الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Master's Thesis

Specialty : Camputer Security And Cryptogrphy

Theme

Deep learning-based cryptanalysis of lightweight block ciphers

Presented by :

- Hocini Chaima
- Lagari Fatima Zohra

Led by :

- Dr. Benamara Djillali

Class Of 2022-2023

Acknowledgements

First of all we thank Allah who guide us to finish this work.

For our sincere gratitude to ***Dr.Benamara Djillali*** ,we would like to thank all the people who have contributed to the success of our project and who have helped during the writing of this report.

We also thank all the juries for their efforts and their care given to our work, To the teachers of our university and IT department.

Finally, we would like to thank all the people who did advise and proofread during the writing of this end-of-study project report: my family, our friends, etc.

Thank's everyone.

Dedication

To my dear parents **Mohamed Hocini** and **Om El khir Hamidi** and Grande Father **Hamidi Bouchrit**, for all their sacrifices, their love, their tenderness, their support and their prayers throughout my studies.

- To my dear sister **"Haifaa".**
- To my dear brothers **"Salah and Isak".**
- MY Friend "BMA"
- To my adorable buddy and roommate **"Fatima Lagari".**

for their constant encouragement, moral support To all my family for their support throughout my university career.
May this work be the fulfillment of your so-called wishes, and the result of your unfailing support.
Thank you for always being there for me.

Chaima Hocini

Dedication

I dedicate this thesis to my father and my queen mather **REGAGBA Mabrouka** for ther support and encouragement during my academic journey. Their love and belief in me and what driving me behind my 18 th studing year .....my greatset acheivments

- to my little piece "Amjed"
- to sweety sisters "asmaa" ,"hanaa", "selsabil" , "khadidja " and "arwa"
- to my brothers "mohamed abd-allah" , "moussab abd-elrahman"
- To my aunt "Fatima Ragagba".
- To my adorable buddy and roommate "Chaima Hocini".
- to my dear friend "Issam eddine Kacimi"
  I extend my deepest gratitude to my colleague ilyas bendjilali and for all who has helped me to finished this work.

- To everyone who loves me.

Fatima LAGARI

Contents

List of Figures

Abréviation

- **AI:** Artificial Intelligence.
- **ML:** Machine Learning.
- **LSTM:** Long Short-Term Memory .
- **GRU:** Gated Recurrent Units.
- **AE:** AutoEncoder.
- **DBN:** Deep Belief Networks .
- **GAN:** Generative Adversarial Network.
- **DL:** Deep Learning
- **ANN:** Artificial Neural Network.
- **CNN:** Convolutional Neural Network.
- **RNN:**A recurrent neural network.
- **CBC:** cipher block chaining .
- **CFB:** cipher block chaining .
- **CBC:** cipher block chaining .
- **P:** Plaintext.
- **C:** Ciphertext.
- **K:** Key.

General Introduction

There are various danger for using computer , computer networks and the biggest networks of them all the Internet .What can happen if we don't set up the right security policies, framework and technology implementations ?

The principle of discuss this help us identify and determining the security thears and possible solutions to tackle them .Since the electronic document and messages are now becoming equivalent to the paper document ,and according the importance of date, the protection of this informations turn into real need and felt like never before .People realized that data on computers is an extremely important aspect of modern life.

The science of protecting communications from outside observers is known as cryptography. The original communication, or plaintext, is transformed into cipher text, or unintelligible text, via encryption techniques.

The user can view the communication since the key enables them to decode it. It is also examined how strong an encryption's randomness is, which makes it more difficult for someone to guess the algorithm's input or key. We can increase our privacy by using cryptography to create connections that are more secure and reliable. Because of improvements in cryptography, only authorized users should be able to access encrypted files, folders, or network connections.

However, the Cryptanalysis is the study of decrypting communications created by cryptography in order to reveal their hidden meaning. It is learning how cryptosystems and ciphers they operate and developing methods for discover weakness in or breaking them.

General Introduction

The goal of cryptanalysis is to determine the cryptosystem's secret key. There are brute-force attack, differential attack, linear attack, and chosen plaintext attack .with the progress of artificial intelligence , cryptanalysis based deep learning become a new obsession and been a subject of research. There are studies that demonstrate known-plaintext attacks against simple block ciphers, including S-DES,SPEAK,PRESENT

0.1 Objectif

In this project We suggest a new cryptanalysis technique based on the art of deep learning technology. To Implementing it for breaking PRESENT lightweight block cipher the symmetric key encryption algorithm using linear cryptanalysis.
This cryptosystem is designed for resource restricted applications such as RFID (Radio Frequency IDentification) and WSN (Wireless sensor networks) .The lightweight cryptography community has paid close attention to because of the powerful hardware performance and security.

General Introduction

Chapter description:

Chapter one:

Introduces Machine learning and her relation with deep learning and explanation of them , including its background, the neural networks that inspired it, how it works, and the many deep learning models..

Chapter two:

The second chapter is an overview for cryptography science and a summary about the goal of cryptanalysis ,their details , his types and methods.

Chapter three:

in this chapter we Explain our contribution for our Key recovery with linear cryptanalysis using Recurrent Neural Network, how it works and the implementation experiment of our model and evaluation of performance.

Deep learning

1 Deep learning

1.1 Introduction

Artificial intelligence (AI) has gained popularity over the past ten years,both inside and outside of the scientific community.

Machine learning (ML), deep learning (DL), and AI have all been the subject of many publications in both technology and non-technology-based journals.In 1956,the idea that computers might be able to think and reason become an obsession,when a group of computer scientists asserted that "any fact of learning or any other attribute of intelligence in principle, be so thoroughly defined that a machine be made to replicate it." This idea was referred as "artificial ntelligence" by them.

AI is based on a learning process in order to reproduce part of human intelligence through an application, a system or a process. Facial recognition, visual perception and more are examples of artificial intelligence systems.

Machine Learning (ML) is a subfield of AI that uses artificial neural networks (ANN) to mimic the way humans make decisions.

ML allows computers to develop learning models on their own, without any programming, from large datasets. The layer immediately below is occupied by Deep Learning (DL) is one of the many machine learning approaches that have been extremely effective in the last years, To understand how these concepts relate to one another in the following figure:[1]

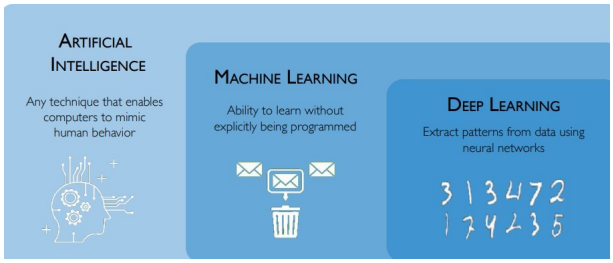

Figure 1: The relationship between AI and ML and Deep Learning[1].

1.2 Artificial intelligence :

Following World War II, several individuals began working on intelligent devices on their own. Perhaps the first was the English mathematician Alan Turing.
In 1947, he presented a talk about it. He may have also been the first to realize that programmingcomputers rather than creating machines was the ideal way to study AI.
Several academics were studying AI by the late 1950s, and the majority of them were using computer programming as the foundation for their research.
AI is the science and engineering of making intelligent machines,especially intelligent computer programs lthough it is connected to the same aim of utilizing computers to comprehend human intellect, AI should not be limited to techniques that can be seen physiologically.
The intelligence is computational component of being able to accomplish things in the world. People, many animals, and some machines all exhibit intelligence in various forms and to varying degrees.
AI can simulate human intelligence at times, but not always or even typically. On the one hand, by studying other individuals or even simply our own processes, we may pick up some tips on how to make machines solve issues. Nevertheless, rather than studying people or animals, the majority of AI research focuses on the challenges that the outside world poses to intelligence. Researchers in AI are allowed to employ techniques that haven't been tested on humans or that need a lot more computational power than humans are capable of [5].

1.2.1 Applications of AI

uses for AI:

1. **playing games :**
   For a few hundred dollars, you can get devices that can play chess at the master level.

Although they have some Intelligence, they mostly use brute force computation—looking at millions of positions— to play effectively against humans It takes the ability to look at 200 million spots per second in order to defeat a world champion using brute force and well-known trustworthy algorithms

2. **speech recognition :**
   Computer speech recognition became useful for a few specific uses in the 1990s. In order to replace its keyboard tree for flight information, United Airlines has devised a system that uses speech recognition for flight numbers and city names. That is quite practical. Yet, despite the fact that certain computers may be programmed via voice, most users still find the keyboard and mouse to be more practical.

3. **comprehension of natural language :**
   It is not sufficient to enter a list of words into a computer. Parsing phrases is also insufficient. It is necessary to provide the computer knowledge of the subject matter the text is about, and at the moment this is only feasible for a very small number of subjects.

4. **heuristic classification :**
   Put some information into one of a predetermined set of categories utilizing many sources of information is one of the most practical types of expert system given the current understanding of AI. example giving advice on whether to approve a planned credit card purchase . There is information accessible on the credit card holder, his payment history, the item he is purchasing, and the business from which he is purchasing it (for example, whether this business has a history of credit card fraud).

1.3 Machine learning

Machine learning is a research field of computer science t involves techniques for figuring out and using systems and algorithms. Computers have the ability to learn, and this discipline is typically tied to artificial intelligence, specifically computational intelligence. Computer intelligence is a data analysis method that automatically creates analysis models. In other words, allow the computer to develop concepts, evaluate, make decisions and plan future choices.

The following set of data is necessary for the entire learning process:

- **Dataset for training:** this is the knowledge base used to train, our learning algorithm, during this phase, the parameters of the model can be tuned (adjusted) according to the performance obtained.
- **Dataset for testing:** this is used just to gauge how well the model performs . Mathematics from probability theory and information theory are used in learning theory. This enables you to evaluate the superiority of one approach over another.

Three different categories of machine learning algorithms exist:

1.3.1 supervised Learning:

The most popular kind of machine learning nowadays is supervised learning. machine learning models are trained with labeled data sets, which allow the models to learn and grow more accurate over time. The reason this type of machine learning is called "supervised" learning is because you feed the algorithm information to aid in learning while it is being "supervised." The remainder of the information you supply is utilized as input features, and the output you give the system is labeled data.

For example, if you were trying to learn about the relationships between loan defaults and borrower information, you might provide the machine with 500 cases of customers who defaulted on their loans and another 500 who didn't. The labeled data "supervises" the machine to figure out the information you're looking for.

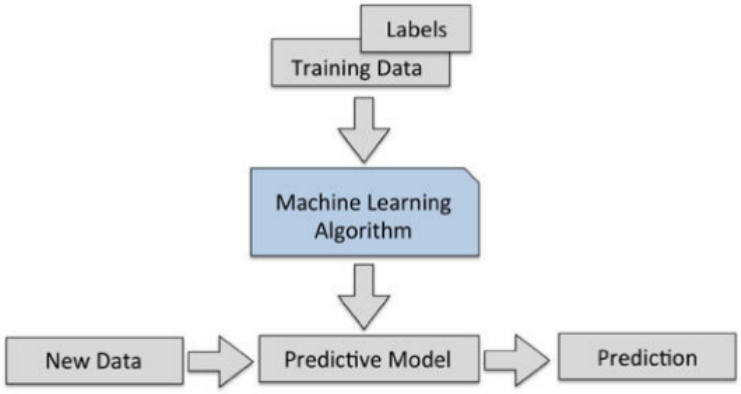

Figure 2: A resume of supervised learning category.[25]

1.3.2 Unsupervised learning:

While supervised learning requires users to help the machine learn, unsupervised learning doesn't use the same labeled training sets and data. Instead, the machine looks for less obvious patterns in the data. This machine learning type is very helpful when you need to identify patterns and use data to make decisions. Common algorithms used in unsupervised learning include Hidden Markov models, k-means, hierarchical clustering, and Gaussian mixture models.

Using the example from supervised learning, let's say you didn't know which customers did or didn't default on loans. Instead, you'd provide the machine with borrower information and it would look for patterns between borrowers before grouping them into several clusters.

This type of machine learning is widely used to create predictive models. Common applications also include clustering, which creates a model that groups objects together based on specific properties, and association, which identifies the rules existing between the clusters.

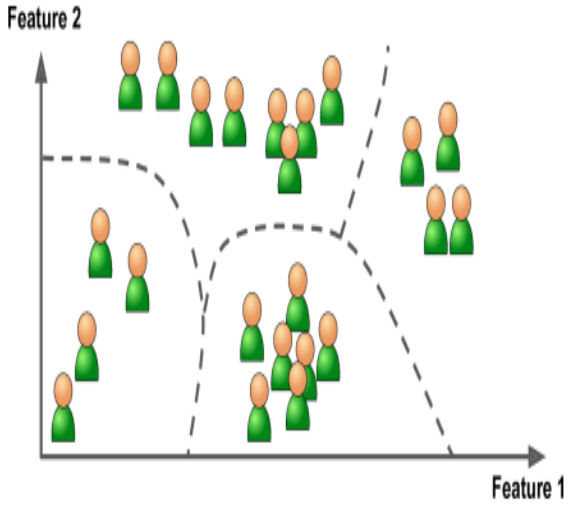

Figure 3: Clustering [2]

1.3.3 Reinforcement learning:

It is the closest machine learning type to how humans learn. By interacting with its environment and receiving rewards, either positive or negative, the algorithm or agent being used learns. Typical algorithms include temporal difference, deep adversarial networks, and Q-learning. By establishing a reward system, machine learning teaches machines to choose the best course of action through trial and error. By letting the machine know when it made the right choices, reinforcement learning can train models to play games or train autonomous vehicles to drive. Over time, the machine will learn what actions to take by letting it know when it made the right choices. Reinforcement learning can train models to play games or train autonomous vehicles to drive. Over time, the machine will learn what actions to take.

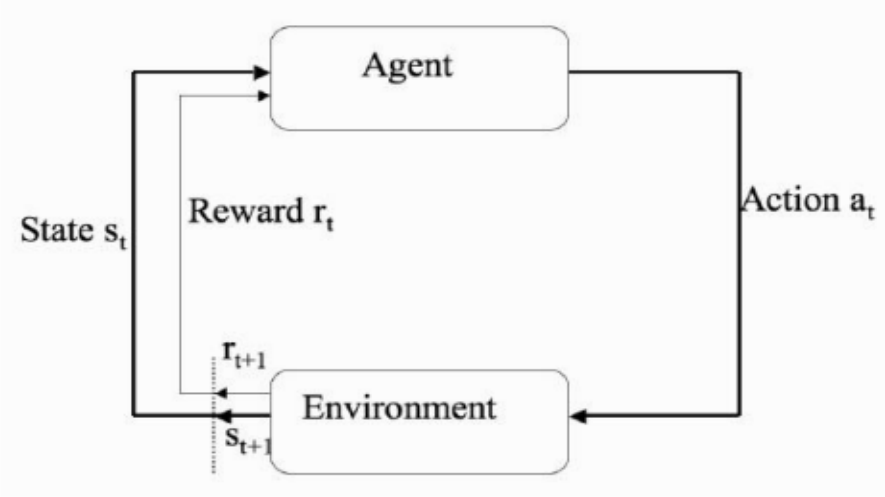

Figure 4: The Reinforcement Learning cycle[10].

