models

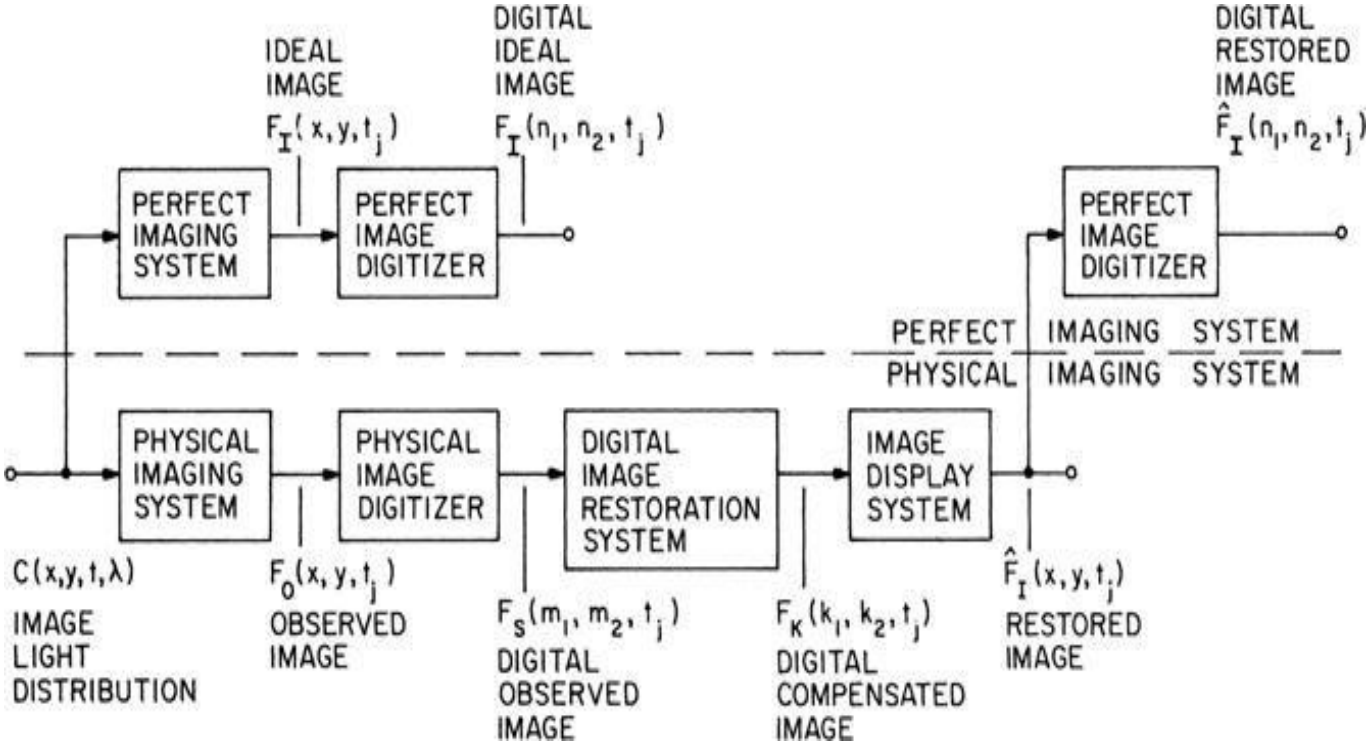
In order effectively to design a digital image restoration system, it is necessary quantitatively to characterize the image degradation effects of the physical imaging system, the image digitizer and the image display. Basically, the procedure is to model the image degradation effects and then perform operations to undo the model to obtain a restored image. It should be emphasized that accurate image modeling is often the key to effective image restoration. There are two basic approaches to the modeling of image degradation effects: a priori modeling and a posteriori modeling. In the former case, measurements are made on the physical imaging system, digitizer and display to determine their response for an arbitrary image field. In some instances, it will be possible to model the system response deterministically, while in other situations it will only be possible to determine the system response in a stochastic sense. The a posteriori modeling approach is to develop the model for the image degradations based on measurements of a particular image to be restored. Basically, these two approaches differ only in the manner in which information is gathered to describe

the character of the image degradation. *(W.Pratt, 2023)*

*Figure 1. 5 Digital image restoration model*

## 6.3. Color Image Processing

The color information (R, G, B) can be used to extract and understand the features from the image. The features that one normally thinks of are color, texture, shape, and structure, and this enables the user to understand the image better for meaningful processing. There are multiple color conversion models available for use that can be used in this step. *(A. Baskar, 2023)*




A color image may be represented by its red, green and blue source tristimulus values or any linear or nonlinear invertible function of the source tristimulus values. If the red, green and blue tristimulus values are to be quantized individually, the selection of the number and placement of quantization levels follows the same general considerations as for a monochrome image. The eye exhibits a nonlinear response to spectral lights as well as white light, and, therefore, it is subjectively preferable to compand the tristimulus values before quantization. It is known, however, that the eye is most sensitive to brightness changes in the blue region of the spectrum, moderately sensitive to brightness changes in the green spectral region and least sensitive to red changes. Thus, it is possible to assign quantization levels on this basis more efficiently than simply using an equal number for each tristimulus value. *(W.Pratt, 2023)*

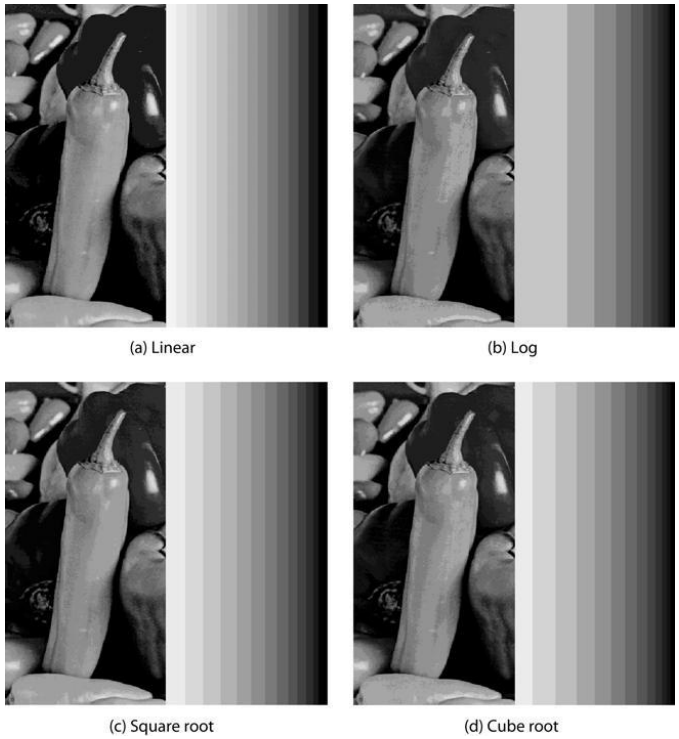



*Figure 1. 6 Comparison of lightness scale quantization of the peppers_ramp_luminance image for 4 bit quantization*

## 6.4. Image Segmentation

As the very term segmentation means "partitioning", it is easier for one to understand what image segmentation could be. It is the process or the step through which a digital image is partitioned into multiple segments. This segmenting allows the user to identify the objects and to extract more meaningful information from the image. One can think of segmentation as a divide-and-conquer approach for the taken image. Segmentation divides the image into multiple segments and extracts the best out of it. *(A. Baskar, 2023)*

### 6.4.1. Use of image properties and features in segmentation

In the most basic of segmentation techniques (intensity thresholding), the segmentation is used only on the absolute intensity of the individual pixels. However, more sophisticated properties and features of the image are usually required for successful segmentation. Before we begin our discussion of explicit techniques, it provides a useful (if somewhat simplified) perspective to recognize that there are three basic properties/qualities in images which we can exploit in our attempts to segment images. *(C,Solomon,2011)*

1) Color is, in certain cases, the simplest and most obvious way of discriminating between objects and background. Objects which are characterized by certain color properties (i.e., are confined to a certain region of a color space) may be separated

from the background. For example, segmenting an orange from a background comprising a blue tablecloth is a trivial task.

2) Texture is a somewhat loose concept in image processing. It does not have a single definition but, nonetheless, accords reasonably well with our everyday notions of a 'rough' or 'smooth' object. Thus, texture refers to the 'typical' spatial variation in intensity or color values in the image over a certain spatial scale. A number of texture metrics are based on calculation of the variance or other statistical moments of the intensity over a certain neighborhood/spatial scale in the image. We use it in a very general sense here.

3) Motion of an object in a sequence of image frames can be a powerful cue. When it takes place against a stationary background, simple frame-by-frame subtraction techniques are often sufficient to yield an accurate outline of the moving object.

### 6.4.2. Image representation and description

Immediately after segmentation, it is important to do further processing. It is not possible to just retain the same result of segmentation to arrive at meaningful results. Image representation is concerned with transforming the raw data arrived at after segmentation to some suitable form (feature vectors) for further processing. This can be achieved by two means *(A. Baskar, 2023)*:

- Boundary representation: Focus is on the external shape, for example, edges and corners.

- Regional representation: Focus is on the internal properties, such as texture and shape.

### 6.4.3. Object recognition

Recognition can be defined as the identification of someone or something from previous encounters or knowledge. This definition is perfect here. Recognition helps in recognizing what is what in the image. For instance, in an image that has a car and a motorcycle, the car will be recognized as a car and the motorcycle as a motorcycle. This is possible through the features present in the image. *(A. Baskar, 2023)*

# 7. Analyzing Image Quality Assessment

## 7.1. NPCR & UACI

The first measure is the number of pixels change rate (NPCR), which indicate the percentage of different pixels between two images.

The second one is the unified average changing intensity (UACI), which measures the average intensity of differences in pixels between two images. Let $I_o\ (i,j\ )$and $I_{ENC}\ (i,j\ )$be the pixels values of original and encrypted images, $I_o$ and $I_{ENC}$ , at the *i*th pixel row and *j*th pixel column, respectively. Equations (13) and (15) give the mathematical expressions of the NPCR and UACI measures:

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\% \quad (1.1)$$

$$with: D(i,j) = \begin{cases} 0 & if\ I_0(i,j) = I_{ENC}(i,j) \\ 1 & othrwise \end{cases} \quad (1.2)$$

$$UACI = \left[\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{I_0(i,j) - I_{ENC}(i,j)}{255}\right] \times \frac{100\%}{M \times N} \quad (1.3)$$

To approach the performances of an ideal image encryption algorithm, NPCR values must be as large as possible and UACI values must be around 33%.

## 7.2. Histogram

The histogram of an image refers to a graph of the pixel intensity values. The histogram is a graph showing the number of pixels in an image at different intensity values found in the image. In an 8-bit grayscale image, there are 256 different possible intensities, and so the histogram will display 256 numbers showing the distribution of pixels amongst those grayscale values. For a good encryption, the distribution of Gray scales in the encrypted image should be fairly uniform. (A.Bhute,2013)

## 7.3. Adjacent Pixel Correlation

The other statistical test consists of computing the correlation between adjacent pixels It is obvious that an arbitrarily chosen pixel in an image is generally strongly correlated with adjacent pixels, and it's in either horizontal, vertical or diagonal directions. However, a secure image encryption algorithm must produce an encrypted image having low correlation between adjacent pixels. This correlation test consists of randomly selecting N pairs of adjacent pixels

(vertical, horizontal, and diagonal) from the original and the encrypted images separately. Then, the correlation coefficient of each pair is calculated using (19) (A.Bhute,2013)

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1.4}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E((x))^2 \tag{1.5}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E((x))(y_i - E(y)) \tag{1.6}$$

$$y_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad with \quad D(x) \neq 0, D(y) \neq 0 \tag{1.7}$$

where $x_i$ and , $y_i$ are the grayscale values of two adjacent pixels, N is the number of pairs ($x_i$ , $y_i$ ), and $E(x)$and $E(y)$, are respectively, the mean values of$x_i$ and $y_i$.

## 7.4. Entropy

The concept of entropy analysis for image encryption algorithm was introduced by Edward for Gray-scale images of 256 levels, if each level of Gray is assumed to be equiprobable, then the entropy of this image will be theoretically equal to 8 Sh (or bits). Ideally, an algorithm for encryption of images should give an encrypted image having equiprobable Gray levels. (S.Sommraj,2015)

## 7.5. PSNR

Peak Signal-to-Noise Ratio is the ratio between the original image and the encrypted image. PSNR is calculated in decibels. The higher the PSNR, the closer the encrypted image is to the original. In general, a higher PSNR value should correlate to a higher quality image. For good encryption scheme the PSNR should be as low as possible. *(N.Sethi,2013)*

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE). \end{aligned} \tag{1.8}$$

# 8. Applications of Image Processing

## 8.1. Computer/Machine Vision

Computer/Machine Vision is one of the interesting applications of Digital Image Processing. Computer Vision sees, identifies things, and processes the whole environment by delivering insights from its observation. This is used in self-driving cars, drones, and other technologies where detection of images is necessary for automation.

