Distinguishing Attacks on Secure Channels using Machine Learning

Dissertation submitted in partial fulfilment for the award of the degree

Master of Technology in Computer Science

by

RAVINDRA SINDHIYA

Roll No.: CS2016
M.Tech, 2nd year

Under the supervision of
Dr. Malay Bhattacharyya

Computer and Communication Sciences Devision
INDIAN STATISTICAL INSTITUTE

July, 2022

CERTIFICATE

This is to certify that the work presented in this dissertation titled "Distinguishing Attacks on Secure Channels using Machine Learning", submitted by Ravindra Sindhiya, having the roll number CS2016, has been carried out under my supervision in partial fulfilment for the award of the degree of Master of Technology in Computer Science during the session 2021-22 in the Computer and Communication Sciences Division, Indian Statistical Institute. The contents of this dissertation, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.
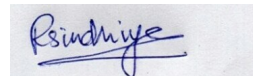
_____
Dr. Malay Bhattacharyya
Assistant Professor, Machine Intelligence Unit
Associate Member, Centre for Artificial Intelligence and Machine Learning
Associate Member, Technology Innovation Hub on Data Science, Big Data Analytics, and Data Curation
Indian Statistical Institute, Kolkata

Acknowledgements

First and foremost, I take this opportunity to express my sincere thankfulness and deep regard to *Dr. Malay Bhattacharyya,* for the impeccable guidance, nurturing and constant encouragement that he had provided me during my post-graduate studies. Words seem insufficient to utter my gratitude to him for his supervision in my dissertation work. Working under him was an extremely knowledgeable experience for a young researcher like me.

I also thank the CSSC and ISI Library for extending their supports in my different ways in my urgent need.

I shall forever remain indebted to my parents, teachers and friends for supporting me at every stage of my life. It is their constant encouragement and support that has helped me throughout my academic career and especially during the research work carried out in the last one year.

Date: 03-07-2022

———————————————————

Ravindra Sindhiya

Roll No.: CS2016

M.Tech, 2nd year

Indian Statistical Institute

Abstract

Block ciphers are the most popular for protecting messages in the field of information security, and their power naturally draws attention. Identifying block ciphers in ECB and CBC mode has been difficult work over the past few decades. This paper proposes a completely new type of distinguishing attack in which we successfully generated ciphers from English text class and from random text class (i.e. rotated plaintext class) with circular rotation of plaintext bits, i.e., rotation by block size n, (first variation for n = 127) and (its other variation) rotation of plaintext (circular rotation) bits by length of plaintext (length (plaintext)-1),

We encrypted plaintexts using DES, DES3, and AES in both variations, using ECB mode and CBC mode.

Under ECB mode average accuracy for first and second variation is : 97.86% and 98.9% using Random forest by encryption using DES, 97% using SVM and 95% using Random forest by encryption using DES3, 86.15% and 91.3% using Random Forest by encryption using AES.

Under CBC mode, average accuracy for first and second variation is : 52.30% & 52.59% using Logistic Regression by encryption using DES, 50.89% using Logistic Regression and 50.25% using SVM by encryption using DES3, 50.8% using Logistic Regression and 50.1% using Random Forest by encryption using AES.

The results demonstrate that ciphertext data can be successfully extracted by constructing a feature based on ciphertext recombination and location specificity.

Contents

List of Figures

List of Tables

1   Introduction

As data increases, security is a major concern in every field in which data is protected by various encryption algorithms. Cryptography techniques are designed to convert plaintext into ciphertext (non-understandable text).

Cryptanalysis techniques are introduced to know the weakness of cryptography techniques to insure the power of encryption algorithms. There are two types of encryption in widespread use today: symmetric-key encryption algorithms and asymmetric-key encryption algorithms. Both sender and receiver use a single common key to encrypt and decrypt messages. The Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are examples of it. Both are block ciphers.

DES is a block cipher with 64 bits key size: 56 bits (+ 8 bits for parity check), rounds: 16, structure: Balanced Feistel network.

AES is block cipher with key size: 128-bits, 192-bits, and 256-bits, also rounds are decided on the basis of key like: 10(128-bits), 12(192-bits), or 14(256-bits).

These Block ciphers have different kind of structures such as substitution permutation networks, Feistel structure & Addition-Rotation-XOR. To know how powerful these block ciphers are various attacks are possible and we will discuss about **DISTINGUISHING ATTACK** with the help of machine learning.

A supervised machine learning approach is used for ciphertext classification, where machine learning (ML) models are trained based on features extracted from documents for different types of classes. Various methods of classification are based on machine learning and statistics. Statistical methods for classification identify statistical parameters such as the frequency of occurrence of letters. Based on the permutation pattern and number of rounds, this machine learning approach extracts ciphertext features.

Various works have been done in this area. In 2006, Dileep et al. [1] proposed a classification approach using Support Vector Machine (SVM) in ECB mode for five block ciphers (DES, AES, RC5, Blowfish and 3DES). In 2011, Manjula et al. [2] proposed a classification approach for 11 encryption algorithms using Decision Tree including classical ciphers, block ciphers, public key ciphers and stream ciphers. In 2013, de Souza et al.[3] proposed distinguishing attack based on a neural network (self-organising map), by clustering and classifying the block ciphers: MARS, RC6, Rinjdael, Serpent and Twofish (the finalist algorithms of AES contest) with a unique 128-bit key, experiment shown that ciphertexts encrypted by the same algorithm stayed close to each other. In 2013, Mishra et al. [4] proposed a classification approach for DES, AES and Blowfish block ciphers based on C4.5 decision tree using entropy and block length as features. In 2016, Tan et al. [5] proposed classification approach based

on support vector machine for 64-bit 3DES, DES, RC5, Blowfish, and 128-bit AES block ciphers. In 2018, Tan et al. [6] proposed classification approach for five type of block ciphers in CBC mode, which are 3DES, DES, RC5, AES, and Blowfish using SVM. In 2018, Huang et al. [7] proposed two-stage identification approach for 42 algorithms based on Random Forest for classical ciphers, stream ciphers, block ciphers, and public key ciphers.

2 Problem Definition

The goal of the problem is to differentiate ciphertext generated from English text class from random text class using circular rotation of plaintext bits, i.e., rotation by block size n ($n = 127$) and (another variation) rotation of plaintext (circular rotation) bits by plaintext length (i.e. length (plaintext)-1).

2.1 Motivation of this dissertation

The first question that arises is why we are rotating plaintext, converting it into ciphertext, and distinguishing it with random text class ciphers.

Suppose Alice and BOB are communicating with a communication channel and all the time sending random text, and in between, for just a few seconds, Alice sends important text (English text). Now if I can train my ML model so that it can classify random text from important text, then I will know that when Alice sends important text (English text) and when Alice sends random text, this task has already been done in research. But if Alice first rotates important text (English text), converts it into ciphertext and then sends it to Bob, it will be hard for an intruder to attack it or to distinguish English text from random text because he doesn't know which possible rotation Alice is sending to Bob. I can train my machine learning model for all possible rotations of English text and random text (ciphertexts) and if I can successfully classify them, then if Alice rotates important text (English text), converts it into ciphertext and then sends it to Bob, then I can classify it from random text with the help of my trained ML model.