1.4 Deep learning

The human brain is the incredible organ that dictates the signals received from sound, sight, smell,touch, and taste.

Brain stores emotions, experiences, memories, and even dreams. The brain takes decisions, solves many problems that even the powerful supercomputers lack (Kuhn, 1998)On the basis of this, scientists have fantasized about building brain-like intelligent devices. Subsequent researchers developed self-driving automobiles, automatic illness detection microscopes, and robots to aid humans in their daily chores. These innovations still needed human assistance with some computing tasks. In order to handle more complicated issues faster than the human brain, researchers are working to create a machine that can learn on its own. These prerequisites open the door to deep learning, the most active branch of artificial machine intelligence.

Artificial neural networks, a subfield of deep learning, are algorithms that were motivated by the structure and operation of the brain (In practice all DL algorithms are neural networks).

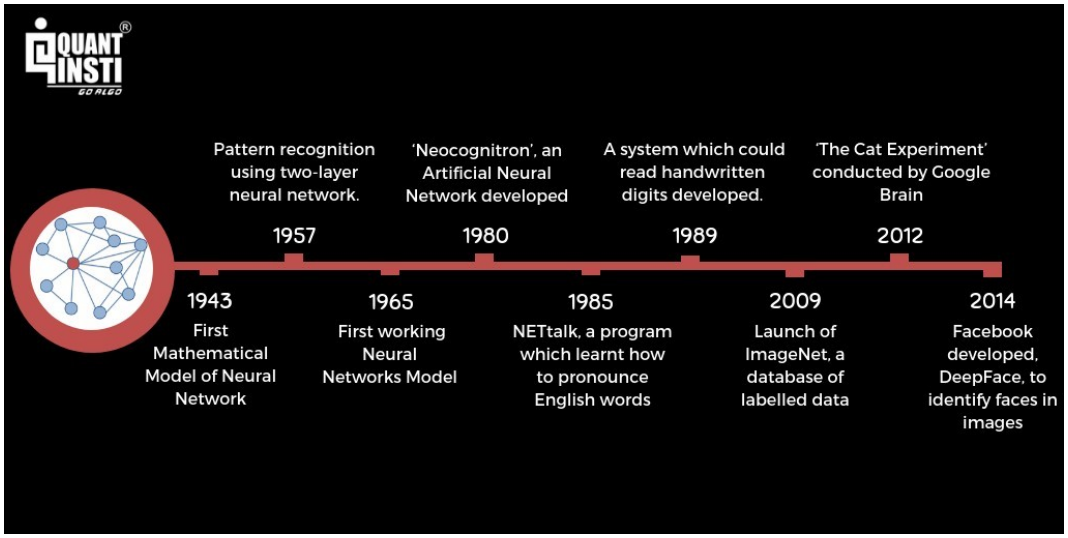

Figure 5: Deep Learning Timeline[8].

1.4.1 Neural Networks:

The basic unit of the human brain is the neurons. Very small portions of the brain, about the size of wheat, have over 10,000 neurons with more than 6,000 connections with other neurons [7] The neurons in the brain capture the information the brain receives, convey it to other neurons for processing, and then send the finished product to other cells. In figure 6, it is shown. The neurons' dendrites are an antenna-like structure that serves as the input receiver.

The inputs are divided into strengthened and weakened categories based on how frequently they are used. The degree whereby the input is related to the neuron's output is estimated by the connection strength. The connection strength is used to weight the input signals, which are then added collectively in the cell body. The estimated total manifests as a new signal that travels up the cell's axon to the target neurons.

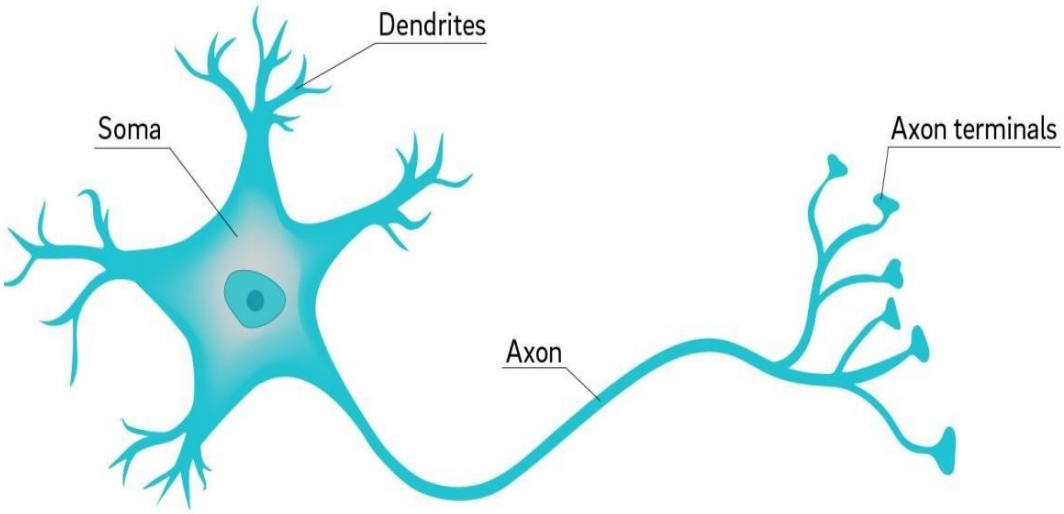

Figure 6: biological Neuron[9].

In 1943, Warren S. McCulloch and Walter H. Pitts [6] concentrated on the functional understanding of the neurons exist in the human brain and created a computer-based artificial model as shown in figure: As in the biological neurons, the artificial neuron re-

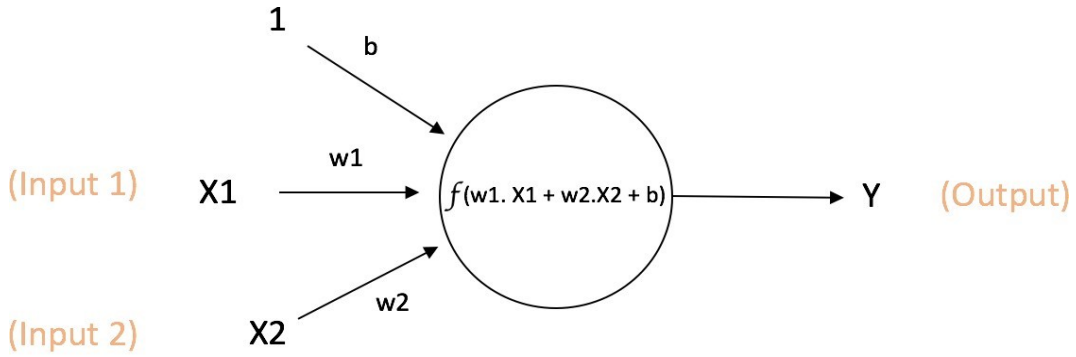

Figure 7: Neuron in an artificial neural net[10].

ceives inputs x1, x2, x3. . . .xn, and respectively input is multiplied by a particular weights w1, w2, w3,. . . . . .wn and the calculated sum is considered to make the logit of the neuron

$$Z = \sum_{\mathrm{i}=0}^{n} w_{\mathrm{i}} x_{\mathrm{i}} \tag{1}$$

Some logit may include a constant value called the bias.Finally, the logit is passed through a function f to make the desired output

$$y = f(z). \tag{2}$$

1.4.2 Activation functions

The activation function is an important feature of the neural network, to decided being activated or not . It calculates the weighted amount of entries and adds the bias. It is a non-linear transformation of the input value.

After processing, this output is sent to the next layer. Without an activation function, a network of neurons is just a linear regression model.

1.4.3 Types of Activation Functions [11]:

- **The sigmoid function:**These are the most widely used functions in the creation of artificial neural networks. Take a real input and reduce it between 0 and 1.

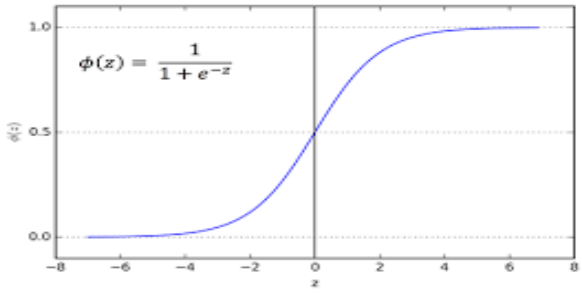

Figure 8: Sigmoid activation function[10].

- **The ReLu Function:**This is a very simple activation function. Suppose the input is the value X and if X is positive the output will be X else 0, The ReLu function (Rectified Linear Unit)is:

$$f(x) = \max(0, x) \tag{3}$$

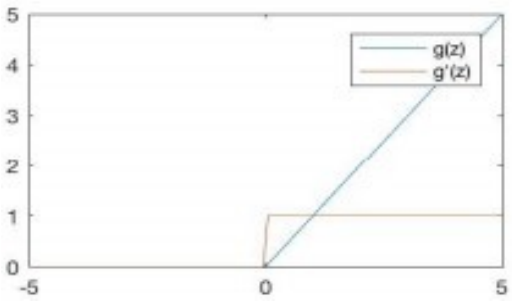

Figure 9: Graphic representation of the ReLu function[10].

- **Soft max function :**It realizes a normalized exponential (i.e. the sum outputs totals 1). In combination with the cross- entropy error function, it allows modification of multilayer perceptron networks for the estimation of class probabilities.

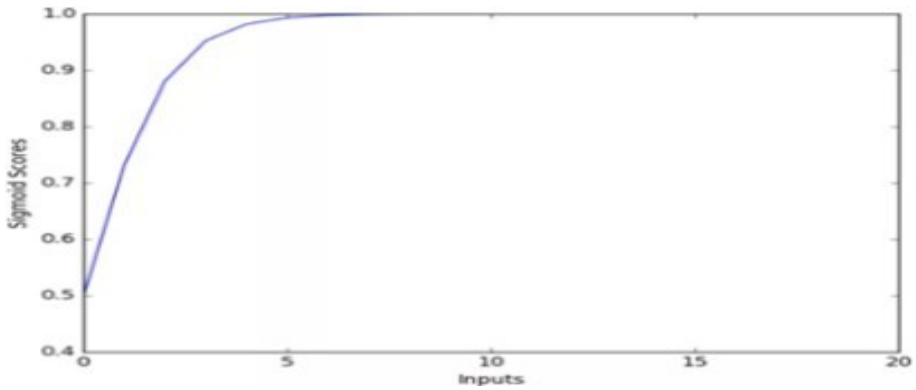

Figure 10: Representation of the softmax function graph[10].

- **The Hyperbolic Tangent function:**takes a real value input and reduces it to a value in [-1, 1].

$$F'(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{4}$$

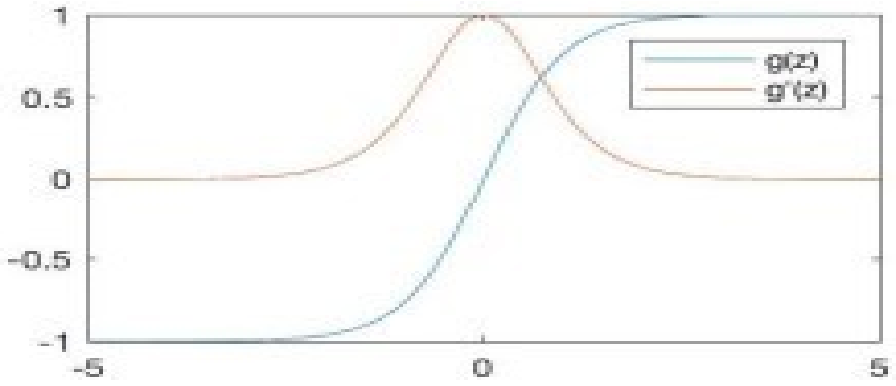

Figure 11: Hyperbolic Tangent function activation[10].

1.4.4 Types of Deep Learning Networks [11] :

Deep architectures come in a lot of different forms. The majority of them are descended from some unique parent architectures. As not all designs are assessed using the same set of data, it is not always viable to compare their performance. The subject of deep learning is rapidly expanding, and new architectures, versions, or techniques are developed every week.

- **Convolutional Neural Network (CNN):**

  CNN is a famous tool in recent years, particularly in image processing and are stirred by the organization of the cat's visual cortex [1]. The local connectivity is imposed on the raw data on CNN. For example, more significant features are extracted by perceiving the image as a group of local pixel patches rather considering 50 by 50 image as individual 2500 unrelated pixels.

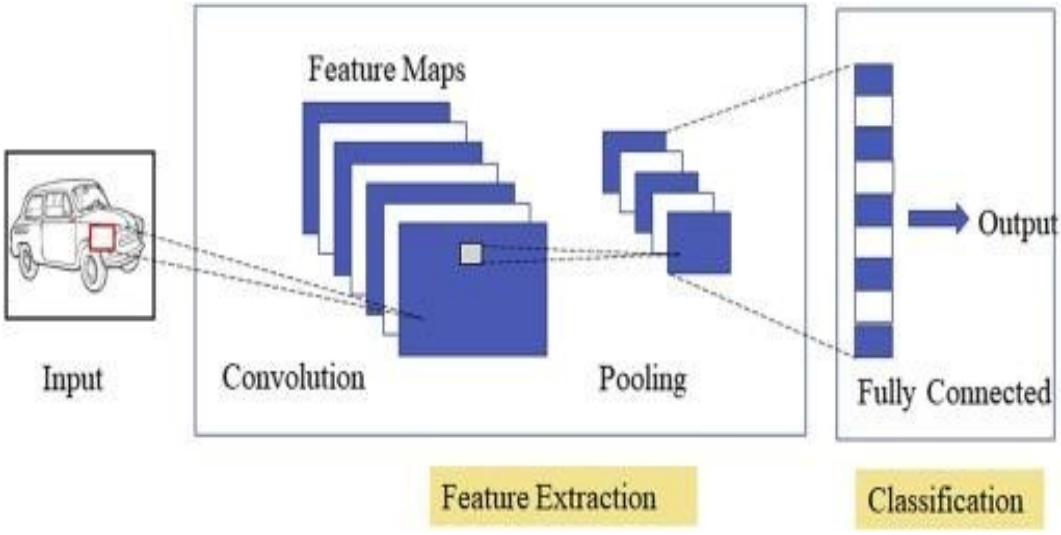

Figure 12: Standard architecture of a convolutional neural network[10].