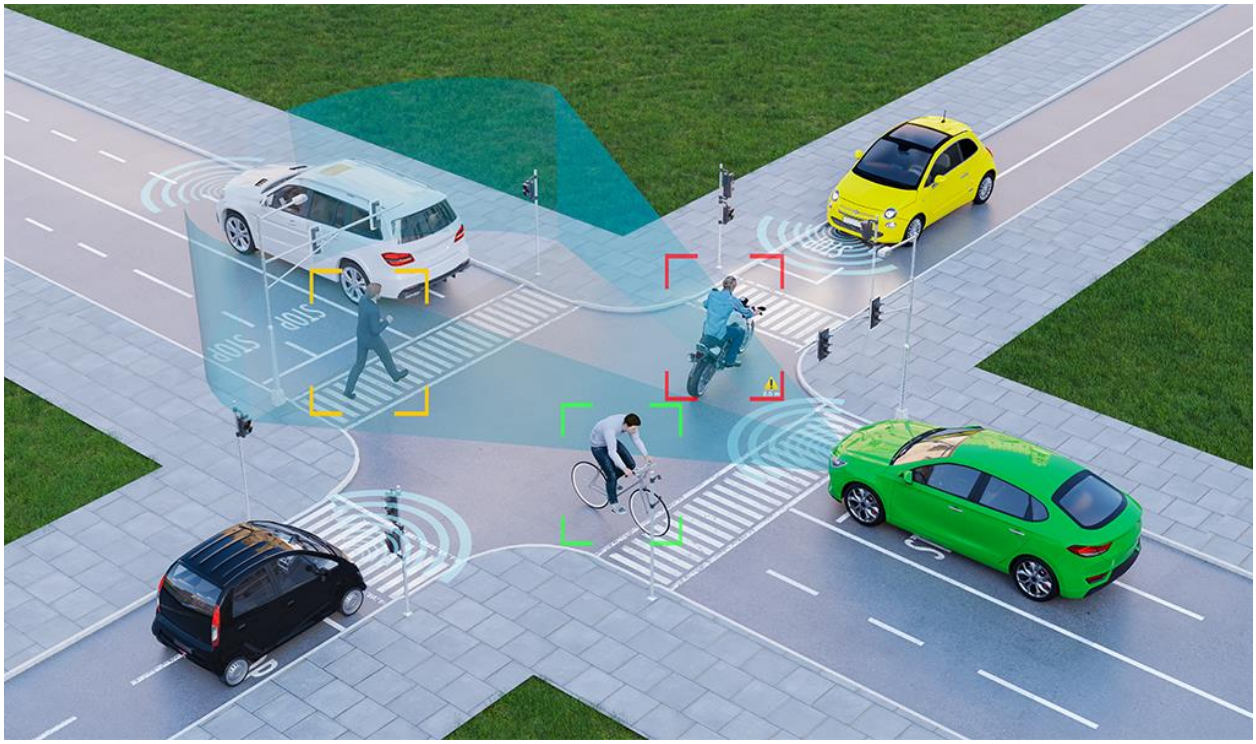



*Figure 1. 7 Example of computer-vision-applications*

## 8.2. Medical Image Retrieval

Image Processing in the field of medicine has played a vital role in the well-being of humankind. It has drastically increased the diagnostic process. It has paved the way for X-ray Imaging, PET Scans, UV Imaging, Medical CT, Cancer Cell Image Processing, and much more.

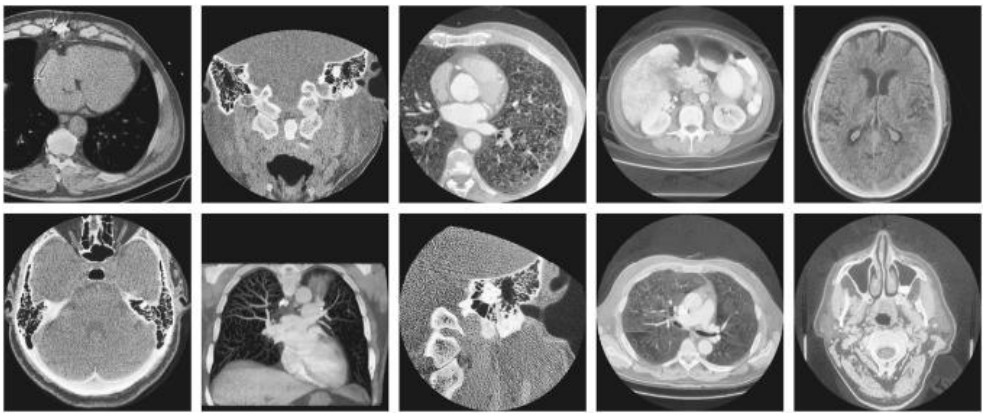


*Figure 1. 8 Example of medical image retrieval*

## 8.3. Image Reconstruction

Reconstruction of old and damaged photographs can be re-built with the help of Image Processing. The function takes up the existing datasets of the photograph and recreates the missing part of the objects in the photographs. Further, it can be used to edit the photographs by detecting edges, sharpening, retrieval, blurring, conversion, and recognizing objects in the image.

*Figure 1. 9 Example of image reconstruction*

## 8.4. Pattern Recognition

Image Processing is often combined with Artificial Intelligence. Pattern recognition often assists in handwriting analysis, computer-aided medical diagnosis, image recognition, etc. The function detects the pattern of objects and other aspects in an image.

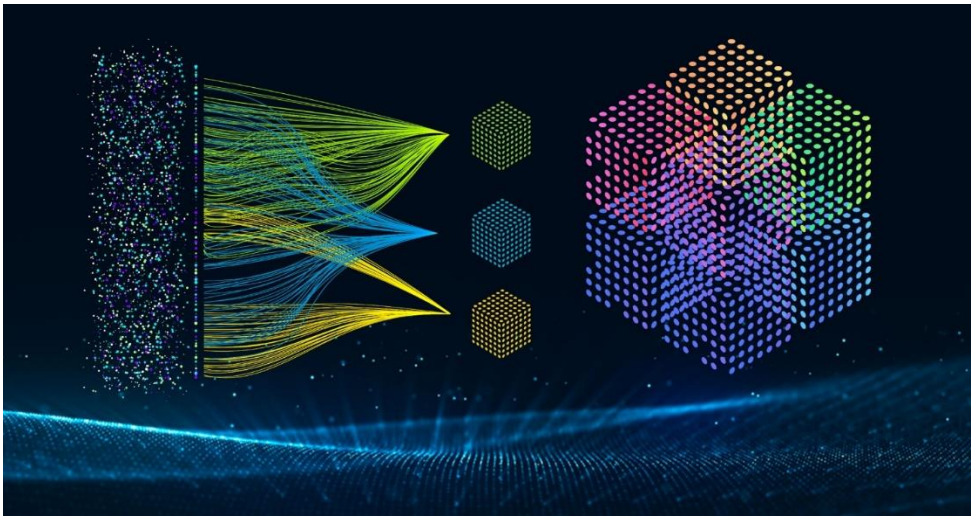


*Figure 1. 10 Example of pattern-recognition using AI*

## 8.5. Face Detection

Image Processing in face detection uses certain computed algorithms to recognize a face in an image. It uses certain features of the face mostly the eyes and edges including the distance between the eyes, the shape of face, etc. The function is used in entertainment industries to detect faces in motion and in smartphones to secure their personal details. RGB, Thermal, EEG, and wearable inertial sensors are used to obtain the details.

*Figure 1. 11 Facial Recognition Software*

## 9. Conclusion

Image processing involves manipulating digital images to extract information and enhance their visual quality. It includes techniques such as segmentation, object recognition, and image representation. Applications of image processing span various fields, including computer vision, medical imaging, image reconstruction, pattern recognition, and face detection. These technologies have found practical uses in automation, diagnostics, restoration, and security. Advancements in machine learning and deep learning have significantly improved object recognition capabilities. Overall, image processing plays a crucial role in extracting valuable insights and improving the interpretation of digital images across diverse industries.

# Chapter 2: Image encryption concepts

## 1. Introduction

This chapter provides an in-depth exploration of cryptography, an essential aspect of modern information security. It covers the historical development of cryptography, from ancient encryption methods to advanced techniques used today. The chapter discusses both classical and modern cryptography, including block ciphers, operational modes, and the concepts of private-key and public-key cryptography. It also examines the field of cryptanalysis, addressing potential vulnerabilities and threats to cryptographic systems. Throughout the chapter, the inseparable connection between information security and cryptography is highlighted, emphasizing its critical role in protecting communication and data. By the end of the chapter, we will have a comprehensive understanding of the importance and historical evolution of cryptography, as well as its significant contribution to securing sensitive information.

## 2. What is cryptography?

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. The Basics of Cryptography is the science of securing data and making it unreadable to anyone who does not possess the key or the secret information required to decrypt it. Cryptography plays a crucial role in ensuring confidentiality, integrity, authentication, and non-repudiation of sensitive information in various fields, including computer science, mathematics, and information security. *(O.Hawkins,2016)*

## 3. History of Cryptography

The art of cryptography is considered to be born along with the art of writing. As civilizations evolved, human beings got organized into tribes, groups, and kingdoms. This led to the emergence of ideas such as power, battles, supremacy, and politics. These ideas further fueled the natural need of people to communicate secretly with selective recipients which in turn ensured the continuous evolution of cryptography as well. The roots of cryptography are

found in Roman and Egyptian civilizations. (*Tutorialspoint,2019)*

## 3.1. Hieroglyph – The Oldest Cryptographic Technique

The first known evidence of cryptography can be traced to the use of 'hieroglyph'. Some 4000 years ago, the Egyptians used to communicate by messages written in hieroglyphs. This code was the secret known only to the scribes who used to transmit messages on behalf of the kings. One such hieroglyph is shown below.

*Figure 2. 1A message written in hieroglyph*

Later, the scholars moved on to using simple mono-alphabetic substitution ciphers from 500 to 600 BC. This involved replacing the alphabets of messages with other alphabets with some secret rule. This rule became a key to retrieving the message back from the garbled message.

The earlier Roman method of cryptography, popularly known as the Caesar Shift Cipher, relies on shifting the letters of a message by an agreed number (three was a common choice), the recipient of this message would then shift the letters back by the same number and obtain the original message. *(Tutorialspoint,2019)*

## 3.2. Evolution of Cryptography

It is during and after the European Renaissance, various Italian and Papal states led the

rapid proliferation of cryptographic techniques. Various analyses and attack techniques were researched in this era to break the secret codes.

- Improved coding techniques such as Vigenère Coding came into existence in the 15th century, which offered moving letters in the message with several variable places instead of moving them the same number of places.

- Only after the 19th century, cryptography evolved from the ad hoc approaches to encryption to the more sophisticated art and science of information security.

- In the early 20th century, the invention of mechanical and electromechanical machines, such as the Enigma rotor machine, provided more advanced and efficient means of coding information.

- During the period of World War II, both cryptography and cryptanalysis became excessively mathematical.

With the advances taking place in this field, government organizations, military units, and some corporate houses started adopting the applications of cryptography. They used cryptography to guard their secrets from others. Now, the arrival of computers and the Internet has brought effective cryptography within the reach of common people. *(Tutorialspoint,2019)*

## 3.3. Ciphers

Throughout history, people have felt the need to write messages that would be unreadable to anyone other than the intended recipient, specifically to anyone who might intercept it en route. Military commands, intelligence reports, instructions to agents, love letters, arrangements for meetings, plans for any kind of action or activity that could prompt counter-measures of any kind by any third party—all these and many more might be deemed by the sender to need encryption.

A cipher is a series of predetermined instructions that, when followed in the field of cryptography, allow a person to encrypt or decode a text message. The process of encrypting data involves transforming information from its clear form into a format that is either unreadable or inaccessible. The text's original version that's easy to understand is called plaintext. Whereas the encrypted version is known as ciphertext. Both of these texts have the same information. The sole difference between the two texts is that the information included

in the ciphertext is written in a format that can only be read or accessed by those who possess the appropriate decryption technique.

The vast majority of cipher algorithms have a certain piece of confidential information, which is called a cryptographic key. The encryption method is different depending on the key model, and the cipher can either be symmetric or asymmetric. In symmetric ciphers, only one key is required for encryption and decryption, but in asymmetric ciphers, distinct keys are used for each stage of the process *(J.Philippe,2018)*

### 3.3.1. How Ciphers work

Based on simplistic ciphers like the Caesar and Vigenère ciphers, we can try to abstract out the workings of a cipher, first by identifying its two main components: a permutation and a mode of operation. A permutation is a function that transforms an item (in cryptography, a letter, or a group of bits) such that each item has a unique inverse (for example, the Caesar cipher's three-letter shift). A mode of operation is an algorithm that uses a permutation to process messages of arbitrary size. The mode of the Caesar cipher is trivial: it just repeats the same permutation for each letter, but as you've seen, the Vigenère cipher has a more complex mode, where letters at different positions undergo different permutations. *(J.Philippe,2018)*

### 3.3.2. The Permutation

Most classical ciphers work by replacing each letter with another letter— in other words, by performing a substitution. In the Caesar and Vigenère ciphers, the substitution is a shift in the alphabet, though the alphabet or set of symbols can vary: instead of the English alphabet, it could be the Arabic alphabet; instead of letters, it could be words, numbers, or ideo- grams, for example. The representation or encoding of information is a separate matter that is mostly irrelevant to security. (We're just considering Latin letters because that's what classical ciphers use.) *(J.Philippe,2018)*

A cipher's substitution can't be just any substitution. It should be a permutation, which is a rearrangement of the letters A to Z, such that each letter has a unique inverse. For example, a substitution that transforms the letters A, B, C, and D, respectively to C, A, D, and B is a permutation, because each letter maps onto another single letter. But a substitution that transforms A, B, C, D to D, A, A, C is not a permutation, because both B and C map onto A. With a permutation, each letter has exactly one inverse.

Still, not every permutation is secure. To be secure, a cipher's permutation should satisfy three criteria:

- The permutation should be determined by the key, to keep the permutation secret as long as the key is secret. In the Vigenère cipher, if you don't know the key, you don't know which of the 26 permutations was used; hence, you can't easily decrypt.

- Different keys should result in different permutations. Otherwise, it becomes easier to decrypt without the key: if different keys result in identical permutations, that means there are fewer distinct keys than distinct permutations, and therefore fewer possibilities to try when decrypting without the key.

- The permutation should look random, loosely speaking. There should be no pattern in the ciphertext after performing a permutation because patterns make a permutation predictable for an attacker, and therefore less secure.

We'll call a permutation that satisfies these criteria a secure permutation. But a secure permutation is necessary but not sufficient on its own for building a secure cipher. A cipher will also need a mode of operation to support messages of any length. *(J.Philippe,2018)*

## 4. Classical cryptography

### 4.1. Caesar cipher

We begin with historical ciphers in antiquity. Roman sources say that the emperor and general Gaius Julius Caesar proceeded for his secret communication in such a way that he replaced each letter of the alphabet by the one three places further. Figure 1.2 shows an example. *(O.Manz,2022)*

Therefore, this cipher procedure is also called Caesar cipher, the underlying alphabet being A, B,…, Z. We first make two observations:

- Of course, instead of three digits, you could have chosen another number i from 0 to 25 and replaced each letter with the one i digits further.

- We have also already learned that instead of letters it is better to use the remainders 0, 1,…, 25 modulo 26. So, based on these observations, the ciphertext for the characters z of our alphabet is $z \rightarrow z + i \pmod{26}$. This is called a shift cipher.