3 Contribution of this dissertation

This work proposes a new distinguishing attack on block-based ciphers in which we distinguish English text from random text with a new kind of rotation of plaintext bits, i.e., rotating plaintext bits by a block size of n, first by left shifting n times, then by right shifting n times, for (n = 127) Using this, we obtained 254 rotated plaintext and 1 original plaintext from the English text class and, using a similar approach, 255 plaintext from a random text class. In the second variation of the problem, we obtained 16384 plaintext files, half of them from the English text class and half from a random text class. The experiments are performed using AES, DES, and DES3 (in ECB and CBC mode) and the ciphers for both the classes (i.e., English text vs. Random text) are obtained from one particular encryption algorithm and are classified using Random Forest, Logistic regression, and SVM. Ciphers are classified in ECB mode for both varieties of the problem with an average accuracy around 90–98%, and in CBC mode for both variations of the problem with an average accuracy of around 50–53%.

4 Preliminaries

Cryptography in everyday life provides us with secure services like email services, private web browsing, cash withdrawal from an ATM, etc.

To make this type of communication secure or to send important documents, people use block cipher. These block cipher give guarantees to save our information from intruder attacks. Here we are giving a brief introduction of the block cipher which we have used in our work.

4.1 Block cipher

Block cipher encrypts n-bit block of plaintext with a secret key and give an n-bit block of ciphertext . The block cipher has two parts: round function and key schedule, Figure 4.1(a) shows the structure of a block cipher, & Figure 4.1(b). For each round function key (k_i) is given by key schedule.

(a) General Structure of a Block Cipher   (b) Round Function and Key Schedule
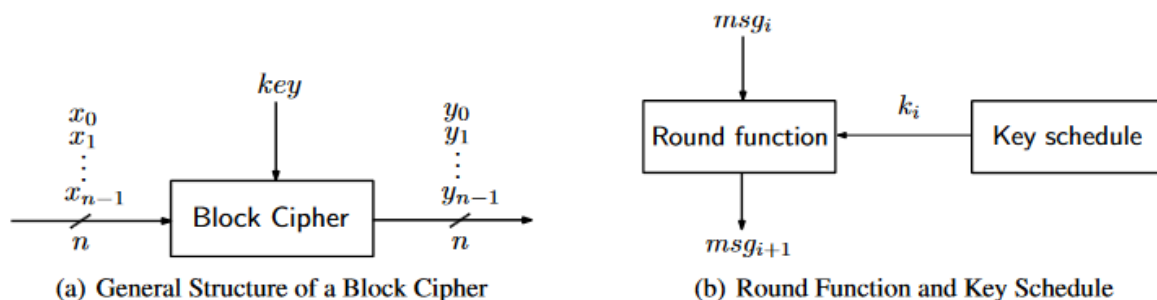
Figure 4.1: Block cipher

Feistel structure & substitution permutation network (SPN) structure are two types of block ciphers. In a Feistel cipher, each state si is divided into two equal halves, say Li and Ri.

The round function h has the following form: $h(L^{i-1}, R^{i-1}, K^i) = (Li, Ri)$, where

$$L^i = R^{i-1}$$
$$R^i = L^{i-1} \oplus f(R^{i-1}, K^i).$$

We observe that the function f does not need to satisfy any type of injective property. This is because a Feistel-type round function is always invertible, given the round key:

(a) The Feistel Structure
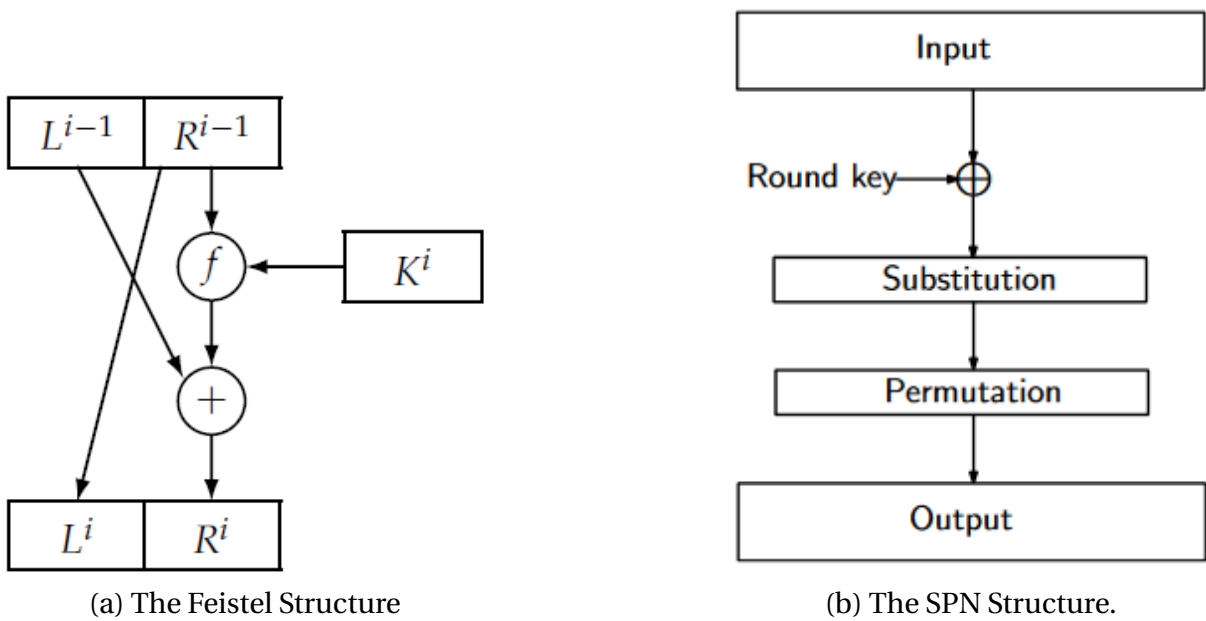
(b) The SPN Structure.

Figure 4.2: structures

$$L^{i-1} = R^i \oplus f(L^i, K^i)$$
$$R^{i-1} = L^i$$

In SPN structure input plaintext will come, then key is taken as inputs in each round function includes adding round key, and layers of substitution, and permutation . DES cipher is example of Feistel structure, and AES cipher is example of SPN structure.

4.2   Operation Modes of Block Cipher

In our work, we will look at two operational modes of Block Cipher: the electronic codebook (ECB)

mode and cipher block chaining (CBC) mode. ECB mode is

It is the simplest and weakest one of all kinds of encryption modes, which uses the same key for encryption of each block of plaintext and uses the same key for decryption (as shown in Figure 4.3). It requires that the plaintext bit length be an integral multiple of the block length. However, since the same ciphertext blocks are encrypted by the same plaintext blocks, ECB is less secure.

It is vulnerable to attacks in some cases, which is a major disadvantage of this method. CBC mode is more secure than that of ECB mode. CBC mode can guarantee data that
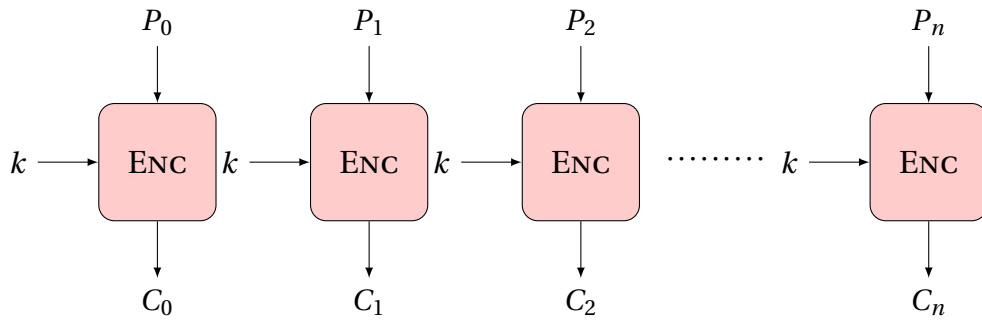
Figure 4.3: encryption operation of ECB mode
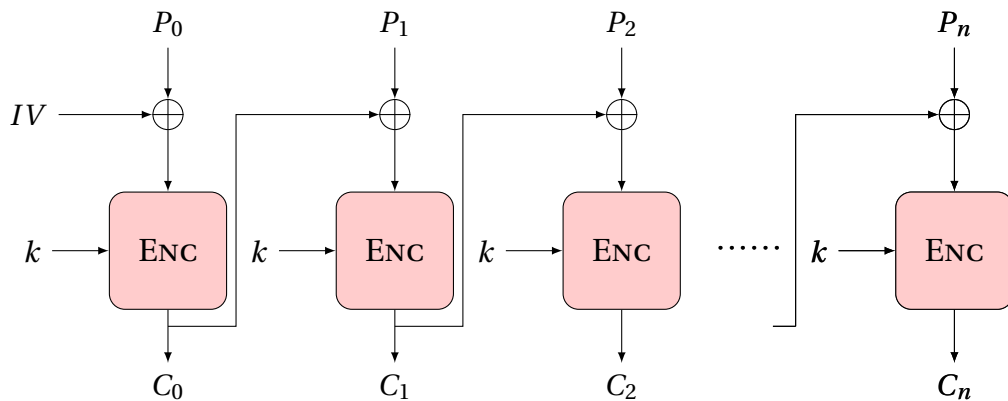
Figure 4.4: encryption operation of CBC mode