- **Recurrent Neural Network (RNN)**

  RNN is a logical choice if the input data is ordered sequentially (e.g, natural language or time series data) RNNs is capable of handling long-range temporal dependencies. In RNN, hidden state ht is updated based on the triggering of current input xt at a time t and the previously hidden state t1. Consequently, the final hidden state contains complete information from all of its elements after processing an entire sequence. RNN include :

  1. Long Short-Term Memory **(LSTM)**
  2. Gated Recurrent Units **(GRU)**

  The symbolic representation of RNN is shown in Figure 10 , with its equivalent extended representation for instance 3-input units, 3-hidden units, and an output. The input time step is united with the present hidden state that depends on the previous hidden state.

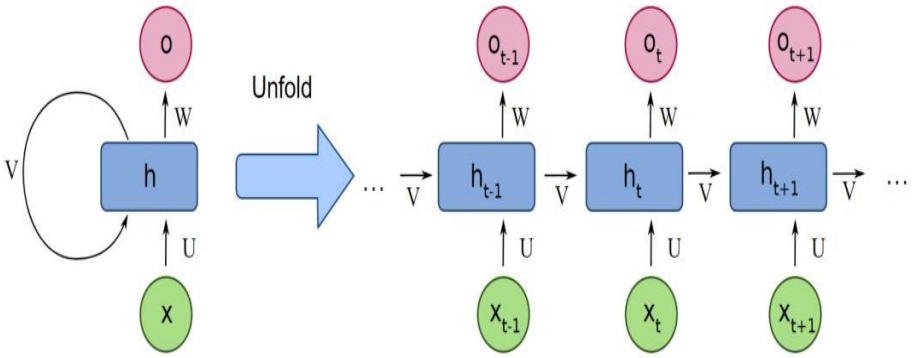

Figure 13: RNN with extended representation[10].

- **Feed-forward Neural Networks (FNN):** It is the highly interconnected network of artificial neurons inspired by the human nervous system, working in union to perform a given task and, in the end, gives the decision based on weights and biases for complex input data.

Type of RNN[35]:

1. **One-to-One:**
   One-to-One RNNs are the most basic RNN types because they only support a single input and output. It operates like a conventional neural network and has fixed input and output sizes. Image Classification contains the One-to-One application.

2. **One to Many:**
   A type of RNN known as one-to-many produces multiple outputs from a single input. It accepts a fixed input size and outputs a series of data. Applications for it can be found in image captioning and music generation.

3. **many-to-One:**
   When a single output from numerous input units or a series of them is required, many-to-one is used. For a fixed output to be displayed, a series of inputs are required. A typical illustration of this kind of recurrent neural network is sentiment analysis.

4. **Many-to-Many:**
   A sequence of output data is produced from a succession of input units using the many-to-many method.

   The next two subcategories for this kind of RNN are as follows:

- **Equal Unit Size:** In this instance, there are the same amounts of input and output units. Name-Entity Recognition is a typical application.
- **Disparate Unit Size:** In this instance, the number of units for the inputs and the outputs varies. Its use in machine translation is evident.

- **AutoEncoder (AE):**

  Autoencoder (AE) is the deep learning model that exemplifies the concept of unsupervised representation learning. Initially, it has pertained to supervised learning models once the labeled data was limited, but it is still remained to be useful for complete unsupervised learning such as phenotype discovery. In AE, the input is encoded into a lower-dimensional space z and it is decoded further by reconstructing x- of the corresponding input x. Hence, the encoding and decoding process of an encoder are respectively given in equation with a single hidden layer. And the encoding and decoding weights are represented as W and W0 and the reconstruction error is minimized. Z is a reliable encoded representation.

  z=$\sigma(\mathrm{W}x + b)$

  $\bar{x} = \sigma\left(\mathrm{W}'z + b'\right)$

  As soon as an AE is well trained then a single input is fed in the network and the innermost hidden layer activated to serve as input for the encoded representation.

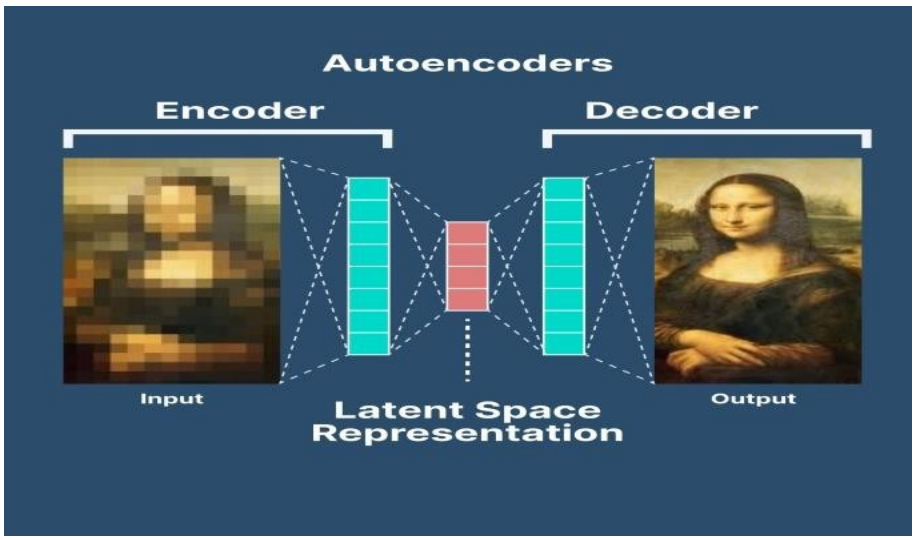

Figure 14: Autoencoder example[10].

- **Deep Belief Networks (DBN):**It is a generative model made up of several RBM and autoencoder layers. With supervision, it may be trained to behave as feature detectors and probabilistically limit its inputs after learning unsupervised.

- **Generative Adversarial Network (GAN)** An analogy from the real world: Let's consider the real-world relationship between a money counterfeiting criminal and the police. Let's enumerate the objective of the criminal and the police in terms of money:

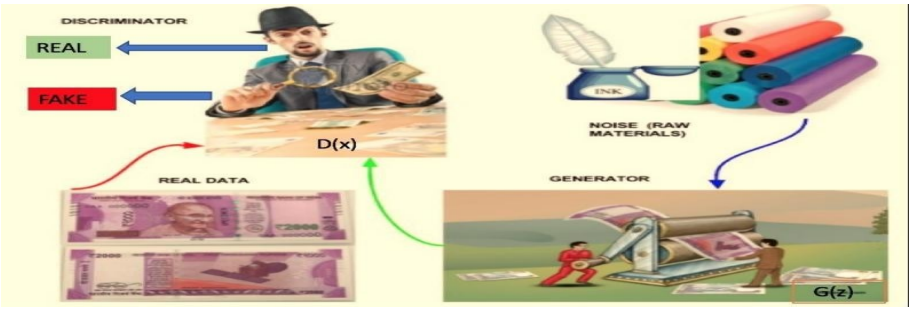

Figure 15: GAN real world analogy[10].

• To become a successful money counterfeiter, the criminal needs to fool the police so that the police can't tell the difference between the counterfeit/fake money and real money
• As a paragon of justice, the police want to detect fake money as effectively as possible This can be modeled as a minimax game in game theory.
This phenomenon is called adversarial process. GAN, introduced by Ian Goodfellow in 2014 at arXiv: 1406.2661, is a special case of an adversarial process where two neural networks compete against each other. The first network generates data and the second network tries to find the difference between the real data and the fake data generated by the first network. The second network will output a scalar [0, 1], which represents a probability of real data.

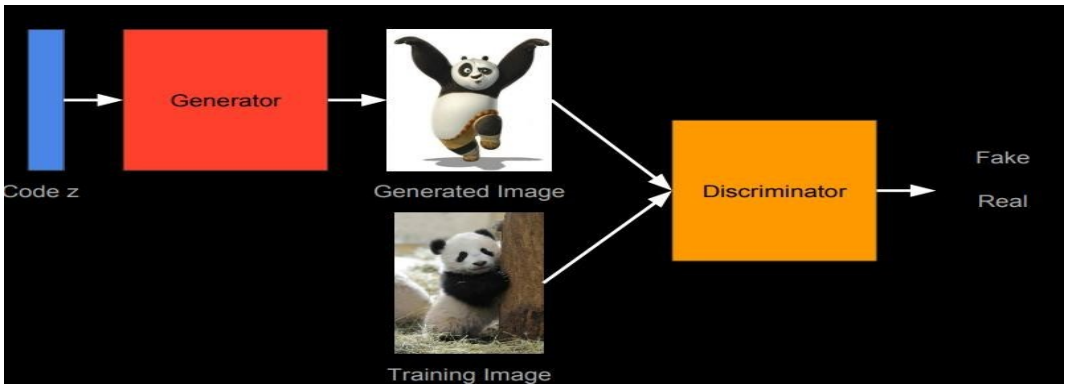

Figure 16: exemple of GAN[10].

1.4.5 Training Neural Networks :

Loss Optimization:
We want to find the network weights that achieve the lowest loss

$$\mathrm{w}^* = argmin_{\mathrm{w}} \frac{1}{n} \Gamma_{\mathrm{i}=1}^{n} f\left(\mathrm{f}(\mathrm{x}(\mathrm{i}) \div \mathrm{w}), \mathrm{v}^{(\mathrm{i})}\right)$$
$$\mathrm{w}^* = argmin_{w} J(w)$$

At the same time, Bernard Widrow and his student Ted Hoff introduced a slightly modified learning rule called LMS. Instead of correcting for classification errors, they proposed using the squared error as a measure of quality.

And minimize the mean square error (MSE)
L(w)=1 / N $\sum_{i=1}^{N} l\,(y_i), f\,(x_i))$

On the objects of the training set. For this purpose, they suggested using a gradient descent method.

- **gradient descent algorithms:**
  There are three main types of variants of the gradient descent algorithm :
- **Batch gradient descent:**
  This is classic gradient descent, we calculate the gradient of the cost function at the parameters for the entire learning set
- **Stochastic gradient descent:**
  Each instance of data set xi and label yi is updated with parameters using SGD
- **(stochastic gradient descent)**.

W=W-$\alpha\nabla_W L\,(x^i, y^i W)$

This approach is quicker, but excessively frequent parameter changes lead to oscillations in the objective function, which can lead to the discovery of possibly better local minima while also making convergence more challenging.

Mini-batch gradient descent:

This technique adjusts the settings for each minigroup of n samples using the best features of both techniques.

W=W-$\alpha\nabla_W L\,(x^{i;i+n}, y^{i;i+n}; W)$

Using this approach, parameter updates' variance is decreased, resulting in a more stable convergence. Minigroups typically have 32 to 256 instances. It is a preferred technique for neural network training.

Classical gradient descent presents various difficulties that need to be overcome and does not always give satisfactory convergence:

- The choice of the learning rate is difficult, if it is too small it can lead to a sconvergence too slow, if it is too large it can cause oscillations in the cost function or even not converge at all
- The same learning rate is applied to all parameters. We may want to apply a larger update to settings for features that occur less frequently if the characteristics of the available observations have a different frequency.
- Another problem of minimizing non-convex cost functions particularly prevalent in neural networks is to avoid becoming stuck in local optima

1.4.6 Gradient descent optimization algorithms:

- **Adagrad:** The gradient-based optimization algorithm Adagrad accomplishes this by tailoring the learning rate to the parameters, performing smaller updates (i.e., low learning rates) for parameters associated with frequently occurring features and larger updates (i.e., high learning rates) for parameters associated with infrequent features. This makes it a good choice for handling sparse data. Adagrad, according to research, significantly increased SGD's resilience. Google employed it to train massive neural networks that, among other things, learnt to detect cats in Youtube videos . Also, Pennington trained GloVe word embeddings using Adagrad since uncommon words need significantly greater updates than frequent ones. Previous, we changed each parameter individually because they all utilized the same learning rate. different learning rate is used by Adagrad for each parameter i at each time step t.
- **RMSprop:** Geoff Hinton has devised the adjustable learning rate approach known as RMSprop (Also known as Father of Deep Science).
  Instead of accumulating all the squares of the previous gradients, we restrict the accumulated gradient window to a fixed size w .Instead of storing the w squares of the previous gradients, an exponential moving average of the squares is applied.

- **Adam optimizer:**

  Adaptive Moment Estimation (Adam)is Another technique that determines adaptive learning rates for each parameter.

  Adam preserves an exponentially decaying average of previous gradients mtmt, comparable to momentum, in addition to keeping an exponentially decaying average of past squared gradients v(t) like Adadelta and RMSprop. Adam acts like a heavy ball with friction, as opposed to momentum, which may be visualized as a ball rolling down a slope, and favours smooth minima in the error surface. In datasets with less variety, Adam performs best.

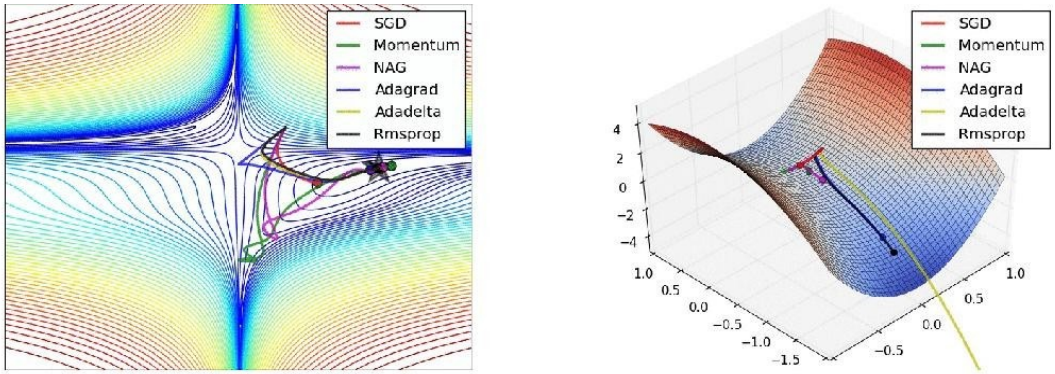

Figure 17: Comparison of Different Optimizers at Saddle point[10].