But how did the governors in the Roman provinces decode Caesar's orders, i.e. decipher them? Of course, they had to know the shift value i = 3, and again they moved all the letters back three places in the alphabet, in which case, for example, "A" becomes "X". Using our algorithmic notation for shift ciphers, this means $z \to z - i = z + (26 - i)$ (mod 26). Thus, for the Caesar cipher with i = 3 and the letter A, i.e., $z = 0$, we get $0 \to -3 = 26 - 3 = 23$ (mod 26), which is the letter X. In this way, one can unambiguously decipher each character in the encoded letter sequence. *(O.Manz,2022)*

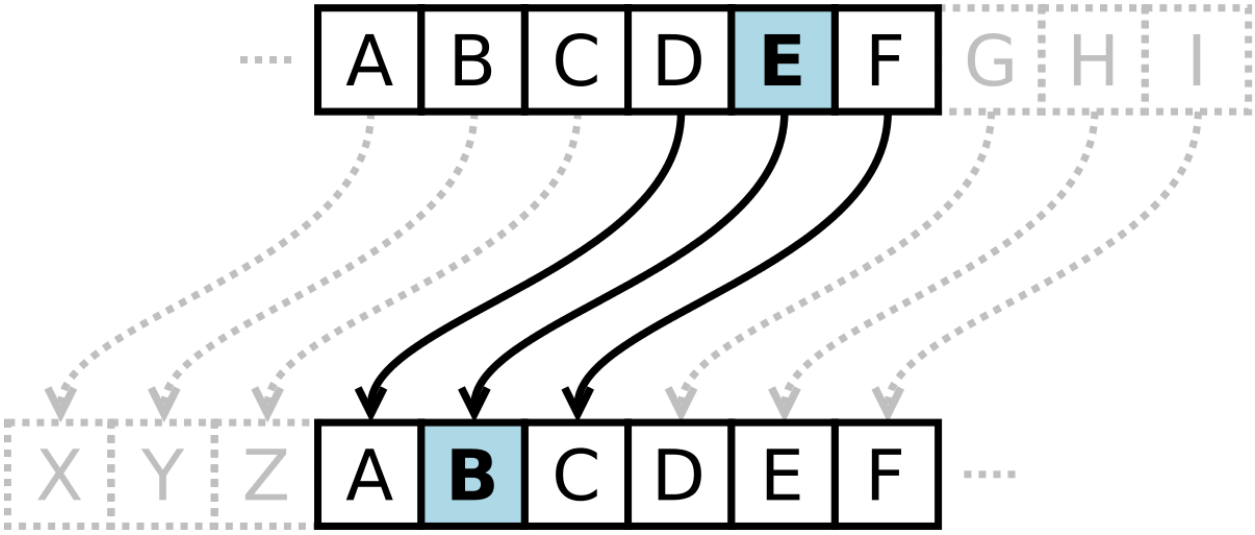



*Figure 2. 2 Caesar Cipher*

## 4.2. Substitution cipher

Instead of shifting everything over by a fixed amount, we can have every letter stand for a different letter, at random perhaps. So, for example, we could consider the following set of substitutions:

abcde fghij klmno pqrst uvwxy z

XHZRW FGPTY JOCAE ULNKQ IVSMD B

The first line above denotes a plaintext character, and the second line denotes the corresponding ciphertext character. This means that, for example, every time we want to write an a, we actually write an X, and every time we actually want to write a b, we write an H, and so on.

*(S.Rubinstein,2010)*

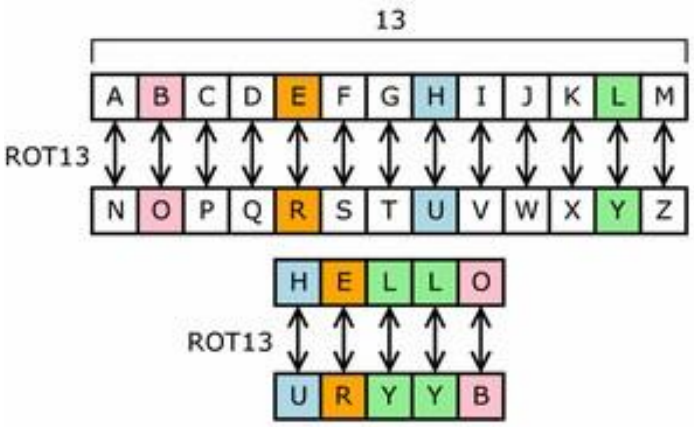



*Figure 2. 3 Simple subtitution*

## 4.3. The One-time Pad

This is the only unbreakable cryptosystem in existence. It can be considered as a variation of the Vigenère cipher, where the keyword is as long as the plaintext. If the keyword is random, then there is no possible correspondence between plaintext and ciphertext, making cryptanalysis impossible. For example, using the same plaintext as before, and keyword *(McAndrew, 2012)*

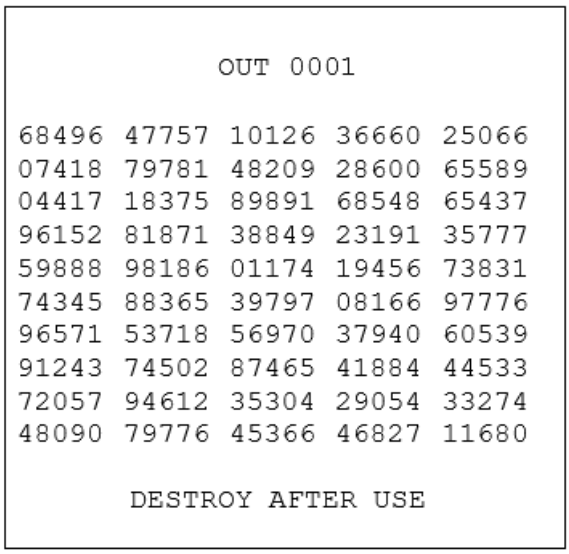
ETJWBOGLVWGESAKYMOPFJKWGG,

the resulting ciphertext is

ABCDEFGHIJKLMNOPQRSTUVWXY.

There are difficulties inherent in using this particular cryptosystem. Both persons must agree on a suitable random string, long enough to be used for all possible messages between them. If they start using a short string more than once, then effectively they are using a Vigenère cipher, and cryptanalysis is then possible. Then there is the problem of getting this string from one person to the other, without it's being intercepted by a third person. *(McAndrew, 2012)*

It is important that the string be random. If the persons are using as a keyword an English text, for example, a book known to both of them, then there is the possibility of a successful cryptanalysis using frequency analysis. *(McAndrew, 2012)*



OUT 0001

68496 47757 10126 36660 25066
07418 79781 48209 28600 65589
04417 18375 89891 68548 65437
96152 81871 38849 23191 35777
59888 98186 01174 19456 73831
74345 88365 39797 08166 97776
96571 53718 56970 37940 60539
91243 74502 87465 41884 44533
72057 94612 35304 29054 33274
48090 79776 45366 46827 11680

DESTROY AFTER USE

*Figure 2. 4 Example of one-time pad sheet*

## 4.4. The Vigenère cipher

Historical note: This cipher is attributed to Blaise Vigenère (1532–1596) who was a diplomat to the Vatican, although it was in fact developed not by him but by Giovan Battista Bellaso in 1553. By a curious stroke of historical fate, this cipher is now called the "Vigenère Cipher," whereas in fact the cipher that Vigenère *did* create, known as an "autokey cipher" is much stronger. It was considered a great improvement over mono-alphabetic ciphers, and was thought at the time to be unbreakable. It was finally broken in 1832. *(McAndrew, 2012)*

In this cipher, encryption depends not on a single value giving the shift, but on a keyword.

The encryption and decryption algorithms can be described as

$$ci = pi + ki \quad \text{(mod 26)}$$

$$pi = ci - ki \quad \text{(mod 26)}$$

where $p_i$, $k_i$ and $c_i$ are the corresponding plaintext, keyword and ciphertext letters. For example, using the same plaintext as before, and the keyword CODE, the encryption is:

plaintext: W I T H D R A W O N E H U N D R E D D O L L A R S

keyword: C O D E C O D E C O D E C O D E C O D E C O D E C

ciphertext: Y W W L F F D A Q B H L W B G V G R G S N Z D V U

## 4.5. Enigma Machine

The electro-mechanical Enigma, used by Germany and Japan, was invented by A. Scherbius in 1923. It consists of three rotors and a reflector. See Figure 2.4. When punching in a letter, an electronic current will enter the first rotor at the place corresponding with that letter, but will leave it somewhere else depending on the internal wiring of that rotor. The second and third rotors do the same, but have a different wiring. The reflector returns the current at a different place and the current will go through rotors 1, 2 and 3 again but in reverse order. The current will light up a letter, which gives the encryption of the original letter.

Simultaneously, the first rotor will turn position. After 26 rotations of the first rotor the second will turn one position. When the second rotor has made a full cycle, the third rotor will rotate over one position.

The key of the Enigma consists of

- the choice and order of the rotors,
- their initial position and
- a fixed initial permutation of the alphabet.

*Figure 2. 5 Enigma Machine*

## 4.6. Rotor machines

The example just given suggests that multiple stages of encryption can produce an algorithm that is significantly more difficult to cryptanalysis. This is as true of substitution ciphers as it is of transposition ciphers. Before the introduction of DES, the most important application of the principle of multiple stages of encryption was a class of systems known as rotor machines.

The basic principle of the rotor machine is illustrated in Figure 1.6. The machine consists of a set of independently rotating cylinders through which electrical pulses can flow. Each cylinder has 26 input pins and 26 output pins, with internal wiring that connects each input pin to a unique output pin. For simplicity, only three of the internal connections in each cylinder are shown.

If we associate each input and output pin with a letter of the alphabet, then a single cylinder defines a monoalphabetic substitution. For example, in Figure 1.6, if an operator depresses the key for the letter A, an electric signal is applied to the first pin of the first cylinder and flows through the internal connection to the twenty-fifth output pin.

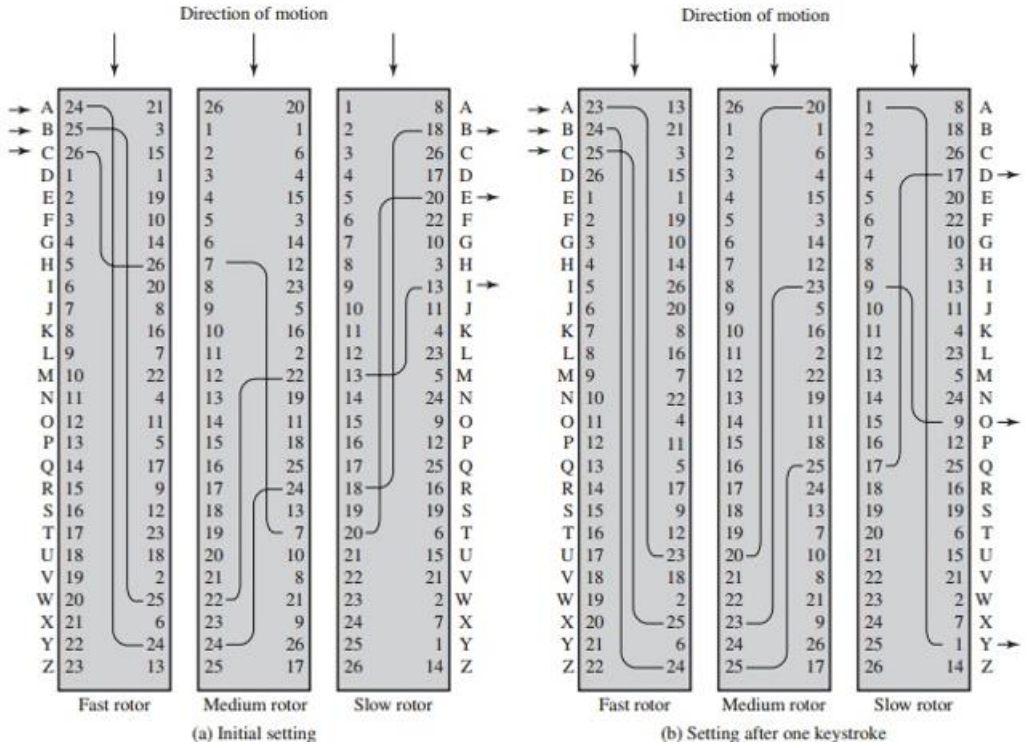



*Figure 2. 6 Three-Roto Machine with wiring represented by numbered contacts*

***https://www.brainkart.com/article/Rotor-Machines_8389/***

# 5. Modern cryptography

## 5.1. Public-Key Cryptography

Until 1976 it was assumed, at least in public, that sender and receiver always needed a shared secret key to communicate confidentially with each other. Then the seminal article "New Directions in Cryptography" by Whitfield Diffie and Martin Hellman was published. In this article, the first public key protocol – the Diffie-Hellman key exchange was introduced. In addition, it described utterly new concepts, such as public-key encryption and digital signatures. Ralph Merkle came up with a similar idea at almost the same time. The terms "public key algorithms" and "asymmetric algorithms" describe the same concept: A public key is publicly accessible to all parties in a communication system. In contrast, the private key is only known to a single entity. In an attempt to refute the speculations of Diffie and Hellman, the probably most famous public key algorithm was born in 1978: initially named "The MIT algorithm" by Ron Rivest, Adi Shamir, and Leonard Adleman, it is known today as the RSA algorithm. Another important public-key encryption scheme is the ElGamal algorithm. (*J.Katz,2007)*

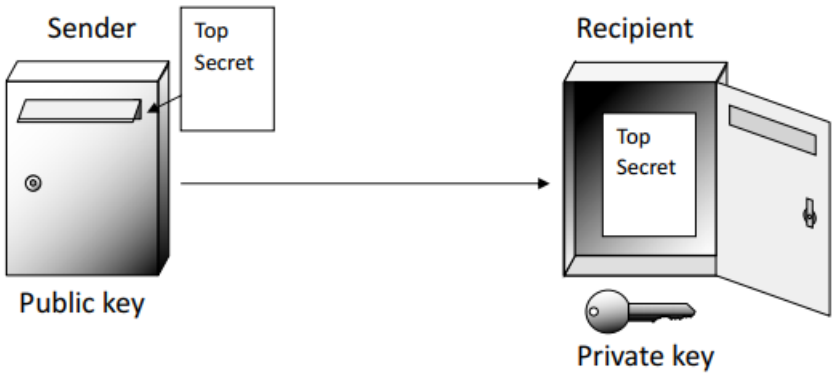



*Figure 2. 7 Exapmle of how Asymmetric encryption works*

**Mental Model** :A mental model for asymmetric encryption is depicted in Figure 2.14: Everyone may encrypt a message by dropping it into a publicly accessible mailbox, i.e., by using the public key. Only the owner of the mailbox, who has the matching private key, can open it and read the message. *(J.Schwenk,2022)*