every part of data is unique. Instead of encrypting each block directly, CBC uses block chaining, where every subsequent plaintext block is XOR-ed with the ciphertext of the previous block. Cipher block chaining uses an initialization vector (IV) of a certain length and adds it to the plaintext before encryption and sets the previous cipher block as the next IV. Each ciphertext block depends on all the plaintext blocks in front (as shown in Figure 4.4).

As CBC mode is more secure, it is hard to attack it. However, we can see that CBC uses sequential encryption and not parallel. So it is a disadvantage, and the message must be padded to an integer multiple of the block length.

5   Theoretical Insights for distinguishing Ciphers

The distinguishing ciphers or classification is based on foundation of machine learning and neural networks. In our work, for ciphers rotated by block length n=127(left rotation and right rotation) we have used random forest, logistic regression and SVM and for Ciphers formed with plaintext,rotated by length of plaintext, we applied the above ML models. We have followed the steps like First, select the object of classification. Second, extract the feature vectors of the experimental object. Third, select and train the appropriate machine learning classifier. Finally, perform the cipher classification. Now we will briefly discuss all the models we used in our work, followed by feature extraction part. Results are evaluated during classification using accuracy (avg-acc.) and standard deviation (STD-dev).

5.1   Random Forest

Random Forest uses bagging and feature randomness when building multiple decision trees and operates as an ensemble. The goal of a random forest is to create an uncorrelated forest of decision trees, whose predictions, as determined by majority vote or average, are more accurate than those of any single decision tree. A random forest has four main characteristics, they are as follows.

- In terms of accuracy, random forest stands out among other machine learning models.

- It can operate effectively on enormous datasets. It has a very effective feature selection approach, which is crucial for classification.

- It produces an internal, unbiased estimate of the generalization error as the forest building progresses.

Random forest works on idea of bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Assuming the sample set has N data points, it will randomly select n samples from the N training sample set, Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling.This step of row sampling with replacement is called bootstrap. The number of sample data points remains unchanged to N. On all samples, each model is trained independently for the n samples. We have repeated the above two steps m times, obtain m classifiers (or models). The final output is based on majority voting results of the m classifiers. This step which involves

combining all the results and generating output based on majority voting is known as aggregation.

5.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) classify data samples of two classes using an optimal decision boundary or hyperplane this hyperplane is a line dividing a plane in two parts where each class of data points lie on either side, such that the distance on either side of that line or hyperplane to the next-closest data points is maximized. In order to easily classify data points, for a multidimensional sample set, the SVM randomly constructs a hyperplane that best splits your data points into classes so that we can easily put the new data points in the correct category. In a classification task, there can be multiple hyperplanes which could seperate dataset and can provide us good accuracy. Therefore, in order to attain the best classification outcome for linearly separable samples, the learning model SVM finds a margin maximizing hyperplane.

5.3 Logistic regression

Logistic regression is another supervised ML model for binary classification (when dependent variable or target variable is categorical). If data is linearly separable (a hyper plane can separate the data points into two classes or multiple classes). Logistic regression model the probability of a binary response variable as a function of one or more input variables. It learns the PMF of the output label given the input i.e. $p(y|x)$, it uses sigmoid function to define conditional probability of y being 1 for mapping the predictions of a model to probabilities.

5.4 Fully connected neural network

Fully connected neural network, also known as MLP (Multilayer perceptrons). It is made up of many perceptrons. They are made up of an input layer and an output layer. We feed our input data into the input layer and make a decision or prediction about the input from the output layer. The layers between these two layers are referred to as hidden layers (the true computational engine of the MLP). Training MLP involves adjusting the weights and biases of the model so that errors can be minimized.

The major steps followed for training the model are listed below.

1. **Forward pass:** Data is propagated from the input layer and then it is multiplied with weights and bias is added at every layer and a decision is taken on the output layer.
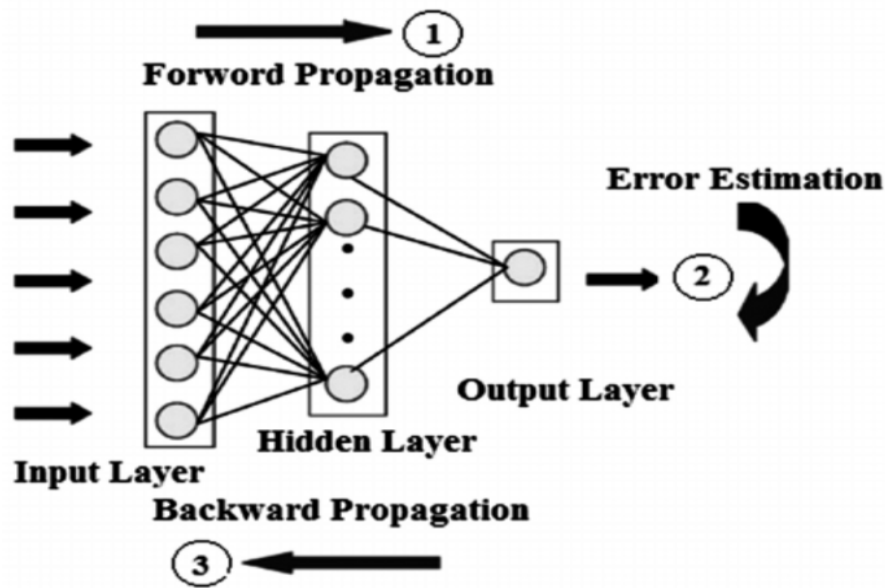
Figure 5.1: Fully connected neural network (MLP working)

2. **Calculate error or loss:**The decision of the output layer is measured against the actual true labels. Calculate the loss (the difference between the predicted and true label). The loss needs to be minimized.

3. **Backward pass:** Backpropagate the loss, find its derivative with respect to each weight in the network, and update the weights of the model. This is the main step in the training of the model, in which the parameters may be adjusted as they move the MLP one step closer to the minimum error. (Any gradient-based optimization approach, such as stochastic gradient descent, can be used for this.

4. Repeat the steps 1-3 over multiple epochs to learn the ideal weights.

5. Finally, the output is taken via threshold function to obtain the predicted class labels.

6 Feature extraction

Since it is difficult to identify that these ciphertext files are produced using this specific block cipher because of their randomness and their core hard mathematics. It is challenging to choose and extract features from ciphertext files, and it is challenging to classify ciphertext files by humans. In order to solve our problem, we must first take into account the internal structure of the block ciphers and their mode of operation. We are aware that each block cipher has a unique block length and that, in ECB mode, each block is encrypted using the same key.