- **Computing Gradients: Backpropagation:**

  Backpropagation algorithms are used extensively to train feedforward neural networks They efficiently compute the gradient of the loss function with respect to the network weights. This technique removes the wasteful step of explicitly computing the gradient with regard to each individual weight.

  It enables the use of gradient methods, like gradient descent or stochastic gradient descent, to train multilayer networks and update weights to minimize loss. The difficulty of understanding exactly how changing weights and biases affect the overall behavior of an artificial

neural network.How does a small change in one weight (ex. w1) affect the final loss J(w)?

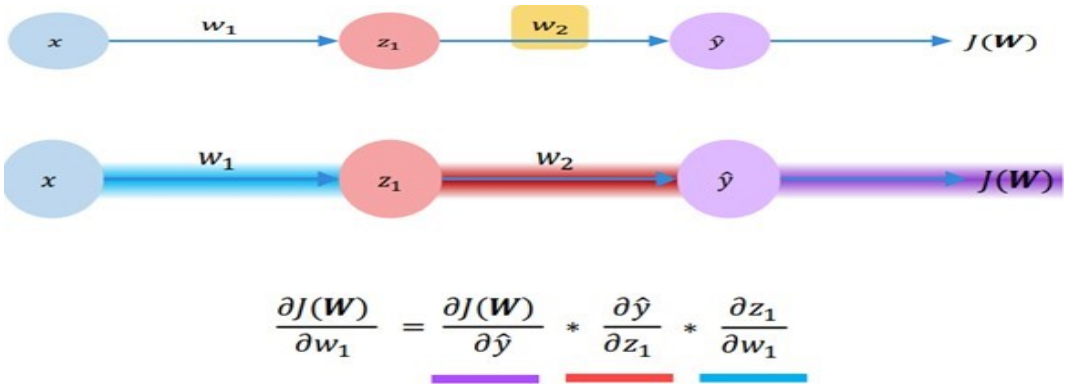

Figure 18: Backpropagation steps[10].

1.5 Conclusion

In this chapter We have discussed what is AI ML and DL (history and explained about his basics and distinguish between different methods and algorithms of training the neural networks) , we have discovor that creation of intelligent systems fall under the broad heading of AI, ML is a subset of AI that focuses on the models and techniques that let computers learn from and be more adept at using data and DL is a branch of ML that uses deep neural networks to represent and understand intricate patterns from data.

Chapter02

Cryptanalysis

2 Cryptanalysis

2.1 Introductions

As mentioned, the confidential exchange of information is critical in society from military orders to credit card numbers, for thousands of years, people have had data that must be protected from unwanted eyes. The science of securely transferring information is known as cryptology and is usually separated into two distinct yet related sciences: cryptography and cryptanalysis.

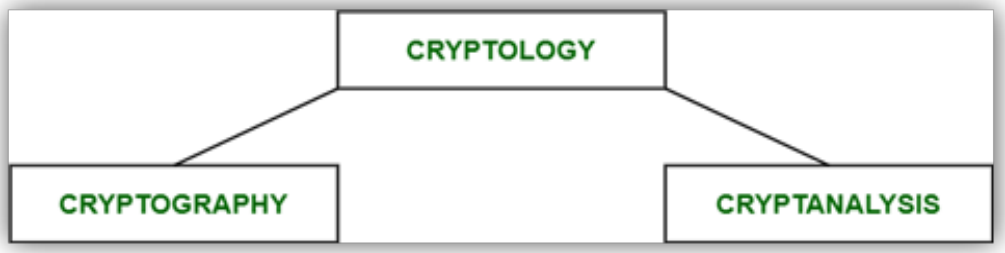

Figure 19: What does it contain cryptology [12].

Cryptography (Figure 19) 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.

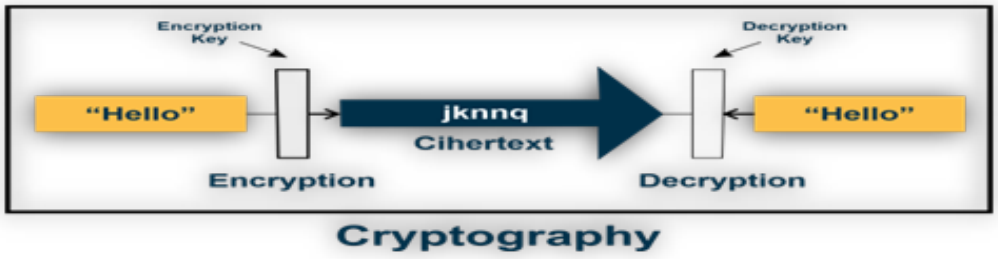

Figure 20: What does it contain cryptology [13].

With a third person (Eve1 or Mallory2) present, the area of cryptography in computer science and mathematics focuses on methods for secure communication between two parties (Alice Bob) (see Figure 13).
Based on techniques like encryption, decryption, signature, and the creation of pseudo-random numbers, among others le cryptanalysis is the science of deciphering secure com-

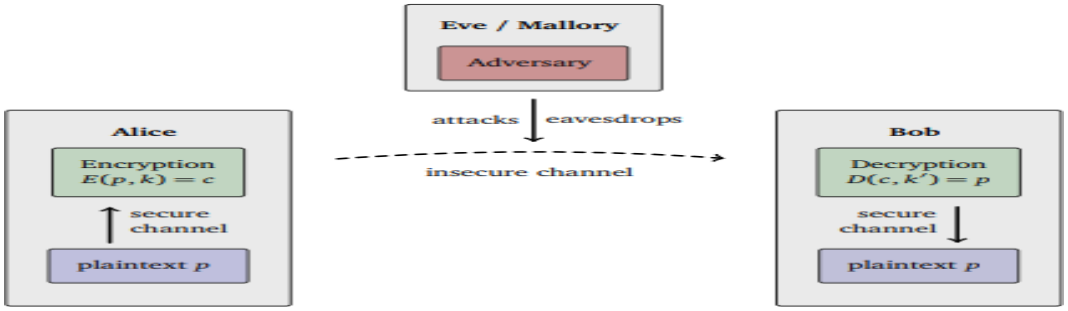

Figure 21: A basic idea for secure communication[14].

munication, cryptography is the science of protecting data. Traditional cryptanalysis requires a unique blend of analytical thinking, using mathematical tools, pattern recognition, patience, perseverance, and luck. Attackers are another name for cryptanalysts. Cryptology includes both cryptanalysis and cryptography. Cryptanalysis is the study of cryptographic system compromise.

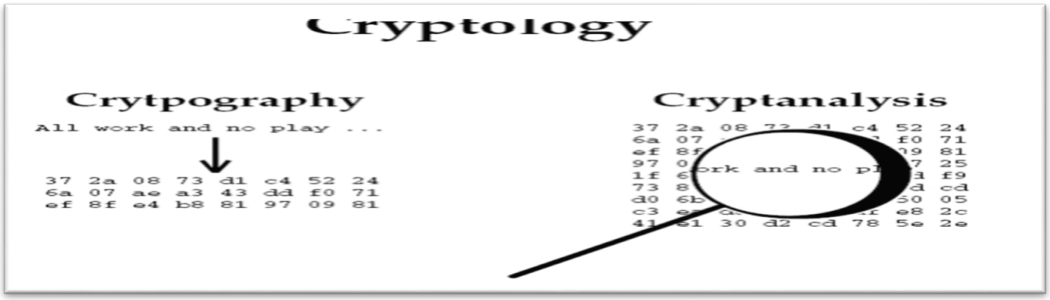

Figure 22: Illustration of the two fundamental areas of cryptology: cryptography, how to use ciphers to encrypt and decrypt information, and cryptanlaysis, how to break ciphers[5].

There is no text that describes recent developments in the cryptanalysis field. The last few hundred years have seen significant advancements in the field of cryptanalysis, which has, for the most part, been thoroughly researched and documented.
However, when we move into the 20th century, the documentation of cryptanalysis has come to a near standstill. Almost every book published on the topic of "cryptanalysis" is stuck nearly 100 years in the past, idling around the area of breaking some of the simplest ciphers, by today's standards[5]
The field itself has not stopped developing. On the contrary, it has been moving incredibly rapidly, especially in the past 30 years, with the rise of ever more powerful computers. While all of this research into cryptanalysis has been documented and presented at various conferences throughout the world, nobody had bothered to create a simple resource with which to learn cryptanalysis from scratch. Bruce Schneier stated that such a resource would not be worthwhile, because the field changes so much, and he has a point. But, the current roads on which cryptanalysis travels are built on the same foundations, and the amount of background material needed to understand current research or participate is becoming very large and complicated. Furthermore, the important papers are written by many different individuals with many diverse goals and audiences, which can make the papers difficult to understand. I must reiterate what Schneier says[5]
THOUGH: THERE IS ONLY ONE WAY TO BECOME A GOOD CRYPTANALYST — TO PRACTICE BREAKING CODES —

2.2 Types of encryption:

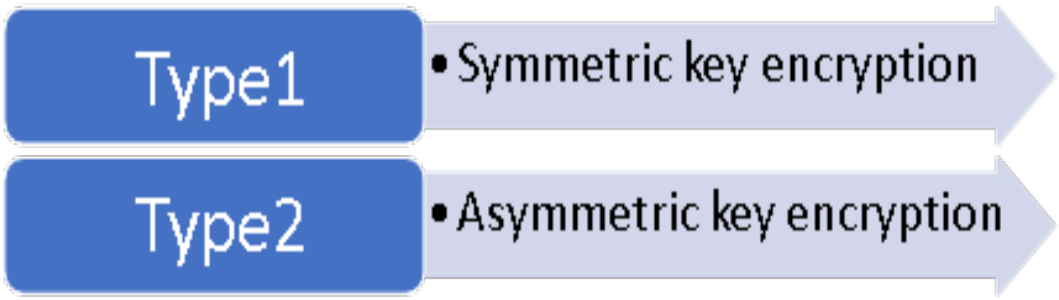

Figure 23: Types of encryption.

I will mention Symmetric key encryption because it has the type that we will work on :

2.2.1 Symmetric key encryption:

Private key cryptography, also called symmetric cryptography, has been used for several centuries. This is the most authentic approach to data encryption and mathematically the least problematic.

The key used to encrypt the data can be easily determined if one knows the key used to decrypt and vice versa. In most symmetric systems, the encryption key and the decryption key are one and the same key.

Consider a encryption scheme comprising two sets Ee:e appartenanat kand d appartenantwhich represent the set of encryption and decryption transformations respectively, where K is the space of the keys. m and c are the plaintext and the ciphertext respectively. The following image represents the scheme of a symmetric key encryption.

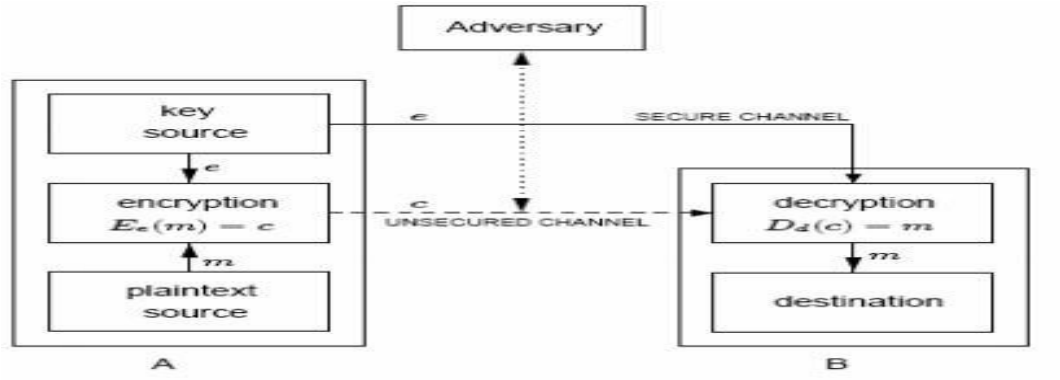

Figure 24: Principle of symmetric encryption[[17].

The main types of private key cryptosystems used today fall into two broad categories: stream cryptosystems and block cryptosystems.

- **Flow cryptosystems**

  In a stream cryptosystem, message encryption is done character by character or bit by bit, the key is generated randomly, its size is equal to the size of the message. The most illustrative example of this principle is the Vernam cipher. This algorithm is also called "One Time Pad" (disposable mask), i.e. the key is only used once.

- **Block cryptosystems**

  Block cipher is any (symmetric) encryption system in which the clear message is cut into blocks of a fixed size, and each of these blocks is encrypted. The length n of blocks and the size l of keys are two characteristics of block cipher systems.

  If the length of the message is not a multiple of the length of a block, it is completed:this is tamping or padding in English

  We care to talk about Block cryptosystems because all our studies will talk about it.

2.3 What Does Block Cipher Mean?

A block cipher is an encryption method that applies a deterministic algorithm along with a symmetric key to encrypt a block of text, rather than encrypting one bit at a time as in stream ciphers. For example, a common block cipher, AES, encrypts 128 bit blocks with a key of predetermined length: 128, 192, or 256 bits[18].

Block ciphers are pseudorandom permutation (PRP) families that operate on the fixed size block of bits.
PRPs are functions that cannot be differentiated from completely random permutations and thus, are considered reliable, until proven unreliable [18].
Block cipher modes of operation have been developed to eliminate the chance of encrypting identical blocks of text the same way, the ciphertext formed from the previous encrypted block is applied to the next block. A block of bits called an initialization vector (IV) is also used by modes of operation to ensure ciphertexts remain distinct even when the same plaintext message is encrypted a number of times.
Some of the various modes of operation for block ciphers include CBC (cipher block chaining), CFB (cipher feedback), CTR (counter), and GCM (Galois/Counter Mode), among others. Above is an example of CBC mode.
Where an IV is crossed with the initial plaintext block and the encryption algorithm is completed with a given key and the ciphertext is then outputted. This resultant cipher text is then used in place of the IV in subsequent plaintext blocks.
However,block cypher cryptanalysis has consistently drawn a lot of attention. Very recently, numerous cryptanalytic methods have been developed.