### 5.1.1. Diffie–Hellman

The invention of public-key cryptography is often attributed to Whitfield Diffie and Martin

Hellman: in a famous paper. which was published in the IEEE Transactions on Information Theory journal in 1976, they gave "new directions in cryptography," describing how we can use one-way functions and notions of trapdoor permutations in cryptography. A public-key cryptosystem is nothing but a kind of one-way permutation (anybody can encrypt, but cannot decrypt) with a hidden trapdoor that enables the decryption to the legitimate party. *(W.Stallings, 2010)*

#### 5.1.1.1. Key Exchange Protocols

It's a simple protocol that makes use of the Diffie-Hellman calculation. Suppose that user A wishes to set up a connection with user B and use a secret key to encrypt messages on that connection. User A can generate a one-time private key $X_A$, calculate $Y_A$, and send that to user B. User B responds by generating a private value $X_B$, calculating $Y_B$, and sending $Y_B$ to user A. Both users can now calculate the key. The necessary public values $q$ and a would need to be known ahead of time. Alternatively, user, A could pick values for $q$ and a and include those in the first message. *(W.Stallings, 2010)*

As an example of another use of the Diffie-Hellman algorithm, suppose that a group of users (e.g., all users on a LAN) each generate a long-lasting private value $X_i$ (for user $i$), and calculate a public value, $Y_i$. These public values, together with global public values for $q$ and □, are stored in some central directory. At any time, user $j$ can access the user $i$'s public value, calculate a secret key, and use that to send an encrypted message to user A. If the central directory is trusted, then this form of communication provides both confidentiality and a degree of authentication. Because only $i$ and $j$ can determine the key, no other user can read the message (confidentiality). Recipient, $i$ know that only user $j$ could have created a message using this key (authentication). However, the technique does not protect against replay attacks. *(W.Stallings, 2010)*

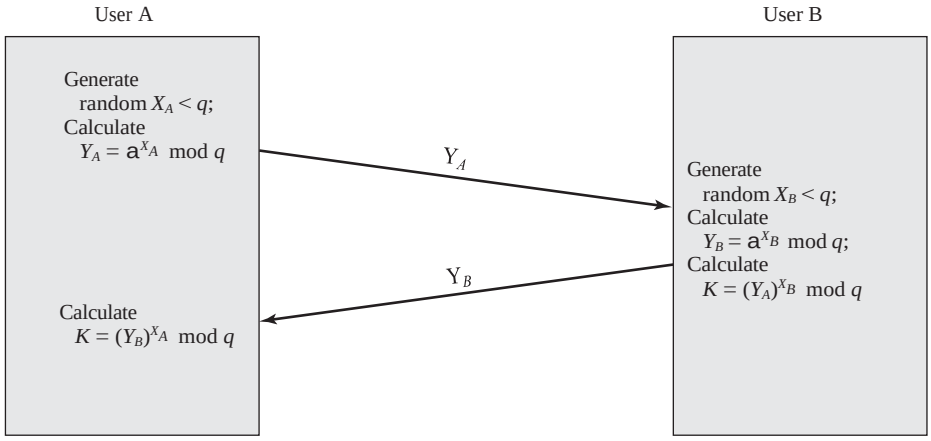



*Figure 2. 8 Diffie-Hellman Key Exchange*

### 5.1.2. RSA

The RSA system, named after its inventors Ron Rivest, Adi Shamir, and Len Adleman, was the first public-key cryptosystem and is still the most important. Its security is closely related to the difficulty of finding the factorization of a composite positive integer that is the product of two large primes. We first explain how the RSA system works and then we discuss its security and efficiency. *(J.Buchmann,2000)*

**Key Generation:** We explain how Bob generates his private and public RSA keys.

Bob generates randomly and independently two large (odd) prime numbers p and q and computes the product n = pq. *(J.Buchmann,2000)*

Bob also chooses an integer e with

1 < e < <p(n) = (p — l)(q — 1) and ged (e, (p — l)(q — 1)) = 1.

Note that e is always odd since *p*-1 is even. Bob computes an integer *d* with

1 < *d* < (p - l)(q — 1) and *de* = 1 mod (p — 1)(<? — 1). (7.1) Since gcd (e, *(p* - l)(q — 1)) = 1, such a number *d* exists. It can be computed by the extended euclidean algorithm.

Bob's public key is the pair (n, e). His private key is *d.* The number n is called the *RSA modulus, e* is called the *encryption exponent,* and *d* is called the *decryption exponent.* Note that the secret key *d* can be computed from the encryption exponent *e* if the prime factors *p* and *q* of *n* are known. Therefore, if the attacker, Oscar, is able to find the prime factorization of *n,*

then he can easily find Bob's secret key *d. (J.Buchmann,2000)*

**Encryption:** We first explain how to encrypt numbers with the RSA system. Then we show how RSA can be used as a block cipher.

In the first variant, the plaintext space consists of all integers *m* with

$0 < m < n$.

A plaintext *m* is encrypted by computing

$c = m^e \bmod n$.

The ciphertext is c. If Alice knows the public key (n, e), she can encrypt. lb make encryption efficient, Alice uses fast exponentiation

**Decryption:** The decryption of RSA is based on the following theorem.

*Let* (n, e) *be a public RSA key and d the corresponding private RSA key. Then*

$$(m^e)^d \bmod n = m$$

*for any integer m with* $0 < m < n$.

*(J.Buchmann,2000)*

#### 5.1.2.1. The Security of RSA

Four possible approaches to attacking the RSA algorithm are

- Brute force: This involves trying all possible private keys.
- Mathematical attacks: There are several approaches, all equivalent in effort to factoring the product of two primes.
- Timing attacks: These depend on the running time of the decryption algorithm.
- Chosen ciphertext attacks: This type of attack exploits properties of the RSA algorithm.

The defense against the brute-force approach is the same for RSA as for other cryptosystems, namely, to use a large key space. Thus, the larger the number of bits in d, the better. However, because the calculations involved, both in key generation and in encryption/decryption, are complex, the larger the size of the key, the slower the system will run. *(W.Stallings, 2010)*

### 5.1.3. ELGAMAL encryption

We introduce the ElGamal public key encryption scheme, which is closely related to the Diffie-Hellman key exchange protocol *(S.Padhye,2018).*

**Key Generation:** Choose a prime number *p*, and construct the corresponding cyclic group Z. Choose a primitive root *g* (mod *p*) and an integer $a$ $_u$ 2, 3, 4 . . . , *p* 2.
Compute $A = g^a$ (mod *p*). The public key of the user (Bob, say) is (*p*, *g*, *A*) and the private key is *a*.

**Encryption:** Let the plaintext be *M* Z*?* To encrypt the message *M*, the sender first chooses a random number $r$ 2, 3, . . . , $p$ 2 and computes $C1 = gr\ (mod\ p)$ and $C2 = MAr\ (mod\ p)$. The complete ciphertext is the pair $(C1, C2)$.
**Decryption:** To decrypt the ciphertext ($C_1$,$C_2$), first computes $k = C^a = g^{ar}$
(mod *p*) and then obtains the plaintext *m* as $m = k^{-1}C_2$ (mod *p*).
1

In 1984, T. Elgamal announced a public-key scheme based on discrete logarithms, closely related to the Diffie-Hellman technique [ELGA84, ELGA85]. The ElGamal[2] cryptosystem is used in some form in several standards including the digital signature standard (DSS), and the S/MIME e-mail standard As with Diffie-Hellman, the global elements of ElGamal are a prime number q and a, which is a primitive root of q. User A generates a private/public key pair as follows:

1. Generate a random integer $X_A$, such that $1 < X_A < q - 1$.
2. Compute $Y^A = a^X A \bmod q$ .
3. A's private key is $X_A$; A's pubic key is {*q*, a, $Y_A$}.

Any user B that has access to A's public key can encrypt a message as follows:

Represent the message as an integer *M* in the range $0 \le M \le q - 1$. Longer messages are sent as a sequence of blocks, with each block being an integer less than *q*.

1. Choose a random integer *k* such that $1 \le k \le q - 1$.
2. Compute a one-time key $K = (Y_A)^k \bmod q$.
3. Encrypt *M* as the pair of integers ($C_1$, $C_2$) where

$C_1 = a^k \bmod q$; $C_2 = KM \bmod q$
User A recovers the plaintext as follows:

1. Recover the key by computing $K = (C_1)^X A \bmod q$ .
2. Compute $M = (C_2K^{-1}) \bmod q$.*(J.Katz,2007)*

## 5.2. Comparison to Private-Key Encryption

Perhaps the most obvious difference between private- and public-key encryption is that the former assumes complete secrecy of all cryptographic keys, whereas the latter requires secrecy for only the private key SK. Although this may seem like a minor distinction, the ramifications are huge: in the private-key setting the communicating parties must somehow be able to share the secret key without allowing any third party to learn it, whereas in the public-key setting the public key can be sent from one party to the other over a public channel without compromising security. For parties shouting across a room or, more realistically, communicating over a public network like a phone line or the Internet, public-key encryption is the only option. *.(J.Katz,2014)*

Another important distinction is that private-key encryption schemes use the same key for both encryption and decryption, whereas public-key encryption schemes use different keys for each operation. That is, public-key encryption is inherently asymmetric. This asymmetry in the public-key setting means that the roles of sender and receiver are not interchangeable as they are in the private-key setting: a single key pair allows communication in one direction only. (Bidirectional communication can be achieved in several ways; the point is that a single invocation of a public-key encryption scheme forces a distinction between one user who acts as a receiver and other users who act as senders.) In addition, a single instance of a public-key encryption scheme enables multiple senders to communicate privately with a single receiver, in contrast to the private-key case where a secret key shared between two parties enables private communication only between those two parties. *.(J.Katz,2014)*

Summarizing and elaborating on the preceding discussion, we see that public-key encryption has the following advantages relative to private-key encryption*(R.Mollin,2003)*:

- Public-key encryption addresses (to some extent) the key-distribution problem, since communicating parties do not need to secretly share a key in advance of their communication. Two parties can communicate secretly even if all communication between them is monitored.

- When a single receiver is communicating with N senders (e.g., an online merchant processing credit card orders from multiple purchasers), it is much more convenient for the receiver to store a single private key *sk* rather than to share, store, and manage N different secret keys (i.e., one for each sender). When using public-key

encryption the number and identities of potential senders need not be known at the time of key generation. This allows enormous flexibility in "open systems."

- The fact that public-key encryption schemes allow anyone to act as a sender can be a drawback when a receiver only wants to receive messages from one specific individual. In that case, an authenticated (private-key) encryption scheme would be a better choice than public-key encryption.

The main disadvantage of public-key encryption is that it is roughly 2 to 3 orders of magnitude slower than private-key encryption.1 It can be a challenge to implement public-key encryption in severely resource-constrained devices like smartcards or radio-frequency identification (RFID) tags. Even when a desktop computer is performing cryptographic operations, carrying out thousands of such operations per second (as in the case of an online merchant processing credit card transactions) may be prohibitive. Thus, when private-key encryption is an option (i.e. if two parties can securely share a key in advance), then it typically should be used. *(R.Mollin,2003)*

## 5.3. Private-key Cryptography

Private-key or Symmetric key cryptography is also known as single-key, secret-key, and private key or one-key encryption. In this technique sender and recoverer share the same key for the encryption and decryption process. This technique was one of the simplest and earliest. It used the concept of a common key. The key was supposed to be some secret info shared by the sender and the receiver. The symmetric key algorithm is divided into two parts: the first one is BLOCK CIPHER which is used for blocks of data. In this technique, data is divided into blocks and then these blocks are used for encryption and decryption. Example of block cipher is AES, and triple DES which is popular techniques of symmetric algorithms. And the second one is STREAM CIPHER which operates on a single bit at a time. Transmitting the secret key on the insecure network is also a curse of destroys the secrecy. There are many advantages of symmetric key cryptography like Symmetric key encryption is much faster. Single-key encryption does not require a lot of computer resources when compared to public key encryption. A different secret key is used for communication with every different party. If a key is compromised, only the messages between a particular pair of sender and receiver are affected. Communications with other people are still secure. *(J. Katz,2014)*

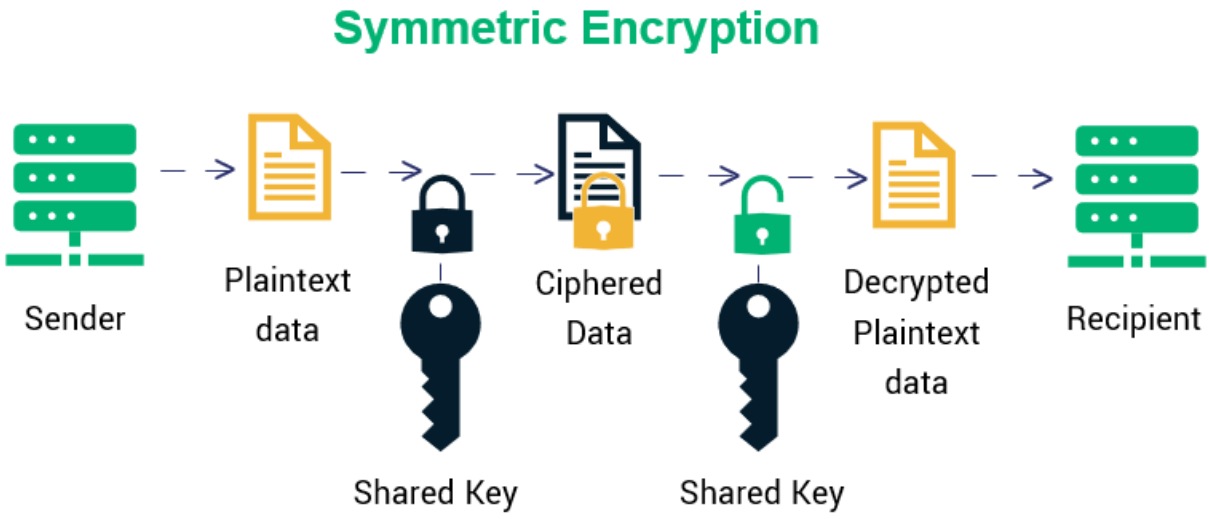



*Figure 2. 9 An example of symmetric encryption in action*

### 5.3.1. What is Symmetric Encryption Used For?

While symmetric encryption is an older method of encryption, it is faster and more efficient than asymmetric encryption, which takes a toll on networks due to performance issues with data size and heavy CPU use. Due to the better performance and faster speed of symmetric encryption (compared to asymmetric), symmetric cryptography is typically used for bulk encryption / encrypting large amounts of data, e.g. for database encryption. In the case of a database, the secret key might only be available to the database itself to encrypt or decrypt. Industry-standard symmetric encryption is also less vulnerable to advances in quantum computing compared to the current standards for asymmetric algorithms (at the time of writing). *(J. Katz,2014)*