We also know that CBC mode adds a random IV (initialization vector) to the first block, and in CBC mode, each plaintext block would XOR with the previous ciphertext block before being encrypted. Therefore, after rearranging the ciphertext, and with best feature extraction it will be easier for our Ml model to classify that these particular ciphertext belongs to this particular class. See the following figure.
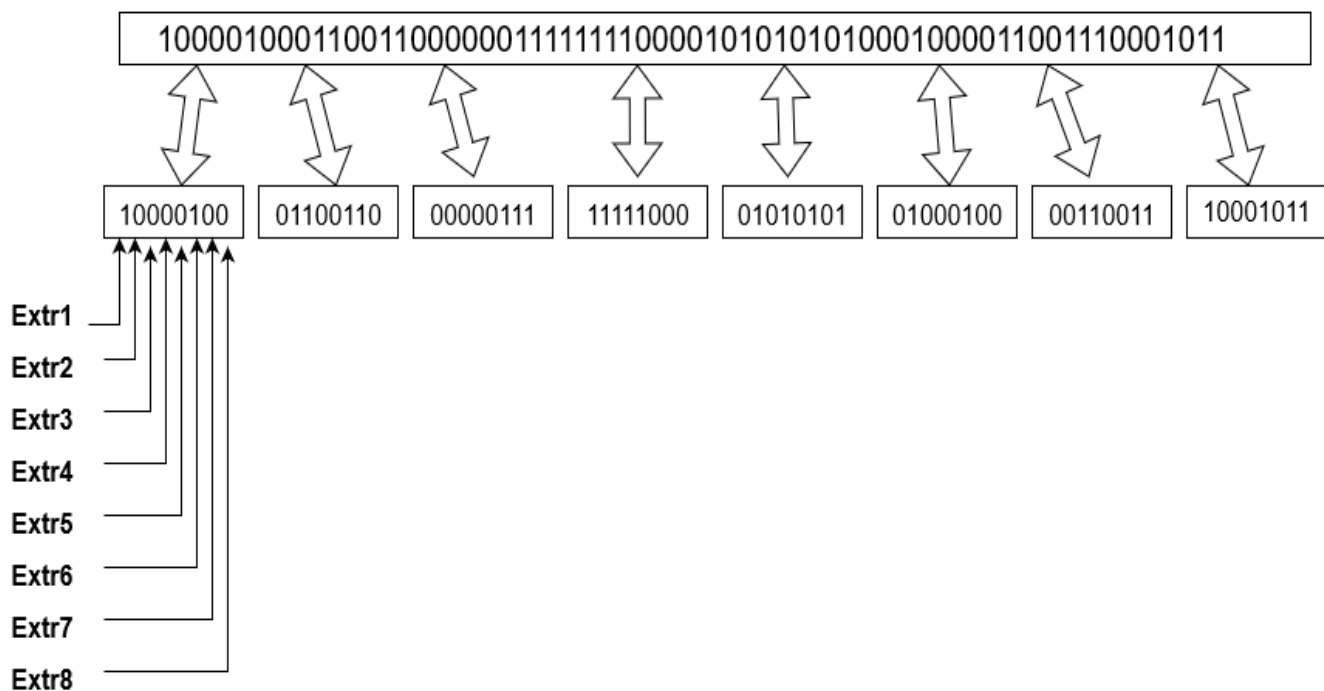

Figure 6.1: Feature selection and extraction

We will do this for each ciphertext as follows.

- Let's suppose this is ciphertext1 of 64 bit (see in Figure 6.1 on top **Extr – Extraction**) length.

- First we will divide it by the length of 8 bits (1 byte), so we will get 8 chunks each of 8 bit length.

13

- Now we will form 8 extraction for ciphertext1.

  - first we will see first byte(see in Figure 6.1 the first chunk or byte 10000100).
  - Now we will take first bit of the first byte and append it to extraction 1.
  - Similarly, the second bit in extraction 2 and so on...8 bit of the first byte in extraction 8.

- After this round, we will repeat the above 3 steps for the second chunk or 2 byte (i.e., 01100110)...so on for the last byte.

- In the end, we will get extraction 1 as 10010001...so on extraction 8 as 00101011.

- In the above example, we appended 1 byte in extraction 1. Similarly, 1 byte in all extractions, so the frequency of each byte is trivial i.e., 1. However, if there are multiple bytes in each extraction then, we will record the frequencies of each byte in each extraction.

  One byte has a total of 2^8 potential possibilities, and the frequency of each case is noted.

  We will get the feature of $2^8 = 256$ (i.e. from 00000000 to 11111111) for extraction 1 and so on... 256 cases for extraction 8. In above example, we will get only 1 frequency for (10000100—1) and the other 255 dimensions will get 0 frequency. Similarly, for all 256 cases, for each extraction, we will get only 1 frequency corresponding to the byte of that extraction.

  Now we will merge all extractions: extraction1+extraction2....extraction8, final dimension is $256 \times 8 = 2048$ (dimension vector for ciphertext).

The ciphertexts were divided into 8-bit chunks (1 byte), and the frequency of fixed bits in each byte was calculated and used to create $(f1, f2, ..., fd)$, where d is the feature dimension.

1. The m-bit long ciphertexts are divided into chunks by bytes to get $m/8$ bytes, and the first bit of each byte is extracted to obtain ciphertext $e1$ as we have shown in above example for extraction1, $e1 = e_1^1, e_2^1, ..., e_{m/64}^1$. The length of $e1$ is $m/64$ bytes(in our example it was 1 byte or 8 bits in extraction1), that is $m/8$ bits.

2. Note the occurrence frequencies of the $m/64$ bytes $e_1^1, e_2^1, ..., e_{m/64}^1$ in $e1$ (in our example it was 1 byte in extraction1 so only got 1 frequency) as the the initial extracted feature.

3. By sequentially extracting the second, third, and eighth bits of each byte of the ciphertexts, the corresponding ciphertexts are obtained, $e^2 = e_1^2, e_2^2, ..., e_{m/64}^2$,

14

$$e^3 = e_1^3, e_2^3, ..., e_{m/64}^3, ..., e^8 = e_1^8, e_2^8, ..., e_{m/64}^8.$$

Record the occurrence frequencies of $e_l^k (l = 1, 2, ..., m/64, k = 1, 2, ..., 8)$ in each byte (this we have to do for each extraction). There are 2^8 different outcomes for each byte, and the frequency of each outcome is noted. One extraction can yield the feature of $2^8 = 256$ dimensions, and since this can be extracted eight times, the final dimension is $256 \times 8 = 2048$.

7   Classification Scheme

As we have shown classification models in chapter 5 and feature extraction method in chapter 6, we will now introduce Flow chart for classification of ciphertext files based on random forest, SVM, logistic regression.

Flow of the classification scheme based on ML models is shown below.

We collected ciphertext files $F1, F2, ..., Fm$ with known classes (i.e 2 in our case) and encryption algorithms. Here, m is the number of files. These ciphertext files may be generated by the same encryption algorithm or a different encryption algorithm (this can be another variation of our problem statement).

Feature extraction is done on all ciphertext files (as we have shown in chapter 6) to obtain the feature $Fea = \{fea1, fea2, ..., feam\}$, where $feai(i = 1, 2, ..., m)$ is a 2048-dimensional vector.