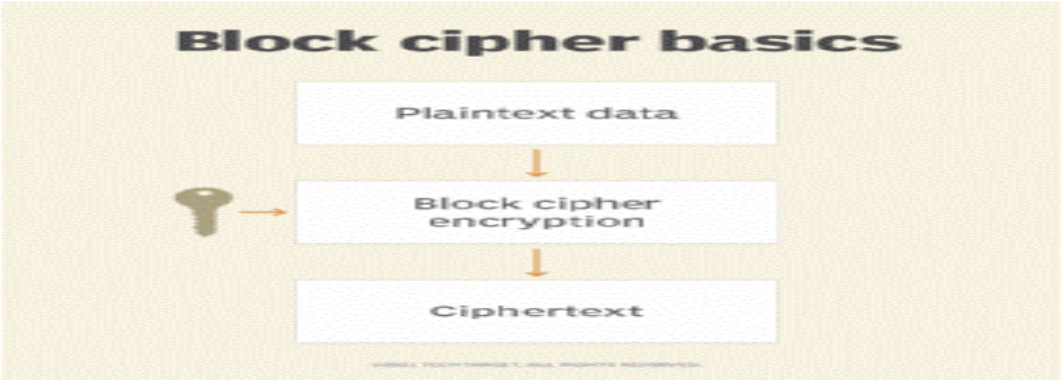

Figure 25: Block cipher basics [10].

2.3.1 Four block cipher modes are possible:

ECB, CBC, CFB and OFB. Their objectives are:

- **They concern only the block cipher.**
- **They must mask the identical clear blocks.**
- **Two identical messages encrypted with the same key do not give the same numbers.**
  In our project we will touch:

ECB mode (Electronic Code Book): This is the simplest mode,the message (M) is split into blocks(mi)greater or equal to 1 and each block is separately encrypted by c i=E(m i)

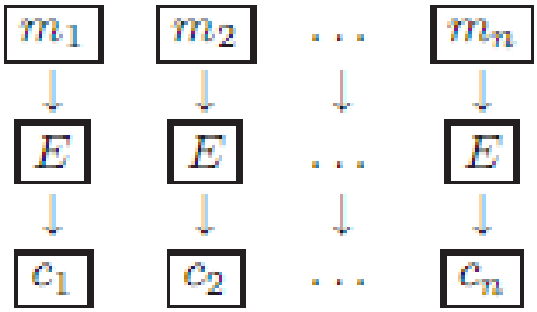

Figure 26: ECB mode diagram [17].

Where E=Ek depends on the secret key k and ci is the corresponding encrypted block. The problem of this mode is that two identical clear blocks always give the same encrypted block for a fixed key k. It therefore offers no security and is therefore not used. **The other modes will not be mentioned as a title only:**

- Cipher Block Chaining Mode(Le mode CBC)
- Cipher FeedBack Mode (Le mode CFB)
- Output FeedBack Mode(Le mode OFB)
- Counter-mode encryption Mode(Le mode CTR)

2.4 Cryptanalysis:

Cryptanalysis is the study of encrypted information with the aim of finding weaknesses (flaws), discovering the secret and decrypting the ciphertexts. Decryption is the art of finding the plain text without knowing the encryption key. Traditional cryptanalysis combines the application of mathematical tools with pattern finding and analytical resolution. Patience, determination and luck can be among the ingredients of a successful cryptanalyst. Cryptanalysts are also called hackers or hackers.

Cryptanalysis techniques can be summarized in five levels of attacks related to the data used:

1. **Brute-force attack:**The cryptanalyst tests all possible key combinations until clear text is acquired.

2. **Ciphertext-only attack:** The cryptanalyst only knows the message encrypted by the algorithm and tries to deduce the key or the plain text. The lack of information makes cryptanalysis more difficult.

3. **Known-plaintext attack:** The cryptanalyst possesses the text or parts of the plaintext and their encrypted correspondence.

4. **Chosen-plaintext attack:** The cryptanalyst can choose the plaintext, and he can produce the encrypted version of this text (he has access to the encryption machine) with the algorithm considered from then like a black box. Asymmetric encryption techniques are particularly susceptible to this type of attack.

5. **Chosen-ciphertext attack**: The cryptanalyst owns the ciphertext and can obtain the associated plaintext.

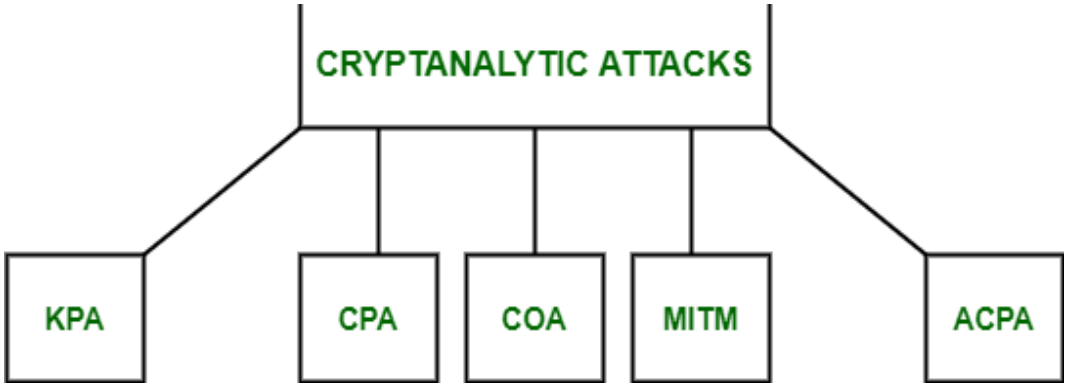

Figure 27: Five Types of Cryptanalytic Attacks [19].

To verify the security of a newly designed cryptosystem, a few elements are essential to analyze. Cryptanalysis algorithms belonging to the modes mentioned above have been designed and developed according to the robustness and the characteristics of the proposed encryption algorithms.
The following categories of cryptanalysis based on the algorithm of algebraic structures exist: a related-key attack, a meet-in-the-middle (MITM) assault, a differential cryptanalysis, a linear cryptanalysis, and a differential-linear cryptanalysis.

2.4.1 Different Forms of Cryptanalysis:

Cryptanalysis basically has three forms[19]:

- **Differential Cryptanalysis**
  Differential cryptanalysis is a sort of cryptanalysis that may be used to decrypt both block and stream ciphers, as well as cryptographic hash functions. In the widest sense, it is the study of how alterations in information intake might impact the following difference at the output. In the context of a block cipher, it refers to a collection of strategies for tracking differences network of transformations, finding where the cipher displays non-random behavior, and using such attributes to recover the secret key (cryptography key)[19].

- **Algebraic Cryptanalysis:**
  Given a particular cipher, algebraic cryptanalysis consists of two steps. First, one must convert the cipher and possibly some supplemental information (e.g. file formats) into a system of polynomial equations, usually over GF(2), but sometimes over other rings. Second, one must solve the system of equations and obtain from the solution the secret key of the cipher. This chapter deals with the first step only[20].

- **Linear Cryptanalysis:** Linear cryptanalysis is a general type of cryptanalysis based on discovering affine approximations to a cipher's action in cryptography. Block and stream ciphers have both been subjected to attacks. Linear cryptanalysis is one of the two most common attacks against block ciphers, with differential cryptanalysis being the other[19]

Certainly the best solution "Linear Cryptanalysis" we have dealt with is because they are more understanding than others.

Linear cryptanalysis together with differential cryptanalysis are the generally used attacks on block ciphers.

2.4.2 What is Linear cryptanalysis

To concentrate on a few different classes of these cyphers, even though there are many cryptanalytic strategies that may rely on a deep analysis of a cypher. Over a crucial group of more recent cryptanalysis techniques in this chapter called linear cryptanalysis. A cipher's diffusion and confusion should be as close to ideal as possible; otherwise, it would be simple to predict the output from the input without the key. There will always be some structural flaws, so no cypher can have truly perfect diffusion. The types of these flaws and how to exploit them result in the various attacks. This chapter describes and illustrates the linear cryptanalysis technique. We also assess how well the technique works with various cyphers.

Definition :

Linear cryptanalysis is a technique invented by Mitsubishi Electric researcher Mitsuru Matsui. It dates back to 1993 and was originally developed to break the DES symmetric encryption algorithm. This type of cryptanalysis is based on a concept prior to Matsui's discovery: probabilistic linear expressions. The latter were studied by Henri Gilbert and Anne Tardy-Corfdir as part of an attack on FEAL.

Linear cryptanalysis is more efficient than differential cryptanalysis, but less practical for the simple reason that it is assumed that the attacker does not have the box black symbolizing the encryption algorithm, and that he cannot submit his own texts.
Linear cryptanalysis consists of making a linear approximation of the encryption algorithm by simplifying it. By increasing the number of pairs available, the accuracy of the approximation and we can extract the key. All new encryption algorithms must ensure that they are resistant to this type of attack.
DES was not designed to prevent this kind of method, the substitution tables (S-Boxes) indeed present certain linear properties, whereas they were precisely planned for add non-linearity to DES.
It was then successfully applied to several algorithms such as LOKI, FEAL or a simplified version of Serpent. Newer algorithms like AES (Rijndael), IDEA, and many others are insensitive to a linear attack. The complexity of the attack in these cases is much greater than that of an exhaustive search.

Principle:

The general principle of this attack is based on binary linear equations (Boolean).

2.4.3 Linear equations:

A linear expression (linear equation) is an expression that is written:

$$X1 \oplus X2 \oplus .. \oplus Xn = Y1 \oplus Y2 \oplus .. \oplus Yn \tag{5}$$

with the exclusive Or (XOR),and X1,..Xn, Y1,..Yn are boolean variables (which can only have the values 0 or 1)

This equation can be written:

$$X1 \oplus X2 \oplus .. \oplus Xn \oplus Y1 \oplus Y2 \oplus .. \oplus Yn = 0. \tag{6}$$

Substitution tables (S-Box):

A substitution table generally takes a variable of m bits as input and produces an output of n bits, the inputs and the outputs do not necessarily have the same size. They are used in the symmetrical digits and are generally designed to be non-linear, however the combination of a few inputs with a few outputs can express some linearity.

$$Y = S(X) => X = S^{-1}(Y). \tag{7}$$

.

Example:

Consider a substitution table S represented by the table below: This box takes as input

Input	000	001	010	011	100	101	110	111
Output	010	100	000	111	001	101	101	011

a hexadecimal number composed of 3 bits X1X2X3 and provided as output another hexadecimal number consisting of 3 bits also Y1Y2Y3. Consider the following two linear equations:

$$X1 \oplus X2 \oplus X3 == Y1 \oplus Y2 \tag{8}$$

$$X2 \oplus X3 = Y3 \tag{9}$$

$$X2 \oplus X3 = Y3 \tag{10}$$

The figure below gives the different cases for which the two equations are satisfied:

1

X	Y	$X_1 \oplus X_2 \oplus X_3$	$Y_1 \oplus Y_2$
000	010	0	1
001	100	1	1
010	000	1	0
011	111	0	0
100	001	1	0
101	110	0	0
110	101	0	1
111	011	1	1

2

X	Y	$X_2 \oplus X_3$	Y_3
000	010	0	0
001	100	1	0
010	000	1	0
011	111	0	1
100	001	0	1
101	110	1	0
110	101	1	1
111	011	0	1

Figure 28: first equation and second equation

We notice that the probability of satisfaction of the 1st equation is 4/8, and for the 2nd equation is equal to 2/8.

The linear cryptanalysis approach consists of looking for approximations that have very high or very low probabilities of occurrence.

Note that there should only be approximations defined on the set of inputs and outputs with both high and low probability of occurrence, otherwise the encryption algorithm is considered trivially weak.

2.4.4 Example of application of linear cryptanalysis:

Consider a very simple encryption algorithm that takes 3 bits as input and gives 3 encrypted bits as output. The process takes place over 3 rounds and uses 4 subkeys.

Let P be the 3-bit plain data and let C be the encrypted 3-bit final result.

Turn1: The text P is encrypted with the subkey K1 (XOR), we obtain the text A1.

$$\mathrm{A_1} = \mathrm{P} \oplus \mathrm{K_1} \tag{11}$$

The result goes into a substitution table S1:

$$\mathrm{B_1} = \mathrm{S_1}(\ \mathrm{A_1}) \tag{12}$$

Turn2:

Mixing of the result of the 1st round by the subkey K2, then substitution by the table S2. We thus obtain:

$$\mathrm{A}_2 = B1 \bigoplus K2 \tag{13}$$

$$\tag{14}$$

$$\mathrm{B}_2 = S_2\,(A_2) \tag{15}$$

equation Turn3:

The same process is applied and we obtain:

$$A_3 = B_2 \bigoplus K_3 \tag{16}$$

$$\tag{17}$$

$$B_3 = S_3\,(A_3) \tag{18}$$

Finalization

At the end a final mix is applied with the K4 subkey.

$$C = B_3 \oplus K_4 \tag{19}$$

We assume the following approximations:

$$S_1 : X_1 \bigoplus X_2 \bigoplus X_3 = Y_2 \tag{20}$$

$$S_2 : X_2 = Y_1 \bigoplus Y_3 \tag{21}$$

equation

1st round:

$$\mathrm{A}_1 = \mathrm{P} \bigoplus \mathrm{K}_1 \tag{22}$$

So

$$\mathrm{A}_1 = \left[\mathrm{A}_{1,1} = \mathrm{P}1,1 \bigoplus \mathrm{K}_{1,1},\ \mathrm{A}_{1,2} = \mathrm{P}_{1,2} \bigoplus \mathrm{K}_{1,2},\ \mathrm{A}_{1,3} = \mathrm{P}_{1,3} \bigoplus \mathrm{K}_{1,3}\right] \tag{23}$$

$$\mathrm{B}_1 = \mathrm{S}_1\,(\ \mathrm{A}_1) \Rightarrow \mathrm{B}_{1,2} = \mathrm{A}_{1,1} \bigoplus \mathrm{A}_{1,2} \bigoplus \mathrm{A}_{1,3} \tag{24}$$

$$\mathrm{B}_{1,2} = \left(\mathrm{P}_{1,1} \bigoplus \mathrm{K}_{1,1}\right) \bigoplus \left(\mathrm{P}_{1,2} \bigoplus \mathrm{K}_{1,2}\right) \bigoplus \left(\mathrm{P}_{1,3} \bigoplus \mathrm{K}_{1,3}\right).(\mathrm{l}) \tag{25}$$

2nd round:

$$\mathrm{A}_2 = \mathrm{B}_1 \bigoplus \mathrm{K}_2$$

So:

$$\mathrm{A}_2 = \left[\mathrm{A}_{2,1} = \mathrm{B}_{1,1} \bigoplus \mathrm{K}_{2,1\underline{1}}\ \mathrm{A}_{2,2} = \mathrm{B}_{1,2} \bigoplus \mathrm{K}_{2,2},\ \mathrm{A}_{2,3} = \mathrm{B}_{1,3} \bigoplus \mathrm{K}_{2,3}\right] \tag{26}$$

$\mathrm{B}_2 = S_2\,(A_2)$ **and from approximation** $2\,(X_2 = Y_1 \bigoplus Y_3)$ **therefore**