### 5.3.2. Stream Ciphers

In the cryptographic literature, an encryption scheme is often called a stream cipher. This is due to the fact that encryption is carried out by first generating a stream of pseudorandom bits, and then XORing this stream with the plaintext. Unfortunately, there is a bit of confusion as to whether the term "stream cipher" refers to the algorithm that generates the stream (i.e. , the pseudorandom generator G) or to the entire encryption scheme. This is a crucial issue because the way a pseudorandom generator is used determines whether or not a given encryption scheme is secure. In our opinion, it is best to use the term stream cipher to refer to the algorithm that generates the pseudorandom stream, and thus a "secure" stream cipher should satisfy the definition of a variable output length pseudorandom generator.8 Using this terminology, a stream cipher is not an encryption scheme per se, but rather a tool for

constructing encryption schemes. *(S.Rubinstein,2010)*

Stream ciphers in practice. There are a number of practical constructions of stream ciphers available, and these are typically extraordinarily fast. A popular example is the stream cipher RC4 which is widely considered to be secure when used appropriately. The security of practical stream ciphers is not yet very well understood, particularly in comparison to block ciphers. This is borne out by the fact that there is no standardized, popular stream Cipher that has been used for many years and whose security has not come into question. For example, "plain" RC4 (that was considered secure at one point and is still widely deployed) is now u known to have soine significant weaknesses. For one, the first few bytes of the output stream generated by RC4 have been shown to be biased. Although this may seem benign, .it was also shown that this weakness can be used to feasibly break the WEP encryption protocol used in 802.11 wireless networks. *(S.Rubinstein,2010)*

(WEP is a standardized protocol for protecting wireless communications. The WEP standard has since been updated to fix the problem.) If RC4 is to be used, the first 1024 bits or so of the output stream should be discarded. *(S.Rubinstein,2010)*

Linear feedback shift registers (LFSRs) have, historically, also been popular as stream ciphers. However, they have been shown to be horribly insecure (to the extent that the key can be completely recovered given sufficiently-many bytes of the output) and so should never be used today. *(S.Rubinstein,2010)*

#### 5.3.2.1. FISH method

FISH is used to refer to several different encryption algorithms.

The FISH stream cipher is a fast software-based stream cipher using Lagged Fibonacci generators, plus a concept from the shrinking generator cipher. The name is an acronym for" Fibonacci Shrinking". It was published by Siemens in 1993. FISH is quite fast in software and has a huge key length. However, in the same paper where he proposed Pike, Ross Anderson showed that FISH can be broken with just a few thousand bits of known plaintext. *(M.Prokop, 2004)*

#### 5.3.2.2. ISAAC

ISAAC was designed in 1996 by Bob Jenkins. The name is an acronym for Indirection, Shift, Accumulate, Add and Count.

ISAAC is one of a family of PRNGs inspired by RC4. It is extremely fast on 32-bit word machines. Averaged out, it requires 18.75 machine cycles to generate each 32-bit value. Cycles are guaranteed to be at least 240 values long, and they are 28295 values long on average. The results are uniformly distributed, unbiased, and unpredictable unless you know the seed. *(M.Prokop, 2004)*

#### 5.3.2.3. Scream

Scream was designed in 1992 by Shai Halevi and Don Coppersmith and was presented together with Charanjit Jutla in february 2002. Scream uses a key length of 128 bit. It has been designed to make the SEAL algorithm more secure. Its software performance is 5 cycles/byte. *(M.Prokop, 2004)*

#### 5.3.2.4. RC4

RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security. It is a variable key- size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation. Analysis shows that the period of the cipher is overwhelmingly likely to be greater than 10100 |ROBS95|. Eight to sixteen machine operations are required per output byte, and the cipher can be expected to run very quickly in software. RC4 was kept as a trade secret by RSA Security. In September 1994, the RC4 algorithm was anonymously posted on the Internet on the Cypherpunks anonymous remailers list. *(W.Stallings, 2005)*

The RC4 algorithm is remarkably simply and quite easy to explain. A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S|0], S[l], ..., S[255]. At all times, S contains a permutation of all 8-bit numbers from 0 through 255. For encryption and decryption, a byte k (see Figure 1) is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of k is generated, the entries in S are once again permuted. *(W.Stallings, 2005)*

### 5.3.3. Block Ciphers

A *block cipher* is an encryption function for fixed-size blocks of data. The current generation of block ciphers has a block size of 128 bits (16 bytes). These block ciphers encrypt a 128-bit plaintext and generate a 128-bit ciphertext as the result. The block cipher is reversible; there is a decryption function that takes the 128-bit ciphertext and decrypts it to the original 128-bit plaintext. The plaintext and ciphertext are always the same size, and we call this the block size of the block cipher. *(N.Gerguson,2010)*

To encrypt with a block cipher, we need a secret key. Without a secret key, there is no way to hide the message. Like the plaintext and ciphertext, the key is also a string of bits. Common key sizes are 128 and 256 bits. We often write $E(K, p)$ or $E_K(p)$ for the encryption of plaintext $p$ with key $K$ and $D(K, c)$ or $D_K(c)$ for the decryption of ciphertext $c$ with key $K$. *(N.Gerguson,2010)*

When using block ciphers, as with any encryption task, we always follow Kerckhoffs' principle and assume that the algorithms for encryption and decryption are publicly known. Some people have a hard time accepting this, and they want to keep the algorithms secret. *(N.Gerguson,2010)*

It is sometimes useful to look at a block cipher as a very big key-dependent table. For any fixed key, you could compute a lookup table that maps the plaintext to the ciphertext. This table would be huge. For a block cipher with 32-bit block size, the table would be 16 GB; for a 64-bit block size, it would be 150 million TB; and for a 128-bit block size it would be $5\ 10^{39}$ bytes, a number so large there is not even a proper name for it. Of course, it is not practical to build such a table in reality, but this is a useful conceptual model. We also know that the block cipher is reversible. In other words, no two entries of the table are the same, or else the decryption function could not possibly decrypt the ciphertext to a unique plaintext. This big table will therefore contain every possible ciphertext value exactly once. This is what mathematicians call a permutation: the table is merely a list of all the possible elements where the order has been rearranged. A block cipher with a block size of $k$ bits specifies a permutation on $k$-bit values for each of the key values. *(N.Gerguson,2010)*

As a point of clarification, since it is often confused, a block cipher does not permute the bits of the input plaintext. Rather, it takes all the $2^k$ possible $k$-bit inputs and maps each to a unique $k$-bit output. As a toy example, if $k$ 8, an input 00000001 might encrypt to 0100000 under a given key but it might also encrypt to 11011110 under a different key, depending on the design of the block cipher. *(N.Gerguson,2010)*

Block ciphers are used for many purposes, most notably to encrypt information. For security purposes, however, one rarely uses a block cipher directly. Instead, one should use a *block cipher mode*, which we will discuss next. *(N.Gerguson,2010)*

### 5.3.4. Block Cipher Modes of Operation

Block ciphers can be deployed in different so-called "Operation Modes". Depending on

external requirements or threats to avoid a suitable mode should be chosen. The most common modes are:

#### 5.3.4.1. ECB mode

Electronic codebook mode: the most obvious mode. Here each block of data is encrypted/decrypted completely independent of all data before or after this block. This has the advantage that encryption of multiple blocks in parallel is possible, transmission errors are confined to the current block. The disadvantage is that this mode is susceptible for replay attacks or traffic analysis: a block containing constant data is encrypted every time to the same cipher block provided the same key is used. *(B.A.Forouzan,2008)*

#### 5.3.4.2. CBC Modey

Cipher block chaining mode: This mode can be a solution against the replay attacks on the ECB mode. Here the output (the ciphertext) of the previous block and the plaintext of the current block are XOR-ed and subsequently encrypted and transmitted/stored. The output of the current

Block is used for the XOR-operation with the next plaintext block. The decryption works in reverse order: the current ciphertext block is decrypted, the result XOR-ed with the previous ciphertext block, the result is the original plaintext block. A careful reader might have noticed that there is a problem at the beginning, to solve this a dummy block must be transmitted first to start the chain, this block is also called “initialisation vector” (IV). Transmission errors have a slightly larger impact, a flipped bit leads to completely different plaintext version of the current block (as in ECB), and it changes a single bit of plaintext in the next block. *(B.A.Forouzan,2008)*

#### 5.3.4.3. CFB Mode

Cipher feedback mode: This mode and the following modes have been invented to transform a block cipher into a stream cipher. These modes can be used when data must be encrypted with a size less than the block length. Again, at the beginning an initialization vector is used, it is put into a shift register (in the following we assume that it shifts from right to left). The contents of this register is encrypted, the left-most n bits are used for an XOR-operation with plaintext of size n. The resulting cipher data (again size ri) is sent or stored and additionally put back into the shift register on the right side. The decryption operation is almost identical to the encryption operation: it is initialised with the same IV. The received cipher data of size n is put into the shift register, its contents encrypted and the left-most n bits of its output XOR-

ed with the just-received cipher data. Care must be taken that the IV is unique for every message, but there are no absolute requirements to keep it secret. *(B.A.Forouzan,2008)*

#### 5.3.4.4. OFB mode

Output feedback mode: is similar to CFB. Whereas in CFB-mode the result of the XOR-function is put back into the shift register, in OFB the loop does not involve data from a user, the feedback-loop comprises just the shift register and the encryption function: the result of the encryption function is fed back to the shift register. This has the advantage that the key stream can be computed independently of the data that must be encrypted. *(E.Aribas, 2017)*

#### 5.3.4.5. CTR Mode

Counter mode: Similar to OFB, but here no shift register is used. Instead, a counter value is used as the input to encryption function, after each encryption the counter is changed, usually it is incremented by one. An advantage of this mode is that random access mode to some data is possible, without the drawbacks of ECB. *(E.Aribas, 2017)*

### 5.3.5. AES Rijndael

AES can be performed using different key sizes: 128-, 192-, and 256-bit. NIST's competition aimed to find an algorithm with some very strong characteristics, such as it should be operating in blocks of 128 bits of input or it should be able to be used on different kinds of hardware, from 8-bit processors (used also in a smart card) to 32-bit architectures, commonly adopted in personal computers. Finally, it should be fast and very robust. (I.Dubinsky,2023)

At the beginning of the algorithm, a round key is first XORed with the input data. This step is called *AddRoundKey*. Then, iterating through rounds, the algorithm performs the *SubBytes* step, where an S-box mechanism with carefully designed substitution tables is applied to the data bits. After the substitution, the algorithm treats the data as a matrix and performs a *ShiftRows* operation (reshuffling rows so that elements of each column in the matrix intermingle) and a *MixColumns* step which applies a linear transformation to each column of the matrix. Finally, the result is XORed with the round key in another *AddRoundKey* invocation.After the substitution, the algorithm treats the data as a matrix and performs a *ShiftRows* operation (reshuffling rows so that elements of each column in the matrix intermingle) and a *MixColumns* step which applies a linear transformation to each column of the matrix. Finally, the result is XORed with the round key in another *AddRoundKey* invocation. (I.Dubinsky,2023)

### 5.3.6. DES

The Data Encryption Standard (DES) is the most well-known symmetric-key block cipher. It was selected by the National Bureau of Standards as an official Federal Information Processing Standard (FIPS) for the United States in 1976, and it subsequently enjoyed widespread use internationally. *(F.Abd El-Samie,2013)*

The DES is a block cipher, which encrypts data in 64-bit blocks. A 64-bit block of plaintext goes at one end of the algorithm, and a 64-bit block of ciphertext comes out at the other end. The same algorithm and key of size 56 bits are used for both encryption and decryption except for minor differences in the key schedule. The key is usually expressed as a 64-bit number, but every eighth bit, one bit is used for parity checking and is ignored. These parity bits are the least-significant bits of the key bytes. The key can be any 56-bit number and can be changed at any time, although some selections can be considered weak keys. *(F.Abd El-Samie,2013)*

The DES is based on four basic operations: expansion, permutation, XOR, and substitution. The data to be encrypted are first divided into 64-bit blocks and fed into an Initial Permutation (IP) stage, in which each block is divided into two subblocks, each with a 32-bit length. The right subblock is fed into a Feistel function (f-function). It operates on half a block (32 bits) at a time and consists of four stages.

**Expansion.** The 32-bit half block is expanded to 48 bits using the expansion permutation, denoted as E in the diagram, by duplicating half of the bits. The output consists of eight 6-bit ($8 \times 6 = 48$ bits) pieces, each containing a copy of 4 corresponding input bits plus a copy of the immediately adjacent bit from each of the input pieces to either side.

**Key mixing.** The result is combined with a subkey using an XOR operation. Sixteen 48-bit subkeys, one for each round, are derived from the main key using a key-schedule mechanism.