Now we will assign labels to all m vectors as label $= (l1, l2, ..., lm)$, in our problem it is a binary classification problem, so we will assign m labels in which labels for the first class would be 0 and labels for the other classes would be 1. $li(i = 1, 2, ..., m)$ represent labels for either 2 different classes generated by same encryption algorithm or 2 different classes generated by 2 different encryption algorithm, We completed the task of encrypting ciphertexts of two different classes using the same encryption algorithm.Then we will get a set of tagged data (Fea, Lab).

Now we will perform a train-test split to separate the training data from the test data.

Training phase:

Let fea* of F ciphertexts is training data, now we will again split train data into training and cross validation the exact procedure we have shown in **algorithm for training and testing the model**.

Test phase:

(1) Let fea** represent test data of the file F, input fea** into trained classification model, and the model will give the classification results r* of ciphertexts F.

(2) That is, the ciphertext files F that belong to class 0 or class 1 will be tagged r*.

In this work, we applied SVM, random forest (RF), and logistic regression (LR) to classify ciphers.

7.1   Algorithm for training and testing the model

Algorithm 1 Training and testing model

1: $X \leftarrow TrainingDataset$
2: $e^* \leftarrow 0$ $\triangleright$ e^* is error variable
3: **for** $i \leftarrow 1, 10$ **do**
4:    $X \leftarrow X_{TR_i} \cup X_{TE_i}$ $\triangleright$ s.t $X_{TR_i} \cap X_{TE_i} = \phi$, Split X into train and test part
5:    $E^i_{n \times n} \leftarrow 0, n = 5$ $\triangleright$ $E^i_{n \times n}$ is error matrix
6:    **for** $j \leftarrow 1, 5$ **do**
7:     $X_{TR_i} \leftarrow X^j_{TR_i} \cup X^j_{TE_i}$
8:     **for** $p \leftarrow l, l = 1...5$ **do** $\triangleright$ p ranges for all parametres
9:      **for** $q \leftarrow k, k = 1...5$ **do** $\triangleright$ q ranges for all parametres
10:       $TrainModel(p_l, q_k, X^j_{TR_i})$
11:       $e^i_j \leftarrow TestModel(X^j_{TE_i})$
12:       $E^i(p_l, q_k) \leftarrow E^i(p_l, q_k) + e^i_j$
13:      **end for**
14:     **end for**
15:    **end for**
16:    $p^*_l, q^*_k \leftarrow \underset{p_l, q_k}{\arg\min} \{E^i\}, \ \forall (p_l, q_k)$
17:    $TrainModel(X_{TR_i}, p^*_l, q^*_k)$
18:    $e_i \leftarrow TestModel(X_{TE_i})$
19:    $e^* \leftarrow e^* + e_i$

---

8   Distinguishing English from random text with a new kind of rotation by block size n (n=127)

In the data collection phase, we have taken two files, one of English text and one of random text, each of 8192 bytes, and obtained two plaintext files, plaintext1 and plaintext2.

Now, we have converted plaintext1 and plaintext2 into binary bit plaintext files where each character represents 8 bits, in this way, we obtained two plaintext files in binary form. Now we have performed a new kind of rotation of plaintext bits, i.e., (left and right) rotating plaintext bits by

a block size of n bits($n = 127$).

For both plaintext files (English text and random text) we will perform this operation.

- First we will left shift 1 bit of the original plaintext and then we will get a new plaintext. We will append this plaintext to one list.

- Then we will left shift 2 bits of the original plaintext and we will get a new plaintext. We will append this plaintext to that existing list. Similarly, we will do a 3 bit

left shift and so on upto 127 bit left shift.

- We will repeat the above two steps for right rotation, and we will now have 254 rotated plaintext.

- Now we will append the original plaintext file (obtained in the first step, i.e., the original plaintext file) to our existing list, and we will get a total of 255 plaintext.

So, with the above 4 steps, we obtained 255 plaintext files for the English text class, and using a similar approach with the above 4 steps, we will get 255 plaintext files for the random text class. The data of 510 files were stitched as plaintext, in which the first 255 plaintext files represent the English text class and the next 255 represent the random text class.

8.1   Results and Analysis

Finally, we encrypted 510 files in ECB mode and 510 ciphertext files in CBC mode using 3 ciphers (DES, DES3, and AES). Then we applied the feature extraction method as shown in chapter 6 and we got 510 vectors each of 2048 dimensions for each encryption algorithm.

The ciphertexts generated for the English text class by the corresponding encryption algorithm are encrypted with the same key. Similarly, ciphertexts generated for random text classes by the corresponding encryption algorithm are encrypted with the same key. We have used the Python 3.9.13 environment to implement feature extraction and ciphertext classification.

To perform the experiment, we divided the dataset into two parts 33% for testing and the other as a training set. We repeated the above step(train-test split, you can see algorithm for training and testing in chapter 7) 10 times, the overall accuracy will be the mean of 10 accuracy reported also we have listed average accuracy(avg-acc.) and standard deviation (STD-dev) for 10 accuracy in tables for each mode.

8.1.1   Evaluation in ECB Mode

In ECB mode, the classification accuracies along with STD-dev for the classification of ciphertext (i.e., it either belongs to English text class or random text class) are shown in Table 1.

We can see that the ML models discussed above can properly classify the ciphertext files in ECB mode with an average accuracy of 97.86% using Random forest for DES, 97% using SVM for DES3 and 86.15% for AES using Random forest, similarly for others
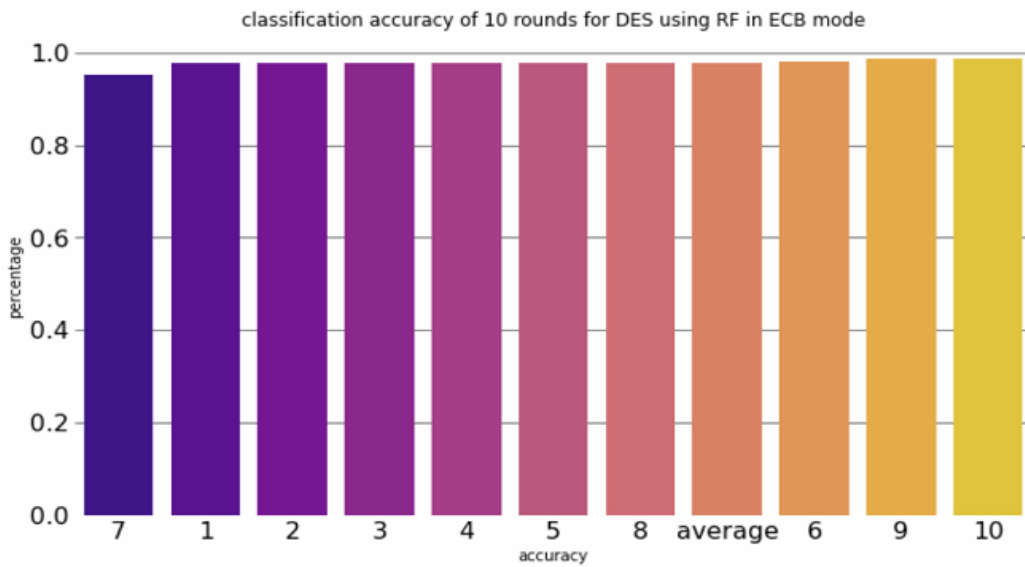
Figure 8.1: Accuracy of 10 rounds for DES in ascending order using ECB mode using RF(rotation by block length)
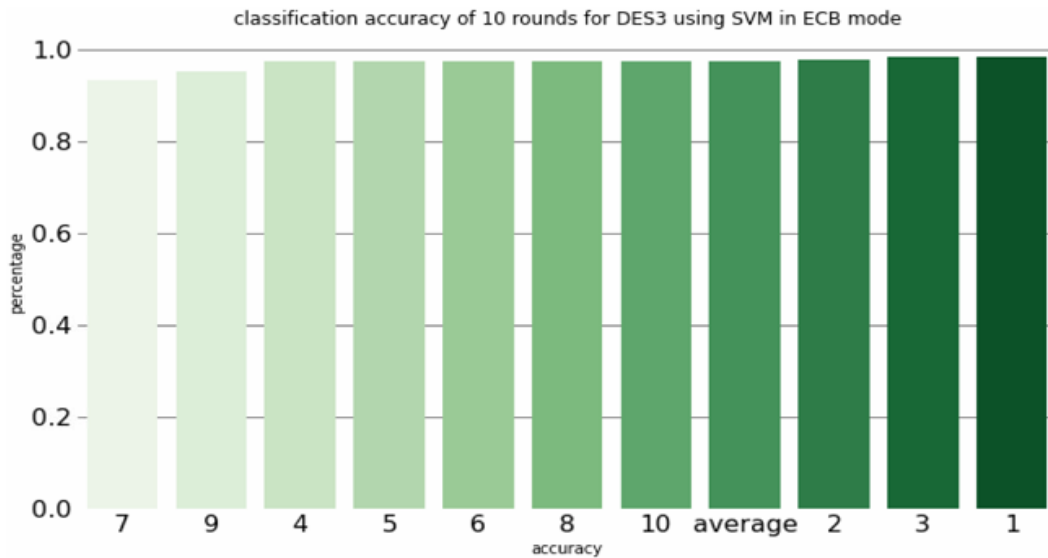