$(A2,2 = B2,1 \bigoplus B2,3)$

$\Rightarrow B_{2,1} \bigoplus B_{2,3} = B_{1,2} \bigoplus K_{2,2}$**By replacing** $B_{1,2}$ **by its value in (I) we obtain:**

$\mathrm{B}_{2,1} \oplus \mathrm{B}_{2,3} = ((\mathrm{P}_{1,1} \bigoplus \mathrm{K}_{1,1}) \oplus (\mathrm{P}1,2 \bigoplus \mathrm{K}_{1,2}) \bigoplus (\mathrm{P}_{1,3} \bigoplus \mathrm{K}_{1,3})) \bigoplus \mathrm{K}_{2,2}.(II)3rdround:$

$A_3 = B_2 \oplus K_3$ **therefore**

$A_3 = \left[A_{3,1} = B_{2,1} \oplus K_{3,\underline{1}} A_{3,2} = B_{2,2} \oplus K_{3,2}, A_{3,3} = B_{2,3} \oplus K_{3,3}\right]$

$\Rightarrow (B_2,1 = A_3,1 \oplus K_{3,1} and B_{2,3} = A3,3 \oplus K3, \underline{3}$

$\Rightarrow B2,1 \oplus B2,3 = (A3,1 \oplus K3,1) \oplus (A3,3 \oplus K3,3).(27)$ We replace $B2,1 \oplus B2,3$ by its value in (II) we obtain:

$$(\mathrm{A}_{3,1} \oplus K_{3,1}) \oplus (\mathrm{A}_{3,3} \oplus \mathrm{K}_{3,3}) = \left(\left(\mathrm{P}_{1,1} \oplus \mathrm{K}_{1,1,1)\oplus}\,(\mathrm{P}_{1,2} \oplus \mathrm{K}_{1,2}) \oplus (\mathrm{P}_{1,3} \oplus \mathrm{K}_{1,3})\right) \oplus \mathrm{K}_{2,2}\right.$$

By grouping the terms, we thus obtain the final equation:

$(\mathrm{K}_{1,1} \oplus \mathrm{K}_{1,2} \oplus \mathrm{K}_{1,3} \oplus \mathrm{K}_{2,2} \oplus \mathrm{K}_{3,1} \oplus \mathrm{K}_{3,3}) \oplus (\mathrm{P}_{1,1} \oplus \mathrm{P}_{1,2} \oplus \mathrm{P}_{1,3}) \oplus (\mathrm{A}_{3,1} \oplus \mathrm{A}_{3,3}) = 0.$ (III)

We put $\sum K = \left(K_{1,1} \bigoplus K_{1,2} \bigoplus K_{1,3} \bigoplus K_{2,2} \bigoplus K_{3,1} \bigoplus K_{3,\underline{3}}\right)$,

we thus obtain: $\sum K \oplus (P_{1,1} \oplus P_{1,2} \oplus P_{1,3}) \oplus (A_{3,1} \oplus A_{3,3}) = 0$. (IV)

We now have an approximation that depends on:

- Part of the three intermediate keys.
- Plain text.
- Part of the entry from the last substitution table

By applying Matsui's Piling-Up lemma (explained below), we set the value of K to 0 or to 1, and we calculate the probability that this approximation is valid.

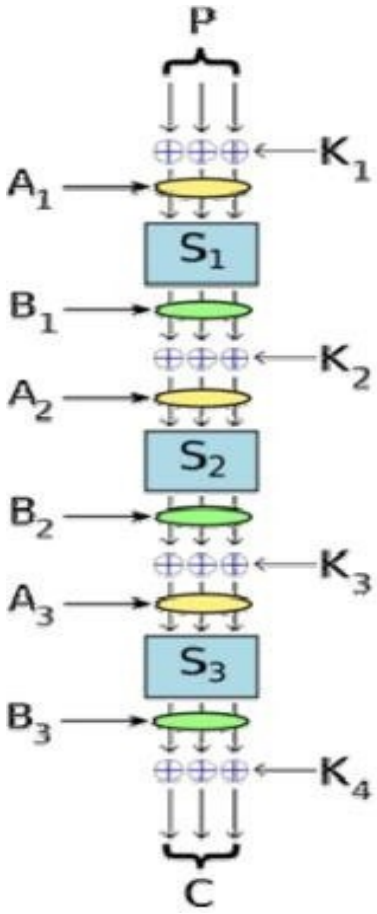

Figure 29: An approximation

2.4.5 Algorithms for linear cryptanalysis:

1-Matsui's Algorithm :

Matsui's Algorithm refers to a family of cryptanalytic techniques developed by Mitsuru Matsui, a renowned cryptographer. These algorithms are designed to analyze and break symmetric key cryptographic algorithms, such as block ciphers. Mitsuru Matsui has made significant contributions to the field of cryptanalysis, particularly in linear cryptanalysis. Matsui's Algorithm encompasses different variations, each tailored to exploit specific weaknesses or properties of the target cipher.

1.1)Matsui's Algorithm 1:

Matsui's Algorithm 1, also known as Linear Cryptanalysis, focuses on exploiting linear approximations in block ciphers. It aims to uncover linear relationships between the plaintext, ciphertext, and key bits, allowing for potential key recovery. By analyzing statistical biases and correlations, Matsui's Algorithm 1 can deduce information about the key and potentially break the cipher.

1.2)Matsui's Algorithm 2

Matsui's Algorithm 2, also known as the Improved Matsui's Algorithm, is an extension of Algorithm 1. It addresses the challenge of performing linear cryptanalysis with a reduced number of known plaintext-ciphertext pairs. By iteratively narrowing down the possible values of the key bits, Algorithm 2 improves the efficiency of the attack, potentially requiring fewer known pairs to recover the key. Both Matsui's Algorithm 1 and Algorithm 2 have contributed to the advancement of cryptanalysis techniques and have been influential in breaking several cryptographic algorithms. However, it's important to note that strong and well-designed cryptographic algorithms incorporate countermeasures to resist known attacks, including those proposed by Matsui. Cryptographers continuously analyze and improve algorithms to ensure their security against various cryptanalytic techniques, including those developed by Matsui.

- **Matsui's Piling-up Lemma:**

  Now that we have linear expressions for S-boxes, how do we combine them to perform linear cryptanalysis, and what kinds of results will we get?

  The simple answer is that we trace the output bits of one S-box to be the input values of other S- boxes, repeating until we have an expression relating only plaintext bits, ciphertext bits, and key bits. But what happens when they combine? With more rounds, the biases of the overall expression are going to change, but how?

  Our natural inclination is that the biases will be multiplicative — meaning that an expression's bias of 1/4, when combined with another linear expression of bias 1/3 (lining up their inputs and outputs appropriately) would be $1/4 \times 1/3 = 1/12$. This is approximately what happens, but not quite. Matsui shows that the linear expressions "pile up" in a different sort of way

- **Matsui's Search for the Best Approximations:**

  The Piling-up Lemma in the previous paragraph provides a useful tool to estimate the strength of a given approximation, but the problem remains how to find the strongest approximations for a given cipher.

  For DES, this open problem was solved by Matsui in 1994. In his second paper, he proposes a practical search algorithm based on a recursive reasoning. Given the probabilities of the best i- round characteristic with 1 i n 1, the algorithm efficiently derives the best characteristic for n rounds. This is done by traversing a tree where branches are cut as soon as it is clear that the probability of a partially constructed approximation cannot possibly exceed some initial estimation of the best n-round characteristic. Matsui's algorithm can be applied to many other block ciphers, but its efficiency varies. In the first place, the running time strongly depends on the accuracy of the initial estimation. Small estimations increase the size of the search tree.

On the other hand, if the estimation is too large, the algorithm will not return any characteristic at all. For DES, good estimations can easily be obtained by first performing a restricted search over all characteristics which only cross a single S-box in each round. This does not work as nicely for other ciphers however. The specific properties of the S-boxes also affect the efficiency of the algorithm.
In particular, if the maximum bias of the S-box is attained by many different approximations (as opposed to the distinct peaks in the DES S-boxes), this will slow down the algorithm.

2.4.6 Lightweight Block Ciphers:

The Advanced Encryption Standard (AES) from NIST has been challenged by a number of lightweight block cyphers, most notably AES-128. Some of these cyphers were created by condensing common, thoroughly researched block cyphers to increase their effectiveness. For instance, to reduce the size of the hardware implementation, DESL is a version of DES where the round function employs a single S-box rather than eight and does not include the beginning and end permutations. As an alternative, several of the algorithms are original, purpose-built block cyphers. One of the earliest light-weight block cypher ideas for hardware limitations was called Present. Lightweight block cypher families SIMON and SPECK were created with simplicity, adaptability, and performance in mind. in both software and hardware. Additionally, there are algorithms from the 1990s like RC5, TEA, and XTEA that have straightforward round structures and are suitable for 241 software environments with limited resources. The list of portable block cyphers in is not all-inclusive. When compared to traditional block cyphers, lightweight block cyphers perform better due to lightweight design decisions like[23].

- **Smaller block sizes:** Lightweight block cyphers, which use smaller block sizes than AES (e.g., 64 or 80 bits instead of 128), can save memory. It should be noted that using small block sizes loosens restrictions on how long plaintexts can be before they must be encrypted. For some of the authorised modes of operations, for instance, outputs of a 64-bit block cypher can be distinguished from a random sequence using around blocks. This may result in plaintext recovery, key recovery, or authentication tag forgeries with non-negligible probabilities, depending on the algorithm[23].

- **Smaller key sizes:** Small key sizes (less than 96 bits), like the 80-bit PRESENT, are used by some efficient lightweight block cyphers. At the time of writing, NIST specifies a minimum key size of 112 bits[23].

- **Simpler rounds:** Lightweight block cyphers typically use simpler parts and operations than traditional block cyphers do. 4-bit S-boxes are preferred over 8-bit S-boxes in lightweight designs that use S-boxes. This size reduction saves a significant amount of space. For instance, the AES S-box required 395 GEs, whereas the 4-bit S-box used in PRESENT only needed 28 GEs. Bit permutations, like those used in PRESENT, or recursive MDS matrices, like those used in PHOTON and LED, may be preferred over intricate linear layers for hardware-oriented designs. Rounds that are simpler might require more iterations to achieve security[23].

- **Simpler key schedules:**Most lightweight block cyphers use simple key schedules that can generate sub-keys instantly because complex key schedules increase memory, latency, and power consumption of implementations. This may open the door for attacks utilising chosen, known, weak, or even related keys. In this situation, it's important to make sure that each key is independently generated using a secure key derivation function (KDF)[23].

- **Present block cipher algorithm :**

  To protect our information, what type of encryption scheme will be utilised, how the secret key will be exchanged, and how will the information be ciphered? Symmetric and asymmetric cryptography are two different types of encryption schemes. We have selected symmetric cryptography for this approach, which uses a single key termed the "secret key," which is known to both the sender and the recipient. Regardless of the position it takes in the binary chain, the information is organised into blocks of a fixed size in this type of encryption, allowing all of the bits to be coded together and attempting to eliminate any potential connections between the encrypted text and the original message. [26].

- **PRESENT:**

  is a lightweight block cipher, developed by the Orange Labs (France), Ruhr University Bochum (Germany) and the Technical University of Denmark in 2007. PRESENT was designed by Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe.The algorithm is notable for its compact size (about 2.5 times smaller than AES [26].

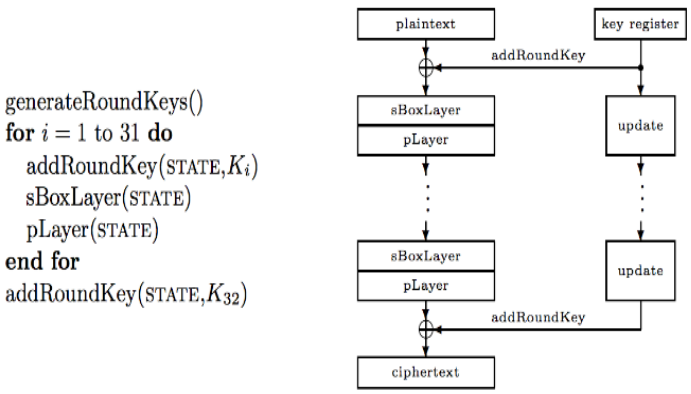

Figure 30: Structure of Present Algorithm

[34]

2.5 Conclusion

In this chapter we have discussed what is Cryptography and Cryptanalysis and their relationship ,we have take a look of different forms of Cryptanalysis:and explained especially what does it mean linear cryptanalysis And the algorithme of Present which was considered the basis for this research.

Chapitre 03

CONTRIBUTION

3 Contribution

3.1 Introduction:

Despite the fact that current symmetric-key cryptography designs mainly rely on security by construction and compelling security justifications (resistance against simple differential or linear attacks, study of algebraic properties, etc.), cryptanalysis still plays a critical role in the process of validating ciphers. It continued to draw a lot of interest, and many cryptanalytic methods have recently been developed.
The following categories apply to cryptanalysis based on algebraic structure algorithms:

(meet-in-the-middle **(MITM)** attack, related-key attack,differential cryptanalysis,linear cryptanalysis, differential-linear cryptanalysis) As we mentioned above, linear cryptanalysis is one of the most powerful analysis techniques used. It can carry out key recovery attacks by using a linear approximation equation that expresses a non-zero correlation between bits of plain-cipher text and the key.[30]
The first linear cryptanalysis was presented to break the Data Encryption Standard **(DES)** successfully in 1994.
This typical linear cryptanalysis needs manual theory derivation and substantial mathematical expertise. Recently, combining deep learning with appropriate statistical cryptanalytic methods has been studied in several papers.[31]
At first, Abadi and Andersen built two neural networks that enable communication between them using a given key without the use of sophisticated encryption. Moreover, a different adversarial network was trained to demonstrate that it cannot retrieve information without the key. Their research did not define net building as it pertains to in cryptography.Soon, Coutinho et al[28].

refined the adversarial network with the chosen-plaintext attack and, in an unsupervised setting, created an unbreakable One-Time Pad technique that investigated the role of adversarial networks in security[32].
Then, several efforts attempted to directly crack ciphers by simulating them but Modern block cryptographic algorithms are more complicated than classic encrypt algorithms, making it impossible for earlier techniques to be effective. As a result, several studies have started to apply more advanced cryptanalysis techniques to increase the availability of attacking using machine learning [33].
Recently,Gohr attempted to use SPECK, a lightweight block encryption method, to apply deep learning. they built a network to more precisely understand the distribution of output difference with a fixed input. However, they didn't give attacks on more complex ciphers.