**Substitution.** After mixing with the subkey, the block is divided into eight 6-bit pieces before processing by the Substitution boxes (S-boxes). Each of the eight S-boxes replaces its six input bits with four output bits according to a nonlinear transformation, provided in the form of a lookup table. The S-boxes provide the core of security of the DES. *(F.Abd El-Samie,2013)*

### 5.3.7. Double DES

A naive way of improving the security of a block cipher algorithm is to encrypt each block twice with two different keys. First, encrypt a block with the first key and then encrypt the resulting ciphertext with the second key. Decryption is the reverse process. In the Double DES encryption algorithm, each 64-bit block of data is encrypted twice with the DES algorithm, first with a key K1 and then with another key K2. The scheme involves a key of 112 bits. *(F.Abd El-Samie,2013)*

The resultant doubly encrypted ciphertext block should be much harder to decrypt using an exhaustive search. Instead of 256 attempts, it requires 2128 attempts to find the key and 2112 attempts to break the encryption. In 1981, Merkle and Hellman declared their "meet-in-the-middle" attack, which proved the weakness of the Double DES algorithm. The "meet-in-the-middle" attack is a known plaintext attack that requires that an attacker have both a known piece of plaintext and the corresponding ciphertext. The attack requires storing 256 intermediate results when trying to crack a message that has been encrypted with the double DES. Merkle and Hellman developed a time-memory trade-off that could break this double-DES encryption scheme in 256+1 trials, not in 2112 trials. *(F.Abd El-Samie,2013)*

### 5.3.8. Triple DES

The dangers of the Merkle-Hellman "meet-in-the-middle" attack can be circumvented by performing three block encryption operations. *(F.Abd El-Samie,2013)*

This method is called Triple DES. Triple DES is performed by executing the DES three times, producing an effective key size of 168 bits. In the Triple DES, each 64-bit block of data is encrypted three times with the DES algorithm. In practice, the most common way to perform the Triple DES is :

1.Encrypt with key 1.

2.Decrypt with key 2.

3.Encrypt with key 3.

**To decrypt, reverse the steps:**

1.Decrypt with key 3.

2.Encrypt with key 2.

3.Decrypt with key 1.

For several applications, we can use the same key for both key 1 and key 3 without creating

a significant vulnerability. The choice between Single, Double, and Triple DES is a trade-off between performance and security requirements. *(F.Abd El-Samie,2013)*

## 6. Hash Functions

A cryptographic hash function is a hash function that is considered practically impossible to invert, that is, to recreate the input data from its hash value alone. The input data is often called the message, and the hash value is often called the message digest or simply the digest (Kaur and Singh, 2012). *(F. Jabali,2016)*

There are a number of hash functions types that can be used in cryptography. Two series of hash functions MD2, MD4, and MD5, and the Secure Hash Algorithm (SHA), a standard algorithm, that makes a larger (60-bit) message digest and is similar to MD4. These different types of hash functions have different features and shortages. *(F. Jabali,2016)*

Common Hash algorithms include the following:

- Message Digest (MD) algorithms: A series of byte- oriented algorithms that produces a 128-bit hash value from an arbitrary-length message.
- Secure Hash Algorithm (SHA): the SHA function is hash algorithm in which n-bit hash produces n-bit length finger print from the arbitrary length data. SHA- 1 produces message digest160, SHA-256, SHA-512, (Guo et al., 2010).

## 7. Digital Signatures

A digital signature is a cryptographic checksum that can only be created by the owner of the private signing key but can be verified by any entity with the corresponding public verification key. Thus, digital signatures can be used to guarantee the authenticity of messages. (J.Schwenk,2022)

A digital signature has similarities to a handwritten signature. This includes the fact that, just like a digital signature, a handwritten signature can only be created by one person but can be verified by many. But there are also differences: While the handwritten signature always looks approximately the same and is only associated with a document by being on the same sheet of paper, the value of the digital signature directly depends on the value of the document

to be signed. (J.Schwenk,2022)

A conventional signature has the following salient characteristics: relative ease of establishing that the signature is authentic, the difficulties of forging a signature, the non-transferability of the signature, the difficulty of altering the signature, and the non-repudiation of the signature to ensure that the signer cannot deny signing. (R.Achary,2021)

A digital signature should have all the aforementioned features of conventional signatures and also the ability to verify the author, date, and time of signature and authenticate message contents. Their applications are also extended to secure e-mail and credit card transactions over the Internet.it is desirable for it to have the following properties: (R.Achary,2021)

- The signature must be a bit pattern that depends on the message being signed; i.e., for the same sender, the digital signature is different for different documents.
- The signature must use some information that is unique to the sender to prevent both forgery and denial.
- It must be relatively easy produce.
- It must be relatively easy to recognize and verify the authenticity of the digital signature.
- It must be computationally infeasible to forge a digital signature either by constructing a new message for an existing digital signature or developing a fraudulent digital signature for the given message.

## 7.1. RSA Signatures

We have described the oldest public-key system, the RSA system. This system can also be used to generate digital signatures. The idea is very simple. Alice signs the document *m* by computing the signature s = s(d, m) = $m^d$ mod *n.* Here *d* is Alice's secret exponent and *n* is the public RSA modulus. Bob verifies the signature by computing $s^e$ mod *n* — $m'^{*1}$ = *m* mod n. *(J.Buchmann,2000)*

Why is this a signature? By raising the randomly looking number s to the power e, Bob can recover the document *m.* Therefore, s can be considered to be the eth root of the document *m,* and currently computing eth roots of an integer *m* mod *n* without the knowledge of *d* is infeasible. But Alice is the only person who knows *d,* so Alice must have computed s and

thereby signed m. *(J.Buchmann,2000)*

**Key generation:** The key generation for RSA signatures is the same as the key generation for RSA encryption. Alice chooses independently two large random primespand q and an exponent e with 1 < e < (p—l)(q —1) and gcd(e, (p — l)(q — 1)) = 1. She computes n = pq and del with 1 < d < (p - l)(q - 1} and de = 1 mod (p - l)(r/ - 1). Her public key is (n, e) and her secret key is d. *(J.Buchmann,2000)*

### 7.2. ElGamal Signature

The ElGamal signature scheme is similar to the ElGamal cryptosystem,although it is not constructed from it. Its security is based on the difficulty of computing discrete logarithms in (Z/pZ)*, where *p* is a prime number. *(J.Buchmann,2000)*

**Key generation:** Key generation is the same as for the ElGamal encryption system.Alice generates a large random prime *p* and a primitive root *g* mod *p*. She also chooses *a* randomly in the set {1,2, ...,p - 2} and computes $A = g^a \bmod p$. Her private key is *a*. Her public key is (p,g, A). *(J.Buchmann,2000)*

## 8. Cryptanalysis and Brute-Force Attack

Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:

**Cryptanalysis**: Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. *(M.Alwuthaynani,2023)*

**Brute-force attack**: The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. *(M.Alwuthaynani,2023)*

If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

We first consider cryptanalysis and then discuss brute-force attacks.

Table 1 summarizes the various types of cryptanalytic attacks based on the amount of

information known to the cryptanalyst. The most difficult problem is presented when all that is available is the ciphertext only. In some cases, not even the encryption algorithm is known, but in general, we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, an EXE file, a Java source listing, an accounting file, and so on. *(M.Alwuthaynani,2023)*

The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of information to work with. In many cases, however, the analyst has more information. The analyst may be able to capture one or more plaintext messages as well as their encryptions. Or the analyst may know that certain plaintext patterns will appear in a message. For example, a file that is encoded in the Postscript format always begins with the same pattern, or there may be a standardized header or banner to an electronic funds transfer message, and so on. All these are examples of known plaintext. With this knowledge, the analyst may be able to deduce the key based on how the known plaintext is transformed. *(M.Alwuthaynani,2023)*

Closely related to the known-plaintext attack is what might be referred to as a probable-word attack. If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message. However, if the opponent is after some very specific information, then parts of the message may be known. For example, if an entire accounting file is being transmitted, the opponent may know the placement of certain keywords in the header of the file. As another example, the source code for a program developed by Corporation X might include a copyright statement in some standardized position. *(M.Alwuthaynani,2023)*

Table 1 lists two other types of attack: chosen ciphertext and chosen text. These are less commonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack.

Only relatively weak algorithms fail to withstand a ciphertext-only attack. Generally, an encryption algorithm is designed to withstand a known-plaintext attack. Two more definitions are worthy of note. An encryption scheme is unconditionally secure if the ciphertext generated

by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. That is, no matter how much time an opponent has, he or her can't decrypt the ciphertext simply because the required information is not there. Except for a scheme known as the one-time pad, no encryption algorithm is unconditionally secure. Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

- The cost of breaking the cipher exceeds the value of the encrypted information.
- The time required to break the cipher exceeds the useful lifetime of the information.

*Table 2. 1 Types of cryptanalytic attacks (M.Alwuthaynani,2023)*

| Type of Attack | Known to Cryptanalyst |
|---|---|
| Ciphertext Only | • Encryption algorithm<br>• Ciphertext |
| Known Plaintext | • Encryption algorithm<br>• Ciphertext<br>• One or more plaintext–ciphertext pairs formed with the secret key |
| Chosen Plaintext | • Encryption algorithm<br>• Ciphertext<br>• Plaintext message is chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key |
| Chosen Ciphertext | • Encryption algorithm<br>• Ciphertext<br>• Ciphertext chose by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |
| Chosen Text | • Encryption algorithm<br>• Ciphertext<br>• Plaintext message is chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key<br>• Ciphertext chose by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |

## 9. Information security and cryptography

The concept of information will be taken to be an understood quantity. To introduce

cryptography, an understanding of issues related to information security, in general, is necessary. Information security manifests itself in many ways according to the situation and requirement. Regardless of who is involved, to one degree or another, all parties to a transaction must have confidence that certain objectives associated with information security have been met. Some of these objectives are listed. *(Alfred J.Menezes,1996)*

Over the centuries, an elaborate set of protocols and mechanisms has been created to deal with information security issues when the information is conveyed by physical documents. Often the objectives of information security cannot solely be achieved through mathematical algorithms and protocols alone, but require procedural techniques and abidance of laws to achieve the desired result. For example, the privacy of letters is provided by sealed envelopes delivered by an accepted mail service. The physical security of the envelope is, for practical necessity, limited and so laws are enacted which make it a criminal *(Alfred J.Menezes,1996)*

- **Privacy or confidentiality**: keeping information secret from all but those who are authorized to see it.
- **Data integrity**: ensuring information has not been altered by unauthorized or unknown means.
- **Entity authentication or identification**: corroboration of the identity of an entity (e.g., a person, a computer terminal, a credit card, etc.).
- **Message** authentication: corroborating the source of information; also known as data origin authentication
- **Signature**: a means to bind information to an entity.
- **Authorization**: conveyance, to another entity, of official sanction to do or be something.
- **Validation**: a means to provide timeliness of authorization to use or manipulate information or resources.
- **Access control**: restricting access to resources to privileged entities.
- **Certification**: endorsement of the information by a trusted entity.

- **Timestamping**: recording the time of creation or existence of information.
- **Witnessing**: verifying the creation or existence of information by an entity other than the creator.
- **Receipt**: an acknowledgment that information has been received.
- **Confirmation**: an acknowledgment that services have been provided.
- **Ownership**: a means to provide an entity with the legal right to use or transfer a resource to others.
- **Anonymity**: concealing the identity of an entity involved in some process.
- **Non-repudiation**: preventing the denial of previous commitments or actions
- **Revocation**: retraction of certification or authorization.

Information was typically stored and transmitted on paper, much of it now resides on magnetic media and is transmitted via telecommunications systems, some wireless. What has changed dramatically is the ability to copy and alter information. One can make thousands of identical copies of a piece of information stored electronically and each is indistinguishable from the original. With information on paper, this is much more difficult. What is needed than for a society where information is mostly stored and transmitted in electronic form is a means to ensure information security that is independent of the physical medium recording or conveying it and such that the objectives of information security rely solely on digital information itself. *(Alfred J.Menezes,1996)*

Analogs of the “paper protocols” currently in use are required. Hopefully, these new electronic-based protocols are at least as good as those they replace. There is a unique opportunity for society to introduce new and more efficient ways of ensuring information security. Much can be learned from the evolution of the paper-based system, mimicking those aspects which have served us well and removing the inefficiencies. *(Alfred J.Menezes,1996)*

## 10.Conclusion

In conclusion, cryptography has evolved significantly from ancient hieroglyphs to sophisticated mathematical algorithms, ensuring secure communication and information protection. Its history, techniques, and applications provide us with valuable insights into the world of cryptography and its vital role in safeguarding sensitive data.

As we conclude our study, it is evident that cryptography continues to be indispensable in safeguarding sensitive information. Its continued advancements and ongoing research are crucial in addressing emerging security challenges and adapting to evolving threats.

By comprehending the principles, techniques, and algorithms of cryptography, we are better equipped to navigate the complex landscape of information security. Through continued innovation and the application of cryptographic best practices, we can ensure the confidentiality, integrity, and trustworthiness of data in an ever-changing and interconnected world.

Ultimately, cryptography remains an essential cornerstone of modern information security, enabling us to communicate securely, protect sensitive information, and build trust in our digital interactions.

# Chapter 3: Results & discussion

## 1. Introduction

This chapter introduces an image encryption algorithm based on the principle of Rubik's cube. The proposed algorithm aims to overcome the limitations of traditional encryption approaches in image protection. The details of the encryption processes utilizing Rubik's cube are discussed in this chapter, highlighting its potential to provide efficient and secure image encryption.

## 2. Rubik's Cube Image Encryption

Let $I_o$ represent an α-bit Gray-scale image of the size M × N. Here, $I_o$ represent the pixels values matrix of image $I_o$. The steps of encryption algorithm are as follows:

1) Generate randomly two vectors KR and KC of length M and N, respectively. Element KR (i)and KC (j)Each take a random value of the set $A = \{0, 1, 2, \ldots, 2\alpha - 1\}$. Note that both KR and KC must not have constant values.

2) Determine the number of iterations, $ITER_{max}$, and initialize the counter ITER at 0.

3) Increment the counter by one: $ITER = ITER + 1$.