Figure 8.2: Accuracy of 10 rounds for DES3 in ascending order using ECB mode on applying SVM(rotation by block length)

ECB	SVM		Logistic Regression		Random forest	
	avg-acc.	STD-dev	avg-acc.	STD-dev	avg-acc.	STD-dev
DES	0.963	0.014	0.962	0.010	0.9786	0.016
DES3	0.97	0.028	0.912	0.009	0.914	0.040
AES	0.81	0.0001	0.86	0.0001	0.8615	0.0821

Table 1: average accuracy and STD-dev for classification of ciphers in ECB mode(rotation by block length)
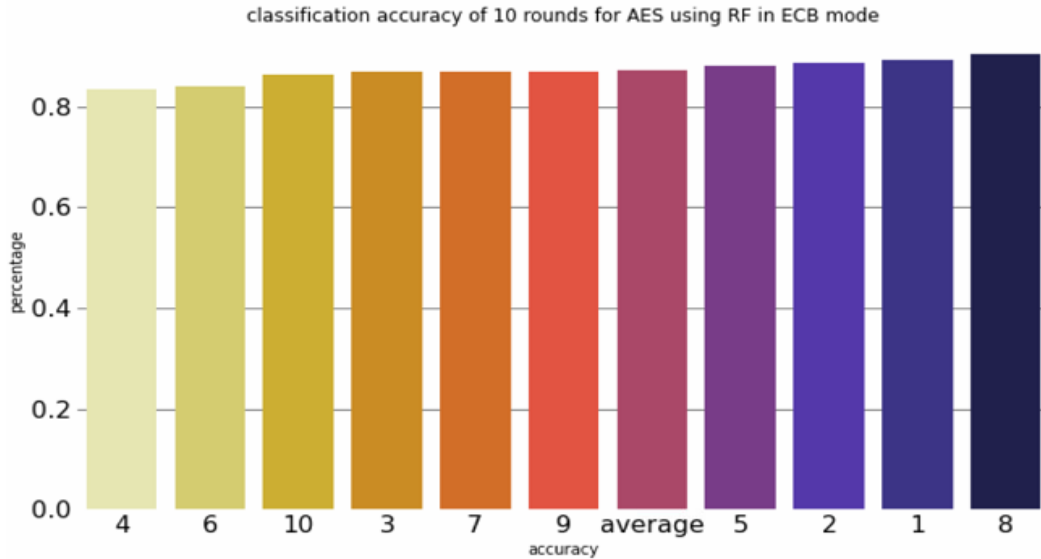

Figure 8.3: Accuracy of 10 rounds for AES in ascending order using ECB mode on applying Random Forest(rotation by block length)

as shown in Table 1. In ECB mode it is easier to classify ciphers in comparision to others modes, so the accuracy is also high for DES and DES3 as expected, we know that ECB encrypts each block of plaintext separately using same key (see figure 4.3), and in comparision to other operating modes, its security is not very high.

8.1.2 Evaluation in CBC Mode

Compared to ECB mode, CBC mode is more complex and very secure. Therefore, it is more difficult for the ML models to classify ciphertexts encrypted in CBC mode. Classification accuracy (average accuracy) for the classification of ciphertext (i.e., it either belongs to English text class or random text class) are shown in Table 2.

we can see in Table 2 the classification accuracies in CBC mode are much lower than those in ECB mode, as expected (&see figure 4.4)in DES the highest is no more than 52.30% (i.e., average accuracy of 10 rounds), similarly for DES3 and AES highest average accuracy is 50.89%. Due to the ciphertext's randomness, CBC mode is less

CBC	SVM		Logistic Regression		Random forest	
	avg-acc.	STD-dev	avg-acc.	STD-dev	avg-acc.	STD-dev
DES	0.4745	0.016	0.5230	0.327	0.5171	0.037
DES3	0.471	0.055	0.5089	0.0002	0.5072	0.034
AES	0.4647	0.0001	0.5089	0.0001	0.4871	0.0033

Table 2: average accuracy and STD-dev for classification of ciphers in CBC mode(rotation by block length)

Figure 8.4: Accuracy of 10 rounds for DES in ascending order using CBC mode on applying Logistic regression (rotation by block length)
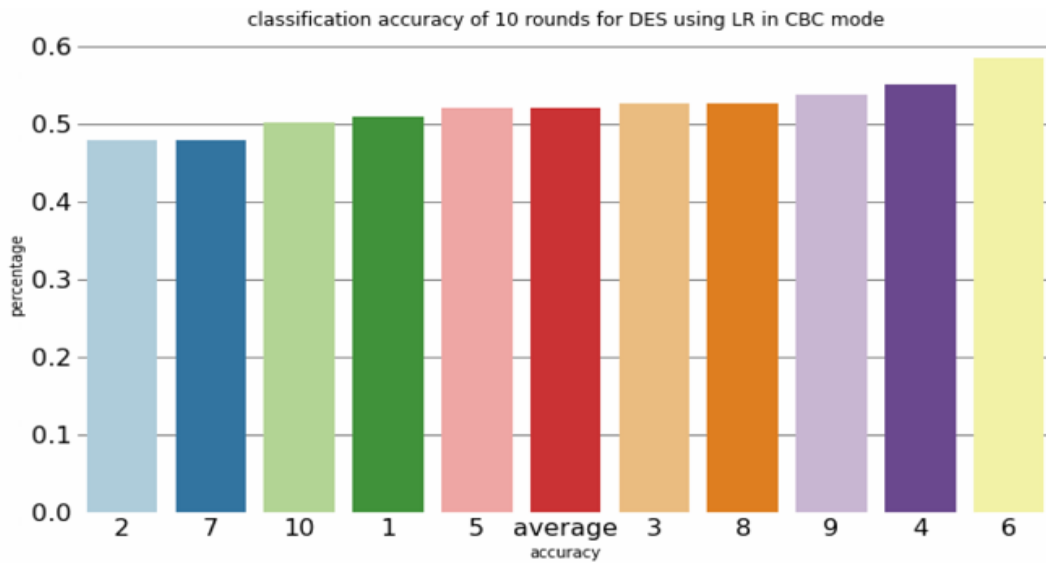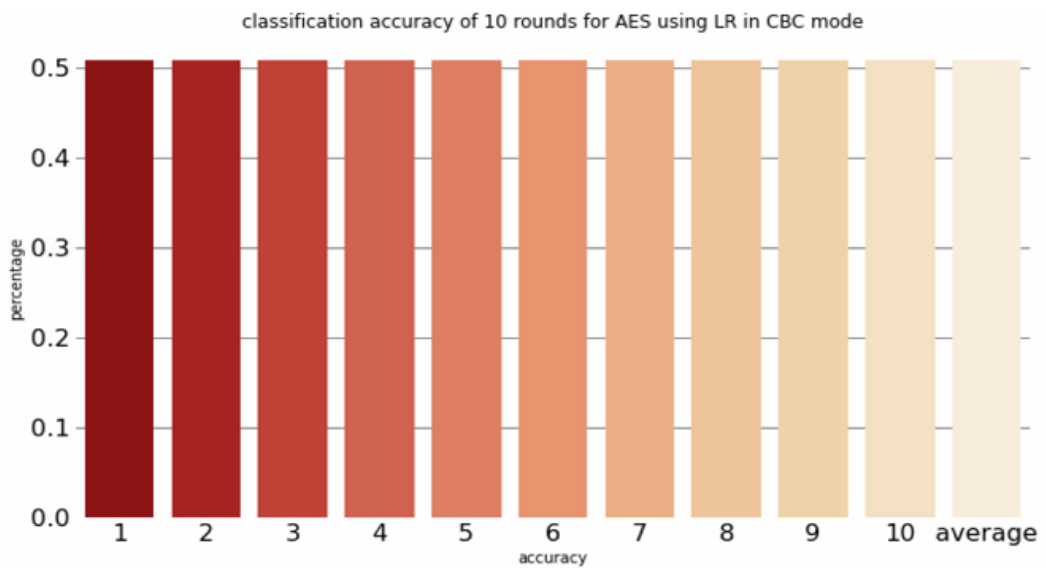