3.2 Deep learning based linear cryptanalysis of PRESENT:

This memo explores the application of linear cryptanalysis in analyzing the security of the **PRESENT** cryptographic algorithm. Linear cryptanalysis is a technique used to extract information about the secret key in an encryption algorithm by constructing linear approximations and employing neural networks. The memo begins with an introduction to linear cryptanalysis and an overview of the Present algorithm. To train the neural network, a dataset of input-output pairs is generated by running the **PRESENT** algorithm with different inputs and secret keys. An **RNN** model is implemented using TensorFlow, taking plaintext, ciphertext, and fixed secret keys as inputs and outputting the results of the linear equations derived from the linear approximations.
After training the neural network, it is utilized for linear cryptanalysis on the **PRESENT** algorithm. Inputs are fed into the network, and the outputs are analyzed to extract information about the secret key. The results of the linear cryptanalysis are then evaluated to assess the algorithm's security.

3.3 Implementation Details:

The implementation of the PRESENT cryptographic algorithm involves various operations that contribute to its encryption process and explore the details of its implementation ,including the operations of S-Box substitution, P-Box permutation, and key scheduling. We will also discuss the linear approximations used in the analysis.

3.3.1 PRESENT Cryptographic Algorithm Implementation:

PRESENT algorithm is implemented in Python and consists of the following components.

- **S-Box Substitution:**
  The algorithm utilizes an S-Box lookup table to substitute the 4 bits of each nibble in the state. The S-Box substitution is implemented using a Python list called sbox, which maps each input nibble to its corresponding substitution value.

- **P-Box Permutation:**
  The P-Box permutation rearranges the bits in the state based on a fixed permutation table. The permutation table pbox is a Python list that specifies the new positions for each bit in the state.

- **Key Scheduling:**
  The key scheduling process involves generating round keys from the initial secret key. In the code, the generate round keys function takes the master key and generates 32 round keys. Each round key is obtained by applying key scheduling operations, including bit shifting, S-Box substitution, and P-Box permutation, to the previous round key

- **Linear Approximations:**
  Linear approximations are crucial in the analysis of the Present algorithm.

In the code, two linear approximations are provided as examples:Lìnear Equation 1 and Linear Equation 2
This approximations are represented by the linear eq l array, which captures a linear relationship between the bits of the state. These linear approximations are used to ap-

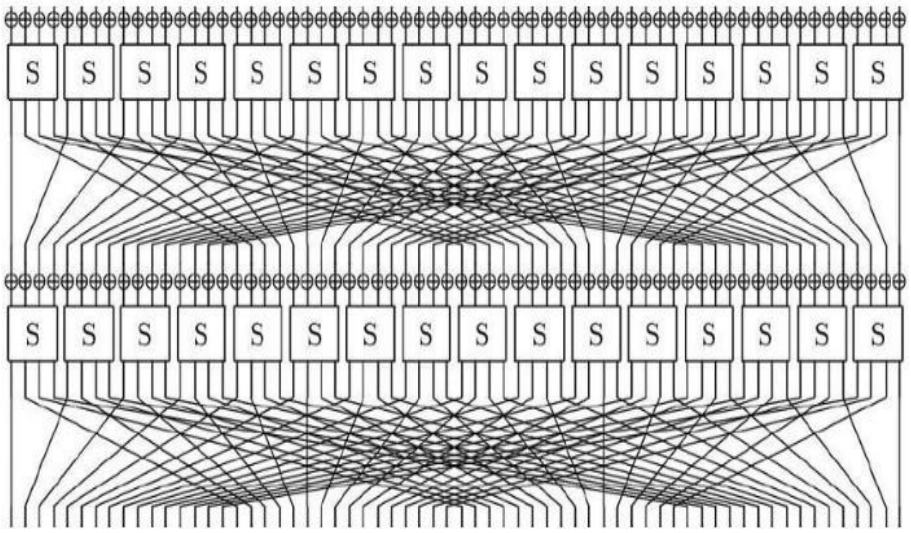

Figure 31: two round of PRESENT

proximate the behavior of the algorithm and extract information about the secret key during the linear cryptanalysis process. By understanding the implementation details of the PRESENT cryptographic algorithm, including the S-Box substitution, P-Box permutation, and key scheduling operations, as well as the linear approximations used in the analysis, we can proceed with generating a dataset, implementing an RNN model, training the network, performing linear cryptanalysis, and evaluating the results. These steps will provide insights into the security of the PRESENT algorithm.

3.3.2 Results And Discussion

Hardware:

Manufacturer	CPU	GPU	RAM
DESKTOP-VSN78IQ	Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz	Nvidua GeForce 920M	8 GB
HP 250 G6 Notebook	intel(R) Core i3-5006U CPU @ 2.00GHZ	intel(R) HD Graphics 520	4 GB

Software:

Software	Description
Anaconda	is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS
Python	Open source high level programming language designed to be easy to read and simple to implement
Jupyter	Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages. It was spun off from IPython in 2014 by Fernando Pérez and Brian Granger.

3.4 Dataset Generation and RNN Training

To perform linear cryptanalysis on the PRESENT cryptographic algorithm, it is essential to generate a dataset of input-output pairs and train a neural network to approximate the linear equations involved. Let's delve into the details of dataset generation and the steps involvedig training the RNN using the generated dataset as follow: Dataset generation :
- Generate plain texts P and master keys K .
- Encrypt P with K by 31round PRESENT cipher and obtain cipher texts C.

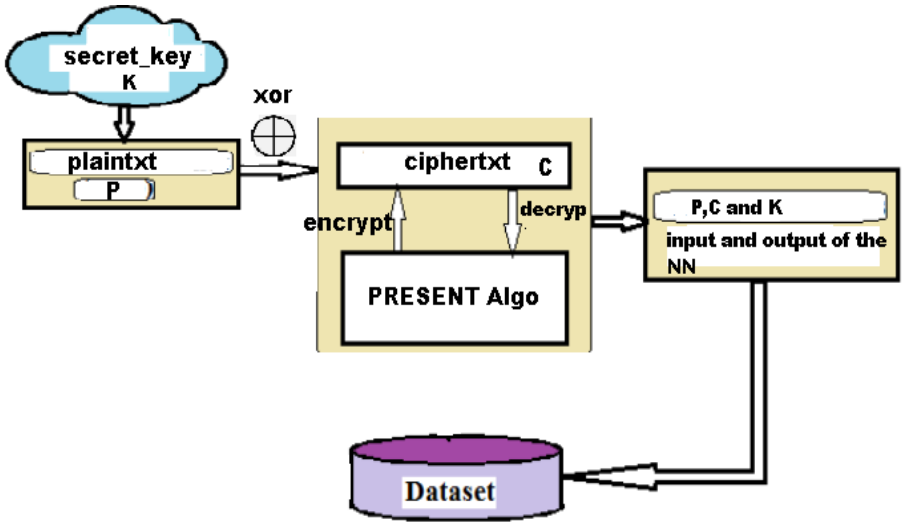

Figure 32: dataset generation

1. Random Selection of Plaintexts and Secret Keys:

   A random string of length 8 is generated using uppercase letters and digits. The random string is then converted into a binary representation, and a key is generated using secrets.token-hex(10), which produces a random hexadecimal string of length 10.

2. Encryption Process:

   The Present algorithm is applied to the randomly selected plaintext and secret key to obtain the corresponding ciphertext. The present-encrypt function is utilized to encrypt the plaintext using the secret key.

3. Dataset Construction:

   Generates 1000 plaintext, key, and ciphertext pairs and stores them in the dataset list. Each plaintext and ciphertext are represented in binary format. The keys are represented in binary format as well.

   It consist of plaintext, key, and corresponding ciphertext encrypted by PRESENT cipher the three components of the dataset, namely plaintexts, keys, and ciphertexts, are extracted using list comprehensions and converted into NumPy arrays. Each component is obtained from the corresponding index of each data item in the dataset.

3.4.1 Network Architectures:

Our objective is to create a learnable, complete model for linear attack ,RNN Model is defined using the TensorFlow Keras Sequential API. In this approach, plaintext and ciphertext pairs are expressed as bits, concatenated, and then input into a neural network.

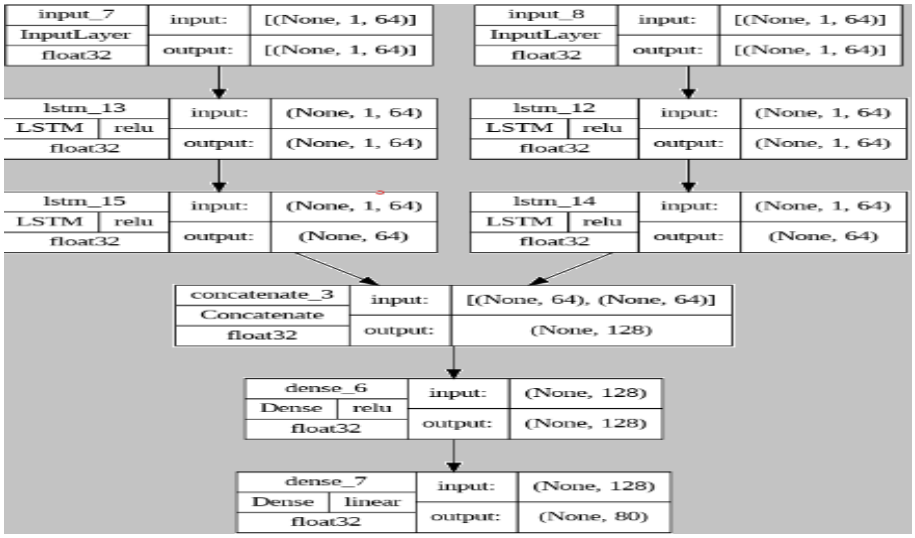

Figure 33: Network Architectures:

3.4.2 Model Compilation and Training:

By generating the dataset and training the RNN model using it, we can obtain a trained neural network capable of approximating the linear equations and performing linear cryptanalysis on the Present cryptographic algorithm. convert the lists of plaintexts, ciphertexts, and keys into NumPy arrays and reshape them to match the expected input shapes of the model.

The input layers of the neural network model. The shape parameter specifies the shape of the input data. The plaintext-input has a shape of (1, 64) and ciphertext-input has a shape of (1, 64).

Apply LSTM layers to the ciphertext-input and plaintext-input respectively. The LSTM layers have 64 units, and return-sequences true is set to return the entire sequence. Additional LSTM layers are added to the flattened ciphertext and plaintext sequences. Both layers have 64 units and a dropout rate of 0.2 to mitigate overfitting. Concatenate the outputs from the previous LSTM layers (lstm2 and lstm) using the Concatenate layer.

The concatenated output is then passed through two Dense layers with 128 and 80 units respectively.

The model is defined using the Model class from Keras. The inputs parameter specifies the input layers, [plaintext-input, ciphertext-input], and the outputs parameter specifies the output layer, output.

The model is compiled with the Adam optimizer, mean squared error (MSE) loss function, and accuracy as the evaluation metric.

The model is trained using the fit function with the input data [plaintext-sequences, ciphertext-sequences] and the target data key-sequences. It is trained for 20 epochs with a batch size of 128.

The model architecture consists of LSTM layers for processing the plaintext and ciphertext sequences, concatenation of their outputs, and subsequent dense layers for the final prediction. The model is trained using the provided dataset to learn the relationship between the plaintext, ciphertext, and keys. The neural network predicts the key corresponding to the pair of plaintext and ciphertext by comparing it with the real key. Finally it calculates the loss function.

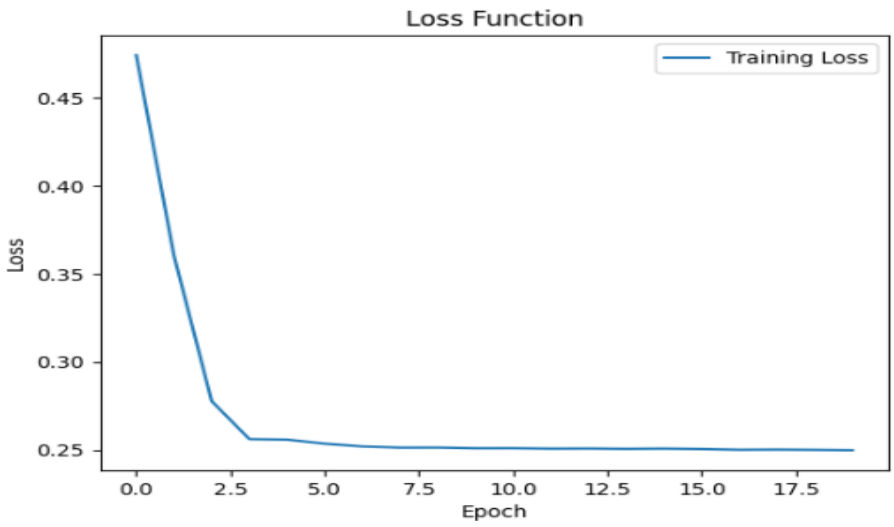

Figure 34: Loss function:

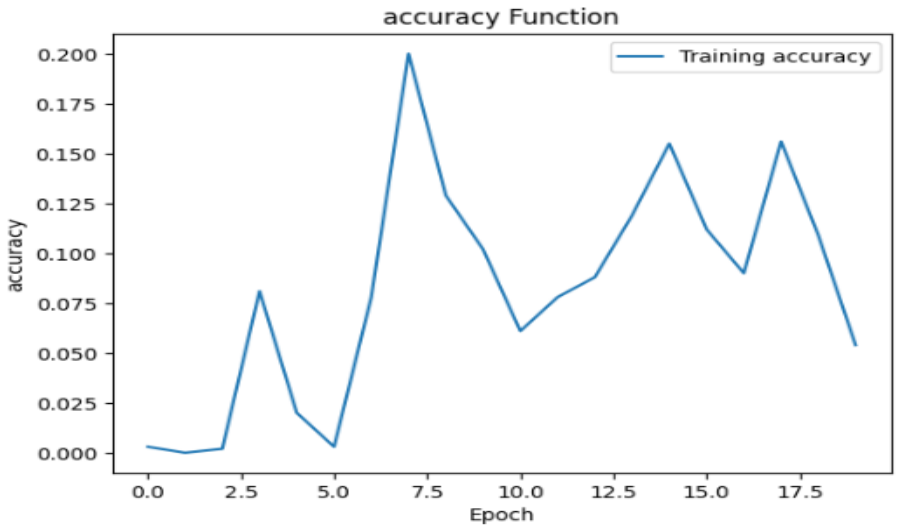

Figure 35: Accuracy function:

3.4.3 Linear Cryptanalysis with the Trained NN:

Once the RNN model is trained using the generated dataset, it can be utilized for linear cryptanalysis to extract information about the secret key. Let's explore how the trained neural network is employed for this purpose.