4) For each row i of image $I_o$

   a. compute the sum of all elements in the row i, this sum is denoted by α(i)

$$\alpha(i) = \sum_{j=1}^{N} I_0(i,j), \quad i = 1,2,\ldots,M, \tag{3.1}$$

   b. Compute modulo2 of α(i), denoted by $M_{\alpha(i)}$

   c. row i is left, or right, circular-shifted by $K_R$ (i) positions (image pixels are moved $K_R$ (i)positions to the left or right direction, and the first pixel moves in last pixel.), according to the following:

if $M\alpha(i) = 0 \rightarrow$ right circular shift (3.2)

else → left circular shift

5) For each column j of image $I_o$ ,

a. compute the sum of all elements in the column j , this sum is denoted by $\beta(j)$,

$$\beta(j) = \sum_{i=1}^{M} I_0(i,j), \quad j = 1,2,\dots,N, \tag{3.3}$$

b. compute modulo2 of $\beta(j)$, denoted by $M_{\beta(j)}$.

column j is down, or up, circular-shifted by $K_C$ (i) positions, according to the following:

if $M\beta(j) = 0 \rightarrow$ up circular shift (3.4)

else $\rightarrow$ down circular shift

Steps 4 and 5 above will create a scrambled image, denoted by ISCR .

Using vector KC, the bitwise XOR operator is applied to each row of scrambled image $I_{SCR}$ using the following expressions:

$$I_1(2i-1,j) = I_{SCR}(2i-1,j) \oplus K_c(j), \tag{3.5}$$

$$I_1(2i,j) = I_{SCR}(2i,j) \oplus rot\ 180\ (K_c(j)),$$

where $\oplus$ and rot 180($K_C$) represent the bitwise XOR operator and the flipping of vector $K_C$ from left to right, respectively.

Using vector $K_R$, the bitwise XOR operator is applied to each column of image $I_1$ using the following formulas:

$$I_{ENC}(i,2j-1) = I_1(i,2j-1) \oplus K_R(j), \tag{3.6}$$

$$I_{ENC}(i,2j) = I_1(i,2j) \oplus rot\ 180\ K_R(j),$$

with rot 180($K_R$) indicating the left to right flip of vector $K_R$.

If $ITER = ITER_{max}$ , then encrypted image $I_{ENC}$ is created and encryption process is done; otherwise, the algorithm branches to step 3.

Vectors $K_R$, $K_C$ and the max iteration number $ITER_{max}$ are considered as secret keys in the

proposed encryption algorithm. However, to obtain a fast encryption algorithm it is preferable to set $ITER_{max} = 1$ (single iteration). Conversely, if $ITER_{MAX} > 1$, then the algorithm is more secure because the key space is larger than for $ITER_{max} = 1$. Nevertheless, in the simulations presented in Section 3, the number of iterations $ITER_{max}$ was set to one.

Rubik's Cube Decryption Algorithm. The decrypted image, $I_o$, is recovered from the encrypted image, $I_{ENC}$, and the secret keys, $K_R$, $K_C$, and $ITER_{max}$ as follows in the following.

1) Initialize $ITER = 0$.

2) Increment the counter by one: $ITER = ITER + 1$.

3) The bitwise XOR operation is applied on vector $K_R$ And each column of the encrypted image $I_{ENC}$ as follow

$$I_1\ (i, 2j-1) = I_{ENC}(i, 2j-1) \oplus K_R(j), \quad (3.7)$$

$$I_1\ (i, 2j) = I_{ENC}(i, 2j) \oplus rot\ 180\ (K_R(j)),$$

4) Then, using the KC vector, the bitwise XOR operator is applied to each row of image $I_1$:

$$I_{SCR}\ (2i-1, j) = I_1(2i-1, j) \oplus K_c(j), \quad (3.8)$$

$$I_{SCR}\ (2i, j) = I_1(2i, j) \oplus rot\ 180\ K_c(j),$$

5) For each column j of the scrambled image $I_{SCR}$,

   a. compute the sum of all elements in that column j, denoted as $\beta_{SCR}(j)$:

$$\beta_{SCR}(j) = \sum_{i=1}^{M} I_{SCR}(i, j), \quad j = 1,2, \dots, N, \quad (3.9)$$

   b. compute modulo 2 of $\beta_{SCR}(j)$, denoted by $M_{\beta_{SCR}(j)}$,

   c. column j is down, or up, circular-shifted by $K_C$ (i) positions according to the following:

$$\text{if} \quad M_{\beta_{SCR}} = 0 \rightarrow \text{up circular shift} \quad (3.10)$$

Else → down circular shift

6) For each row i of scrambled image $I_{SCR}$,

   a. compute the sum of all elements in row i, this sum is denoted by $\alpha_{SCR}\ (i)$:

$$\alpha_{SCR}(j) = \sum_{j=1}^{N} I_{SCR}(i,j), \quad j = 1,2,\dots,M, \tag{3.11}$$

   b. compute modulo 2 of $\alpha_{SCR}(j)$, denoted by $M_{SCR}(j)$

   c. row $i$ is then left, or right, circular-shifted by $K_R\ (i)$ according to the following:

$$\text{if} \quad M_{\alpha_{SCR}(j)} = 0 \rightarrow \text{right circular shift} \tag{3.12}$$
$$\text{Else} \rightarrow \text{left circular shift}$$

7) If $ITER = ITER_{max}$, then image IENC is decrypted and the decryption process is done; otherwise, the algorithm branches back to step 2. *(K. Loukhaoukha,2012)*

## 3. Experimental Results

In this section, we present the tests that were conducted to assess the efficiency and security of the proposed image encryption algorithm. These tests involve visual testing and security analysis.

### 3.1. Visual Testing

For visual testing, five Gray-scale images were used, Table 1 shows these images:

*Table 3. 1: Images Resolution*

| Image Name | Image Size |
|---|---|
| *Lena* | $1024 \times 1024$ |
| *Baboon* | $512 \times 512$ |
| *Boat* | $256 \times 256$ |
| *Peppers* | $512 \times 512$ |
| *Cameraman* | $64 \times 64$ |

Figure 3.1 depicts these test images ‘’Lena, Baboon, Boat, Peppers, Cameraman’ ’as well as the images encrypted and decrypted using Rubik’s cube algorithm. From this figure, we can see that there is no perceptual similarity between original images and their encrypted counterparts, although after the decryption we can see the images restore its original format.


*Figure 3. 1 Original, encrypted and decrypted images*

The encrypted image should greatly differ from its original form. In general, two difference measures are used to quantify this requirement.

## 3.2. NPCR & UACI

Table 2 gives the NPCR and UACI values for the original images and their encrypted versions. The values are very close to unity for the NPCR measure. The UACI values are also appropriate. The high percentage values of the NPCR measure indicate that the pixels positions have been randomly changed.

*Table 3. 2: Difference measures between original and encrypted images*

| *Image* | *NPCR (%)* | *UACI (%)* |
|---|---|---|
| *Lena* | 99.60 | 50.08 |
| *Baboon* | 99.61 | 49.94 |
| *boat* | 99.61 | 50.06 |
| *peppers* | 99.58 | 49.94 |
| *Cameraman* | 99.63 | 49.91 |

Furthermore, the UACI values show that almost all pixel Gray-scale values of encrypted image have been changed from their values in the original images, making the original and encrypted image pixels more difficult to discriminate.

## 4. Statistical Analysis

The statistical analysis is done by performing two series of tests: **histograms** analysis of the encrypted images and the **correlations** computation of the adjacent pixels in encrypted images.

### 4.1. Histogram Analysis

Figure 2 represents the histograms of the original, encrypted and decrypted images illustrated previously in Figure (3.2).

The histograms of the encrypted images are almost uniform and are significantly different from that of the five original images, and the histogram of the decrypted images are identical to the original images.

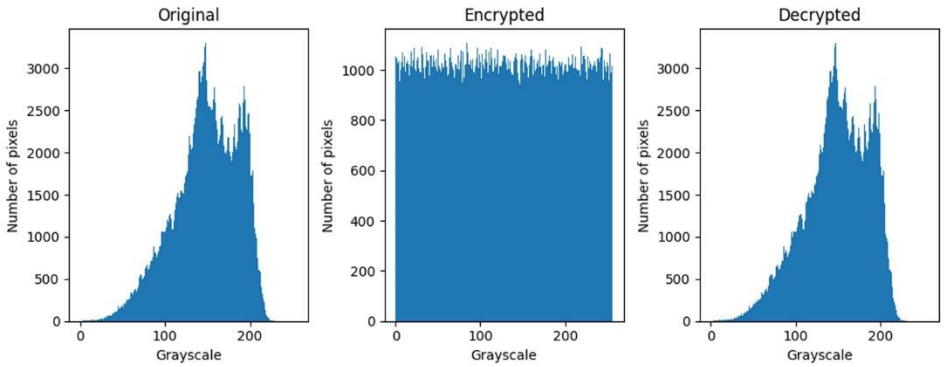



*Image :Baboon*

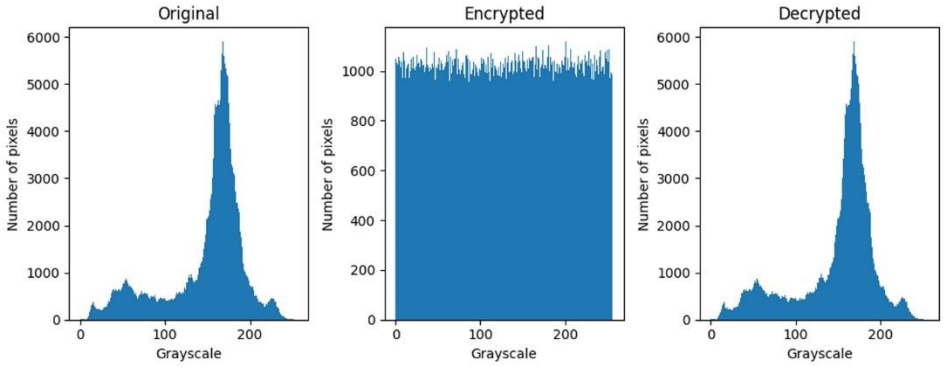



*Image :Boat*

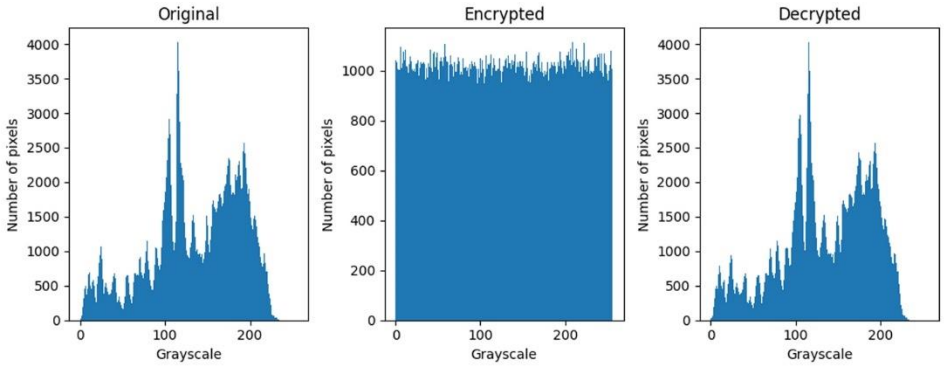



*Image :Peppers*

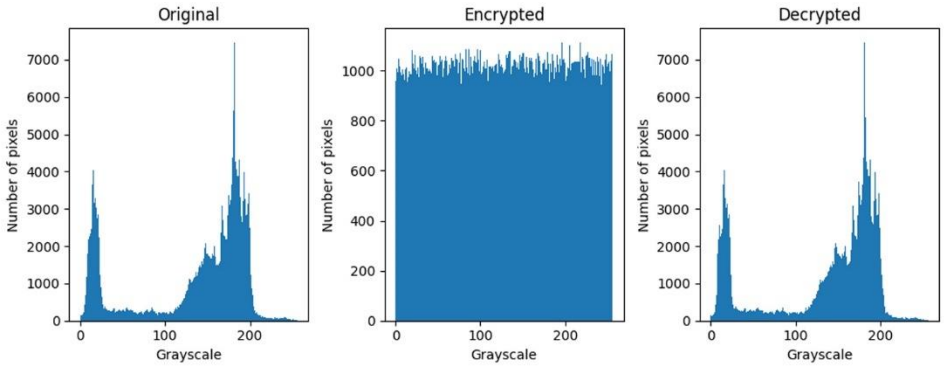



*Image :Cameraman*

*Figure 3. 2 Histograms of original, encrypted and decrypted images.*

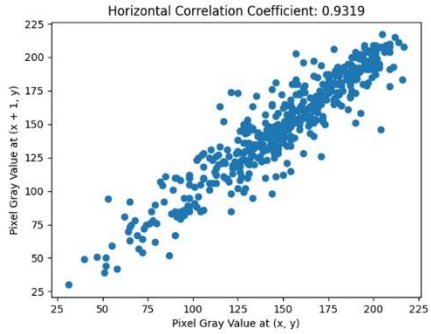
## 4.2. Adjacent Pixel Correlation Analysis

Table 3.3 gives the correlation coefficient values of adjacent pixels in the horizontal, vertical, and diagonal directions of the original images and their encrypted versions. It is clear that for the original images, the coefficient correlation values are very high (close to one) contrary to those observed for the encrypted images. This confirms that adjacent pixels in the original images are strongly correlated. However, for the encrypted images, those values are close to zero or negative, which means that the adjacent pixels (horizontal, vertical and diagonal directions) are very weakly correlated.