Figure 8.5: Accuracy of 10 rounds for AES in ascending order using CBC mode on applying Logistic regression (rotation by block length)

susceptible to attacks, and it is also challenging to classify ciphertext in CBC mode. There are not any comparisons with previous studies for CBC mode because there are not many classification studies on the CBC mode in the literature.

9 Distinguishing English from random text with a new kind of rotation by length of plaintext

In data collection phase, we have taken two files one of English text and one of random text each of 1024 bytes and obtained two plaintext files say plaintext1 and plaintext2.

Now we have converted plaintext1 and plaintext2 into binary bits plaintext files where each character represent 8 bits so in this way we obtained two plaintext files in binary form, now we have performed new kind of rotation of plaintext (circular rotation) bits by length of plaintext.

For both plaintext files (English text and random text) we will perform this operation.

- First we will left shift 1 bit of original plaintext and we will get new plaintext we will append this plaintext in one list.

- Then we will left shift 2 bit of original plaintext and we will get new plaintext we will append this plaintext in that existing list, similarly we will do operation of 3 bit left shift .....so on till we rotate all bits of plaintext except last bit .

- we can either perform right rotation or left rotation but not both otherwise we will be simply making a copy of plaintext.

- Now we will append original plaintext file (obtained in first step binary bits plaintext file) in our existing list and we will get total 8192 plaintext.

So, with the above 4 steps we obtained 8192 plaintext files for english text class and using similar approach with above 4 steps we will get 8192 plaintext files for random text class. The data of 16384 files was stitched as plaintext in which first 8192 plaintext files represent english text class and next 8192 represent random text class.

9.1 Results and Analysis

Finally, we encrypted 16384 files in ECB mode and 16384 ciphertext files in CBC mode using 3 ciphers (DES, DES3, AES). Then we applied feature extraction method as shown in chapter 4 and we will get 16384 vectors each of 2048 dimension for each encryption algorithm.

The ciphertexts generated for English text class by corresponding encryption algorithm are encrypted with the same key in and similarly, ciphertexts generated for random text class by corresponding encryption algorithm are encrypted with the same

key. We have used Python 3.9.13, environment to implement feature extraction and ciphertext classification.

To perform the experiment, we divided the dataset into two parts 33% for testing and the other as a training set. We repeated the above step(train-test split, you can see algorithm for training and testing in chapter 7) 10 times, the overall accuracy will be the mean of 10 accuracy reported also we have listed average accuracy(avg-acc.) and standard deviation (STD-dev) for 10 accuracy in tables for each mode.

9.1.1 Evaluation in ECB Mode

In ECB mode, the classification accuracies for the classification of ciphertext along with STD-dev (i.e., it either belongs to English text class or random text class) are shown in Table 3:

ECB	SVM		Logistic Regression		Random forest	
	avg-acc.	STD-dev	avg-acc.	STD-dev	avg-acc.	STD-dev
DES	0.967	0.023	0.97	0.0067	0.989	0.002
DES3	0.916	0.038	0.93	0.019	0.95	0.0056
AES	0.91	0.001	0.89	0.00031	0.913	0.0002

Table 3: average accuracy and STD-dev for classification of ciphers in ECB mode(rotation by length of plaintext)

We can see that above ML models can successfully classify the ciphertext files in ECB mode with an average accuracy of 98.9% for DES , 95% DES3 and 91.3% for AES using Random Forest, similarly for others as shown in Table 3. In ECB mode it is easier to classify ciphers in comparision to others modes, so the accuracy is also high for DES and DES3 as expected, we know that ECB encrypts each block of plaintext separately using same key (see figure 4.3), the security is not as high than other modes of operation.

9.1.2 Evaluation in CBC Mode

Classification accuracy(average accuracy) along with STD-dev for the classification of ciphertext (i.e., it either belongs to English text class or random text class) are shown in Table 4.

We can see in Table 4 the classification accuracies in CBC mode are much lower than those in ECB mode, as expected (see figure 4.4)in DES the highest is no more than 52.59%(i.e., average accuracy of 10 rounds), similarly for DES3 it is 50.25% and for AES highest average accuracy is 50.1%. As CBC mode is less vulnerable to attack and

CBC	SVM		Logistic Regression		Random forest	
	avg-acc.	STD-dev	avg-acc.	STD-dev	avg-acc.	STD-dev
DES	0.5050	0.016	0.5259	0.027	0.5031	0.037
DES3	0.5025	0.055	0.4988	0.012	0.4973	0.034
AES	0.4645	0.0001	0.498	0.0001	0.501	0.0318

Table 4: average accuracy and STD-dev for classification of ciphers in CBC mode(rotation by length of plaintext)
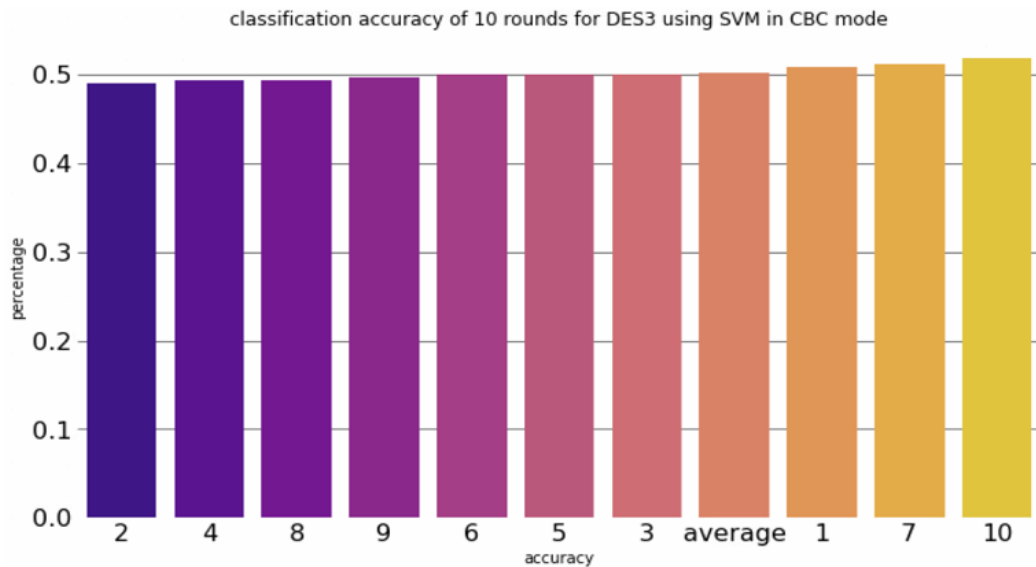