3.4.4 Utilizing the Trained Neural Network:

1. Inputs for Linear Cryptanalysis: To perform linear cryptanalysis, both P and C are required as inputs to the trained neural network. These inputs are used to predict the corresponding K. the C is obtained by encrypting the P using the Present algorithm.

2. Feeding Inputs into the Network: P and C are preprocessed similarly to the training Phase. the preprocessed P and C are fed into the trained NN. The network processes the inputs and produces a predicted output, which corresponds to the approximated K.

3.4.5 Evaluation Process and Success-Rate:

• Comparing Predicted Key with Actual Key:

To evaluate the success of the linear cryptanalysis, the predicted secret key is compared with the actual secret key used for encryption. The code provided calculates the success rate by measuring the absolute difference between the predicted key and the actual key. If the absolute difference is zero or less than 0.5, it is considered a successful extraction of the secret key.

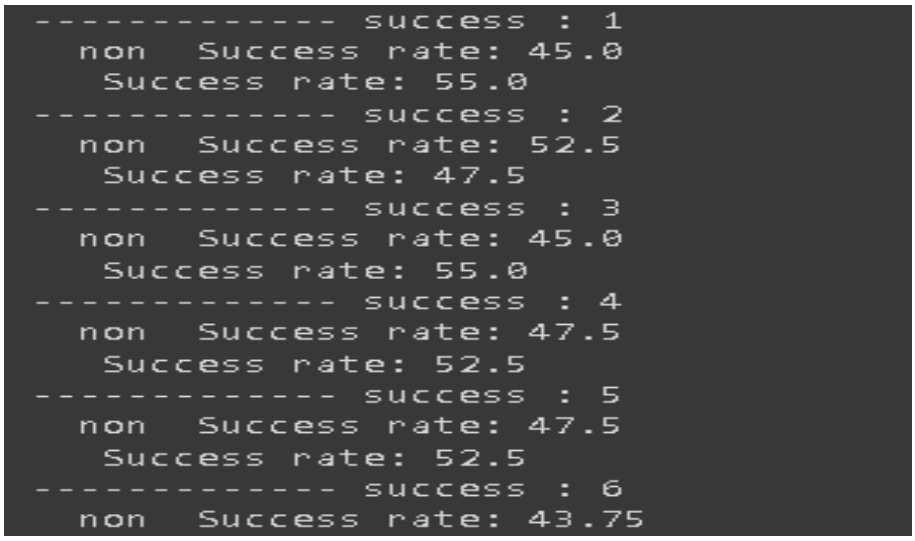

Figure 36: Evaluate the results of linear cryptanalysis:

the results of the experement show that the success rate of our model can acheive 50 percent from the secret key and this indicate that the evaluate of the linear cryptanalysis is cosedered as a succesfull attack .

3.4.6 Determining the Security of the Algorithm:

The success rate obtained from the linear cryptanalysis can be used to assess the security of the Present cryptographic algorithm. A higher success rate indicates a higher vulnerability of the algorithm to linear cryptanalysis, suggesting a potential weakness in the algorithm's security. On the other hand, a lower success rate indicates a stronger resistance against linear cryptanalysis, implying a higher level of security.

By utilizing the trained neural network and feeding inputs (plaintext and ciphertext) nto the network, the linear cryptanalysis process enables the extraction of information about the secret key. The success rate obtained from the analysiS serves as a measure of the algorithm's vulnerability and contributes to the assessment of its overall security.

3.4.7 Main Findings and Contributions:

1. The implementation of the PRESENT cryptographic algorithm was described, including the operations involved such as S-Box substitution, P-Box permutation, and key scheduling.

2. Linear approximations were used to analyze the behavior of the algorithm and extract information about the secret key.

3. dataset of input-output pairs was generated by running the algorithm with different inputs and secret keys, and an RNN model was trained using this dataset.

4. The trained neural network was utilized for linear cryptanalysis by feeding inputs (P and C) into the network to predict the secret K.

5. The success rate of the linear cryptanalysis was evaluated, providing insights into the vulnerability of the Present algorithm to this type of attack.effectiveness of Linear Cryptanalysis

6. Linear cryptanalysis, in combination with the trained neural network, has demonstrated the ability to extract information about the secret K used in the Present algorithm.

7. The success rate obtained from the linear cryptanalysis indicates the effectiveness of this approach in breaking the security of the algorithm.

8. The findings highlight the importance of considering potential vulnerabilities and analyzing the resistance of cryptographic algorithms against linear cryptanalysis.

3.4.8 Implications for the Security of the Present Algorithm:

The success rate obtained from the linear cryptanalysis has implications for the security of the Present cryptographic algorithm.
A higher success rate suggests a greater vulnerability of the algorithm to linear cryptanalysis, indicating potential weaknesses in its design and implementation. Conversely, a lower success rate indicates a stronger resistance against linear cryptanalysis, implying a higher level of security for the algorithm.

3.5 Conclusion:

In conclusion, this memo has focused on the implementation of linear cryptanalysis on the PRESENT cryptographic algorithm using a Recurrent Neural Network (RNN).

General Conclusion

In conclusion, for improving the security analysis of cryptographic algorithms applied for lightweight PRESENT block cipher based in deep learning networks against linear attacks , deep learning modelsmay speed up the cryptanalysis procedure, requiring less time and effort to decrypt or decode a lightweight block cipher. This can be especially useful in situations where quick analysis is essential, like when vulnerabilities are found or when speedy evaluations of new cryptographic algorithms are required.

Furthermore, we spent a lot of time before our engagement consulting with experts and going over relevant scientific literature to get the right perspective on how to apply an recurrent neural network model to our issue. Through this work, we were able to apply and advance our knowledge of deep learning, but in the realm of academic research, the accuracy of results is of paramount importance to ensure the credibility and reliability of scientific findings it entails analyzing the values, unpredictability, and features of the investigational Dataset. On the other hand , the effect of incomplete data on the accuracy of the results ,must include effective data analysis as a vital component, so For accurate data analysis, access to trusted and deep information sources is essential.

So Due to the random and uncurful study of the dataset's values , and the lack of sufficient resources for information it is logical that the accuracy of the results for our modal is not convinced enough. However, the methods described in this work are not without limitations.

It should be noted that even if a reasonably accurate result is not produced as it should ,so the error cannot be ignored 50 percent, it is essential to make sure that the security and privacy of cryptographic systems are not jeopardised by the use of deep learning in cryptanalysis. The trained models must be safeguarded and potential attacks or exploits against them must be avoided.

General Conclusion

Overall, deep learning-based cryptanalysis of lightweight block ciphers offers a curious new technique to increase system security. By utilizing deep learning techniques, it is possible to enhance the security analysis of basic cryptographic algorithms and contribute to the development of stronger and more secure cryptographic solutions for resource-constrained contexts.

Unquestionably, more research and study in this area will result in the development of cryptanalysis techniques and a rise in the general security of cryptographic systems. It is hoped that this work will also help other data scientists to create better predictive models for this large field.

Future works:

- Explores the application of deep learning techniques in the domain of algebraic cryptanalysis.

- designing models that can effectively handle complex Encryption and decryption processes, for testing there security and efficiency.

References

[1] Alexander.(2021).Advanced Deep Learning Application for Engineers and Scientists.

[2] Ganguly,k.(n.d.). Learning Generative Adversrial Networks.

[3] Grubin, p. R. (2001). The Secret Life of the Brain.

[4] Kuhn, D. (1998). Handbook of Child Psychology. Cognition, Perception, and Language.

[5] McCarthy, J. (2007). WHAT IS ARTIFICIAL INTELLIGENCE.

[6] McCulloch, W. S. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics .

[7] Restak, R. M. (2001). The Secret Life of the Brain. Wang, S. (n.d.). Deep Generative Models.

[8] https://blog.quantinsti.com/introduction-deep-learning-neural-network/

[9] David Baillot.(2018).Illustration of a neuron.Why are Neuron Axons Long and Spindly.https://www.analytica-world.com/en/news/pdf/1156404/1156404.pdf

[10] Ade sueb.(2020).The Basic Weights Calculation Behind of DeepLearning.https://adesueb.medium.com/

[11] KACIMI Aymen Issam Eddine/MEDJDOUBI Aymen.(2021).Le deep learning et la bioinformtique pour analyser la degradation du vaccin covid-19 a arnm

[12] https://www.geeksforgeeks.org/cryptanalysis-and-types-of-attacks/

[13] https://www.javatpoint.com/advantages-and-disadvantages-of-cryptography

[14] https://cjrequena.com/(2019-09-06)/cryptography-foundations

[15] https://www.itu.int/en/ITU-D/Cybersecurity/Documents/01-Introduction

[16] Christopher Swenson.(2008) .Modern Cryptanalysis.

[17] Christopher Swenson.(2008).Modern Cryptanalysis.

[18] David Aspinall.(2012).Cryptography ,Introduction Computer Security Lecture 2.

[19] https://www.wolfssl.com/what-is-a-block-cipher/

[20] https://www.geeksforgeeks.org/differential-and-linear-cryptanalysis/

[21] Cryptography and Coding,Steven D.Galbraith,2007

[22] https://members.loria.fr/MMinier/static/images/Crypanalyse-lineaire.pdf

[23] NIST.(2017).Report on Lightweight Cryptography

[24] .Botao Hou, Yongqiang Li, Haoyue Zhao.Linear Attack on Round-Reduced DES Using Deep Learning, and Bin Wu1.University of Chinese Academy of Sciences

[25] Shamsa Khalid.Predicting Risk through Artificial Intelligence Based on Machine Learning Algorithms: A Case of Pakistani Nonfinancial Firms.

[26] https://www.wikipedia.org

[27] Joo Yeon Cho.Linear Cryptanalysis of Reduced-Round PRESENT

[28] Coutinho, M., et al.(2018) . Learning perfectly secure cryptography to protect communications with adversarial neural cryptography. Sensors 18(5), 1306

[29] Preishuber, M., et al.(2018). Depreciating motivation and empirical security analysis of chaos-based image and video encryption. IEEE Trans. Inf. Forensics Secur. 13(9), 2137–2150

[30] Matsui, M.(1994). Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg .

[31] Abadi, M., Andersen,D.G.(2017). Learning to protect communications with adversarial neural cryptography. arXiv Cryptography and Security

[32] Greydanus, S.(2017). Learning the enigma with recurrent neural networks. arXiv Neural and Evolutionary Computing

[33] Paterson, K.G., Poettering, B., Schuldt, J.C.N.: Big bias hunting in amazonia: large-scale computation and exploitation of rc4 biases (invited paper). In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 398–419. Springer, Heidelberg . https://doi.org/10.1007/978-3-662-45611-8 21

[34] Anil G. Sawant.(2019).DESIGN AND IMPLEMENTATION OF "PRESENT BLOCK CIPHER ALGORITHM

[35] https://www.educative.io/answers/what-are-the-types-of-rnn

ملخص

تم استخدام طرق مختلفة ، مثل هجمات القوة الغاشمة ، والهجمات التفاضلية ، والهجمات الخطية ، وهجمات النص العادي المختارة في تحليل التشفير ، الذي يحاول اكتشاف المفتاح السري لخوارزمية التشفير.ومع تطوير الذكاء الاصطناعي أدي إلى زيادة الاهتمام بتحليل التشفير المستند للتعلم العميق..

في هذه الأطروحة ، نقترح طريقة جديدة لتحليل التشفير المستخدم الخفيف الوزن ، والاستفادة من أحدث تقنيات التعلم العميق لتوفير تحليل تشفير خطي (هجوم معرفة النص العادي و النص المشفر) وتحقيق استرداد المفتاح من خلال استخدام الشبكات العصبية المتكررة باستخدام بنية الذاكرة قصيرة طويلة المدى ، لمعرفة الأنماط المعقدة والعلاقة بين النص العادي والنص المشفر واستعادة المفتاح المقابل.

نتائج نموذجنا أعطت نتائج 50٪ لنسبة نجاح الهجمة (استرداد 50٪ من المفتاح)، وهي النتائج الأولى من نوعها لهذا النوع من الهجمات ضذ التشفير المستخدم

الكلمات المفتاحية : تحليل الشفرات الخطي - التعلم العميق - التشفير المستخدم - الكتل الخفيفة.

Abstract

Different methods, such as brute-force assaults, differential attacks, linear attacks, and selected plaintext attacks, have been used in cryptanalysis, which tries to discover the secret key of a cryptographic algorithm. The development of AI has increased interest in DL -based cryptanalysis.

In this thesis, we propose a novel cryptanalysis method for the PRESENT · lightweight block cipher THE symmetric key encryption ,leveraging state of the art DL technologies to provide a linear cryptanalysis (know plainttxt-ciphertxt attack) and achieve a key recovery by employing recurrent neural networks using LSTM architecture, to learn the complex patterns and relation between plaintext, ciphertext and recover the corresponding key .

The results of our model gave 50% results for the success rate of the attack (recovering 50% of the key), which is the first results of its kind for this type of attack against PRESENT.

Keywords: linear Cryptanalysis - Deep Learning - Lightweight Block - PRESENT Ciphers

Résumé

Diverses méthodes, telles que les attaques par brute force, les attaques différentielles, les attaques linéaires et les attaques en(connaissances texte clair),qu' ont utilisées dans la cryptanalyse, qui tente de découvrir la clé secrète d'un algorithme cryptographique. Avec le développement de l'intelligence artificielle, l'intérêt pour la cryptanalyse basée sur l'apprentissage profond a augmenté.

Dans cette thèse, nous proposons une nouvelle méthode de cryptanalyse pour le chiffrement PRESENT Poids léger, utilisant les dernières techniques d'apprentissage en profondeur pour fournir une cryptanalyse linéaire (attaque des connaissances de texte en clair et texte chiffré) et obtenir une récupération de clé en utilisant des réseaux de neurones récurrents utilisant l'architecture LSTM, pour apprendre le complexe et la relation entre le texte en clair et le texte chiffré et la récupération de clé correspondante.

Les résultats de notre modèle ont donné des résultats de 50% pour le taux de réussite de l'attaque (récupération de 50% de la clé), ce qui est le premier résultat du genre pour ce type d'attaque contre PRESENT

Mots-clés : Cryptanalyse linéaire - Apprentissage Profond - Chiffrements PRESENTS - Blocs Légers