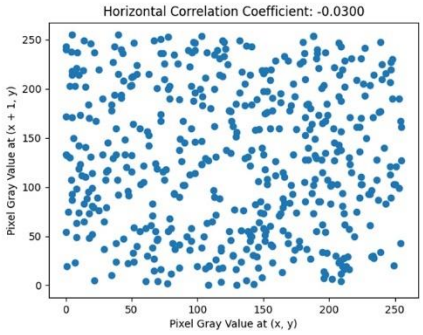
*Table 3. 3 Correlation coefficients between adjacent pairs of pixels for original and encrypted images*

| ***Correlation*** | ***Horizontal*** | ***Vertical*** | ***Diagonal*** |
|---|---|---|---|
| *Original image : Lena* | 0.98 | 0.98 | 0.96 |
| *Encrypted image : Lena* | 0.03 | -0.11 | 0.012 |
| *Original image: Baboon* | 0.94 | 0.92 | 0.89 |
| *Encrypted image: Baboon* | -0.04 | -0.02 | -0.03 |
| *Original image: Boat* | 0.97 | 0.97 | 0.96 |
| *Encrypted image: Boat* | -0.03 | 0.01 | 0.026 |
| *Original image: Peppers* | 0.97 | 0.96 | 0.96 |
| *Encrypted image: Peppers* | -0.04 | 0.0092 | 0.0092 |
| *Original image: Cameraman* | 0.97 | 0.99 | 0.97 |
| *Encrypted image: Cameraman* | -0.06 | -0.0059 | -0.019 |

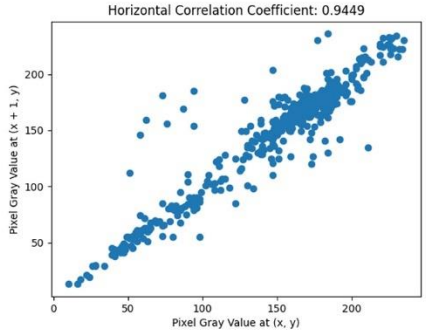
Figure 3.3 illustrates the correlation distributions of the horizontal adjacent pixels of the original images and the corresponding encrypted images using the proposed algorithm. One can see from Figure 3.3 those adjacent pixels in encrypted images are indeed very weakly correlated.



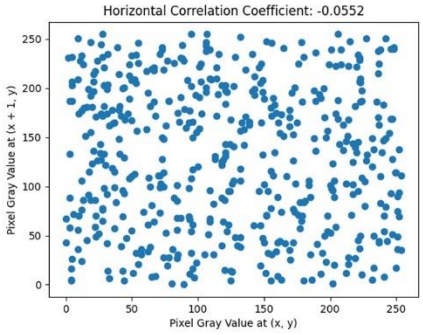

*Original : Baboon*



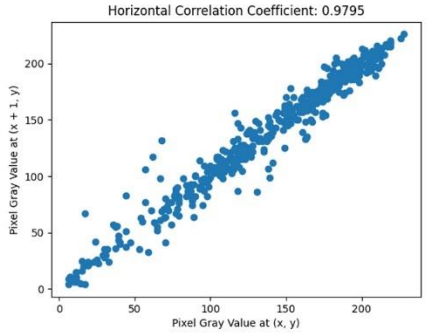

*Encrypted : Baboon*



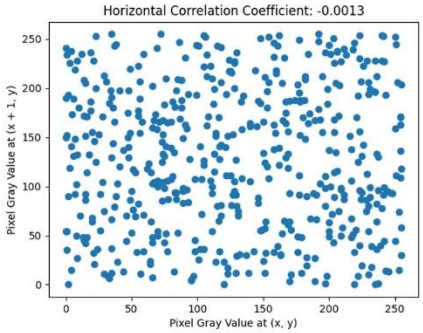

*Original : Boat*



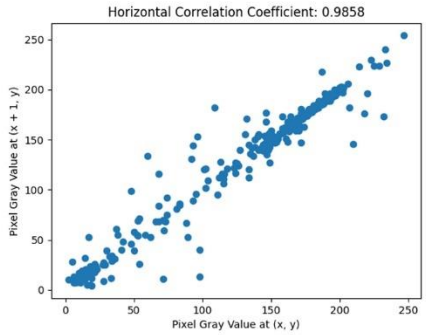

*Encrypted : Boat*



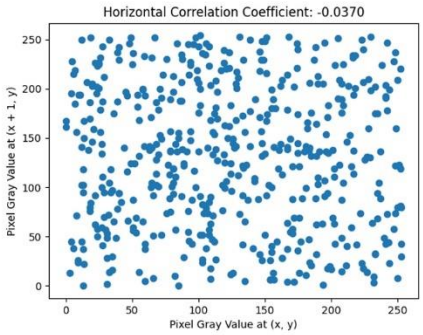

*Original : Peppers*




*Encrypted : Peppers*




*Original : Cameraman*




*Encrypted : Cameraman*

*Figure 3. 3 Correlation distribution of the pairs horizontal to adjacent pixels*

## 4.3. Entropy Analysis

Table (3.4) gives the entropy values of the five original images and those of their encrypted versions.

*Table 3. 4 :Comparison of entropy values of original images and their encrypted version*

| *Image* \ *Entropy Value* | *Original* | *Encrypted* |
|---|---|---|
| *Lena* | 7.37 | 7.99 |
| *Baboon* | 7.19 | 7.99 |
| *Boat* | 7.13 | 7.99 |
| *Peppers* | 7.54 | 7.99 |
| *Cameraman* | 7.08 | 7.99 |

From Table (3.4), we note that the entropy values of original images are far from ideal value of entropy since information sources are highly redundant and thus rarely generate uniformly distributed random messages. On the other hand, the entropy values of the encrypted images are very close to the ideal value of 8, which means that the proposed encryption algorithm is highly robust against entropy attacks.

## 4.4. PSNR

*Table 3. 5: PSNR values of original images and their encrypted version.*

| *Image* | *PSNR Value (dB)* |
|---|---|
| *Lena* | 47.92 |
| *Baboon* | 48.22 |
| *boat* | 40.62 |
| *peppers* | 49.15 |
| *Cameraman* | 40.65 |

A Peak Signal-to-Noise Ratio (PSNR) of over 40dB signifies a very high-quality image with minimal noise and distortion. It means that the decrypted image closely resembles the original, and the human eye can hardly notice any differences.

In practice, the acceptable or "good" range of PSNR values can vary depending on the specific application and the nature of the image. Here are some general guidelines:

1. **Excellent Quality**: PSNR > 40 dB
2. **Good Quality**: PSNR 30-40 dB
3. **Fair Quality**: PSNR 20-30 dB
4. **Poor Quality**: PSNR < 20 dB

In image encryption, a high PSNR value suggests that the encrypted images closely resemble the original images, indicating minimal distortion or loss of visual quality. This can be seen as a positive outcome, as it implies that the encryption method successfully preserves the content and details of the images while protecting them from unauthorized access. *(S.E.Naffouti,A.Kricha, 2022)*

## 5. Conclusion

In summary, the image encryption algorithm based on the Rubik's Cube method offers a combination of security, randomness, and fidelity. It successfully obscures the original image content, disrupts correlations, and maintains a high level of visual quality. These characteristics make it a reliable solution for protecting sensitive image data.

# *General conclusion*

In conclusion, this research has explored the application of the Rubik's Cube principle in image encryption, with the aim of enhancing the security of digital images containing sensitive information.

The Rubik's Cube-based encryption method has proven to be effective in providing robust security. By leveraging the complexity and nonlinearity of the Rubik's Cube algorithm, the encryption process introduces a high level of protection against unauthorized decryption. The reversible nature of the encryption technique ensures the faithful reconstruction of the original image during decryption, preserving image integrity. The integration of the Rubik's Cube principle adds an additional layer of security, making the encrypted image resistant to cryptographic attacks.

Experimental evaluations have confirmed the efficiency and effectiveness of the proposed method in terms of encryption performance, resistance to attacks, and preservation of image quality. These results validate the superiority and practicality of the Rubik's Cube-based image encryption approach. The implications of this research are significant, particularly in applications that require secure image transmission. The successful integration of the Rubik's Cube principle offers promising advancements in securing digital images and protecting their confidentiality.

In summary, this research contributes to the field of image encryption by presenting a novel approach that combines the Rubik's Cube principle with cryptographic techniques. The findings have practical implications for securing digital images in various domains, highlighting the importance of protecting sensitive visual information in today's interconnected world.

When considering potential improvements for the algorithm., These are some perspectives we can add:

Image Format Support: Expand the algorithm to support encryption and decryption of images in different formats, not just grayscale images. This could involve modifying the code to handle RGB or other color channels, allowing for encryption and decryption of colored

images while preserving their original form.

Encryption Strength: Evaluate the algorithm's encryption strength and consider incorporating stronger encryption techniques. This could involve exploring more advanced cryptographic algorithms or incorporating additional layers of encryption, such as using multiple encryption rounds or combining symmetric and asymmetric encryption methods.

Performance Optimization: Assess the performance of the algorithm and identify areas for optimization. This could involve analysing the code for any inefficiencies, considering algorithmic optimizations, or exploring parallel processing techniques to enhance encryption and decryption speed.

User Interface and Integration: Develop a user-friendly interface or integrate the algorithm into existing image processing software, making it more accessible and convenient for users to encrypt and decrypt their images. This could include features such as drag-and-drop functionality, batch processing, or integration with popular image editing tools.

# *Perspective*

When considering potential improvements for the algorithm.These are some perspectives we can add :

Image Format Support: Expand the algorithm to support encryption and decryption of images in different formats, not just grayscale images. This could involve modifying the code to handle RGB or other color channels, allowing for encryption and decryption of colored images while preserving their original form.

Encryption Strength: Evaluate the algorithm's encryption strength and consider incorporating stronger encryption techniques. This could involve exploring more advanced cryptographic algorithms or incorporating additional layers of encryption, such as using multiple encryption rounds or combining symmetric and asymmetric encryption methods.

Performance Optimization: Assess the performance of the algorithm and identify areas for optimization. This could involve analyzing the code for any inefficiencies, considering algorithmic optimizations, or exploring parallel processing techniques to enhance encryption and decryption speed.

User Interface and Integration: Develop a user-friendly interface or integrate the algorithm into existing image processing software, making it more accessible and convenient for users to encrypt and decrypt their images. This could include features such as drag-and-drop functionality, batch processing, or integration with popular image editing tools.

# *Bibliography*

*(A.Bhute,2013) Avinash N Bhute, Content Based Image Indexing and Retrieval, International Journal of Graphics & Image Processing, 2013*

*(Baskar, 2023) A Baskar, Muthaiah Rajappa, Digital Image Processing, CRC Press,2023.*

*(Behrouz, 2008) Behrouz.A.Forouzan, Introduction to Cryptography and Network Security, Publisher McGraw-Hill Higher Education,2008)*

*(C,Solomon,2011) Chris Solomon, Fundamentals of Digital Image Processing, Wiley & Sons,2011.*

*(E.Aribas, 2017)Erke Aribas, Comparative Analysis of Block Cipher Modes of Operation,Conference Paper · November 2017*

*(F. Abd El-Samie,2013) Fathi E. Abd El-Samie, Image Encryption, Publisher CRC Press,2013 .*

*(Ferguson, 2010) Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno, Cryptography Engineering, Published by Wiley Publishing,2010*

*(J. Buchmann, 2000) Johannes Buchmann, Introduction to Cryptography, Publisher Springer,2000.*

*(J. Schwenk, 2022) Jörg Schwenk, Guide to Internet Cryptography,2022, Publisher Springer.*

*(J.Aumasson, 2018) Jean-Philippe Aumasson, Serious Cryptography A Practical Introduction To Modern Encryption, Publisher no starch press San Francisco, 2018.*

*(J.Katz, 2007)Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography: Principles and Protocols, Publisher CRC Press,2007.*

*(J.Katz, 2014)Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography Second Edition Publisher CRC Press,2014.*

*(Jähne,2002) Bernd Jähne, Digital Image Processing 5th revised and extended edition, Springer,2002.*

*(K. Loukhaoukha,2012) Khaled Loukhaoukha,A Secure Image Encryption Algorithm Based on Rubik's Cube Principle, Hindawi Publishing Corporation,2012*

*(McAndrew,2012) McAndrew, Alasdair - Introduction to Cryptography with Open-Source Software, Publisher CRC Press ,2012.*

*(Menezes, 1996) A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography, Publisher CRC Press, 1996.*

*(N. Sethi,2013) Nidhi Sethi, Comparative Image Encryption Method Analysis Using New Transformed - Mapped Technique,Conference on Advances in Communication and Control Systems 2013*

*(O. Manz,2022) Olaf Manz, A compact introduction to cryptography, Publisher Springer 2022.*

*(O.Hawkins,2016)Olivia Hawkins, An Introduction to Cryptography, Network Associates, Inc,2016*

*(R. Mollin,2003) Richard Mollin, RSA and public-key cryptography, Publisher CRC Press,2003.*

*(Rijmenants,2004) Dirk Rijmenants, Cipher Machines & Cryptology, 2004 – 2023.*

*(S. Montabone,2010) Sebastian Montabone, Beginning Digital Image Processing, Apress,2010.*

*(S. Padhye,2018) Sahadeo Padhye, Rajeev A. Sahu, Vishal Saraswat, Introduction to Cryptography, Publisher CRC Press, 2018.*

*(S. Robertson,2020) Stephen Robertson, BC, Before Computers: On Information Technology from Writing to the Age of Digital Data, Open Book Publishers, 2020.*

*(S.Rubinstein, 2010) Introduction to Modern Cryptography, Publisher CRC Press,2010.*

*(S.E.Naffouti,A.Kricha, 2022)A sophisticated and provably grayscale image watermarking system using DWT-SVD domain,S.E.Naffouti,A.Kricha, 2022*

*(S.Sommraj,2015)Shrija Somaraj, Performance and Security Analysis for Image Encryption using Key Image, Indian Journal of Science and Technology,2015*

*(T.Denis, 2007)Tom St Denis, Cryptography for Developers 1st Edition, Publisher Syngress, 2007.*

*(W. Stallings,2010) William Stallings, Cryptography and Network Security Principles and Practice, United States Edition, Pearson,2010.*

*(W.Pratt, 2023)William K. Pratt, Introduction to Digital Image Processing, CRC Press,2023.*

*(W.Stallings, 2005)William Stallings, The RC4 Stream Encryption Algorithm,2005*

## *Webography*

*https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-image-enhancement/[Accessed on June 17, 2023].*

*https://www.tutorialspoint.com/ebook/cryptography-tutorial/index.asp [Accessed on May 04, 2023].*

*(M.Prokop, 2004) https://michael-prokop.at/blog/2004/10/19/paper-stream-ciphers-book-computer-security/[Accessed on May 18, 2023].*

*Maha Alwuthaynani, Classical Encryption Techniques- https://shms-prod.s3.amazonaws.com/media/editor/144716/CS-Lecture3.pdf[Accessed on June 1, 2023].*

*Professor Vipul Goyal, Introduction to Cryptography -https://www.cs.cmu.edu/~goyal/15356/ [Accessed on June 1, 2023].*