Figure 9.1: Accuracy of 10 rounds for DES3 in ascending order using CBC mode on applying SVM(rotaion by length of plaintext)
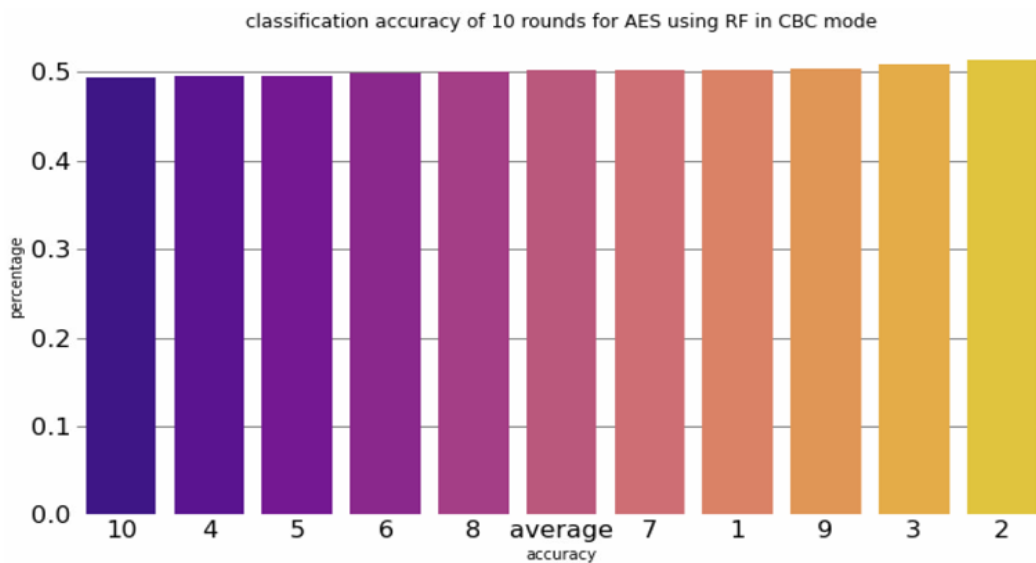

Figure 9.2: Accuracy of 10 rounds for AES in ascending order using CBC mode on applying Random Forest (rotation by length of plaintext)

the ciphertext is also very random, it is difficult to classify ciphertext in CBC because of its randomness. In the existing studies, there are few classification studies on the CBC model so there is no comparison with the existing researches.

10   Conclusion and Future Work

In this thesis, we discussed distinguishing attacks for classification of ciphertext generated from English text class from random text class with a new kind of rotation based on block-size (i.e., 127 in our problem) and its other variation, i.e., classification of English text from random text with rotation based on length of plaintext. We encrypted plaintexts using 3 different algorithms in both variations, using ECB mode and CBC mode.

There is a possibility that there can be some hidden characteristics that exist in the underlying mathematics of the ciphers, and classification of ciphers can be done by recognising features or patterns in ciphertext. Aiming at the problem definition, we made a feature of the frequency of specific locations based on recombination of ciphertext and we classified ciphers generated from English text and random text using SVM, Random Forest, and Logistic regression.

To distinguish English text from random text using a block rotation

- **under ECB mode ,** We achieved average accuracy of 97.86% using Random forest by encryption using DES, similarly for the same problem definition we achieved average accuracy of 97% using SVM by encryption using DES3 and average accuracy of 86.15% using Random Forest by encryption using AES.

- **under CBC mode ,** Because it's security is much higher, we achieved average accuracy of 52.30% using Logistic Regression by encryption using DES, similarly for the same problem definition we achieved average accuracy of 50.89% using Logistic Regression by encryption using DES3 and average accuracy of 50.8% using Logistic Regression by encryption using AES.

For distinguishing English text from random text with a rotation by length of plaintext

- **under ECB mode ,** We achieved average accuracy of 98.9%, 95% and 91.3% using Random forest by encryption using DES, DES3 and AES .

- **under CBC mode** We achieved average accuracy of 52.59% using Logistic Regression by encryption using DES, similarly for the same problem definition we achieved average accuracy of 50.25% using SVM by encryption using DES3 and average accuracy of 50.1% using Random Forest by encryption using AES.

In the future, one may develop more effective feature extraction techniques based on the properties of encryption algorithms to attack security modes like CBC mode. Based on the characteristics of encryption algorithms. Additionally, we may utilise

fully connected neural networks with the appropriate configuration to discriminate between English text and English text with a 1-bit difference in plaintext (which is more challenging task to be done). We can also try the classification research for the identical problem we accomplished using different encryption strategies like AES-256, Blowfish-64, Camellia-128, SMS4-128, stream ciphers, and public key ciphers.

References

[1] Dileep A. D. , C. C. Sekhar, Identification of Block Ciphers using Support Vector Machines// 2006, International Joint Conference on Neural Networks , 2696-2701 .

[2] R. Manjula, R. Anitha, clas sification scheme for 11 encryption algorithms using Decision Tree //2011, International Conference on Computer Science and Information Technology, 237-246.

[3] William A. R. de Souza and Allan Tomlinson, A distinguishing attack with a neural network // 2013, IEEE 13th International Conference on Data Mining Workshops.

[4] S. Mishra, A. Bhattacharjya, Classification scheme for block ciphers based on C4.5 decision tree//2013, IEEE International Conference on Recent Trends in Information Technology, 393-398.

[5] C. Tan, Q. Ji., Classification approach based on support vector machine for block ciphers using SVM//2016, IEEE International Conference on Communication Software and Networks,19-23. .

[6] C. Tan, X. Deng, L. Zhang, classification approach for five type of block ciphers in CBC mode//2018, Procedia Computer Science 131, 65-71.

[7] L. Huang, Z. Zhao, Y. Zhao, Two-stage identification approach for 42 algorithms based on Ran- dom Forest //2018, Chinese Journal of Computers 41, 382-399 .

[8] S. Nagireddy,Master of Science Dissertation – Indian Institute of Technology Madras// A Pattern Recognition Approach to Block Cipher Identification//2008 http://lantana.tenet.res.in.

[9] W. A. R. de Souza, L. A. V. Carvalho and J. A. M. Xexeo, Identification of N Block Ciphers//2011, IEEE Latin America Transactions, v. 9, p. 184-191.

[10] G. Griffin, A. Holub, P. Perona, Califonia Institute of Technology, (2007)

[11] L. Breiman, Machine Learning 45, 5-32 (2001)

[12] S. O. Sharif, L. I. Kuncheva, S. P. Mansoor, IEEE International Conference on Information Theory and Information Security, 1168-1172 (2010).

[13] N. Fergunson, B. Schneier and T. Kohno. Cryptography engineering: design principles and practical applications, Wiley, 2010.