MICROPROCESSORS AND INTERFACING (EP-206)

**CRYPTOGRAPHY WITH 8086 ASSEMBLY**

horizontal line



# INNOVATIVE MID-TERM PROJECT

BY **ADITYA SINGH 2K19/EP/005**

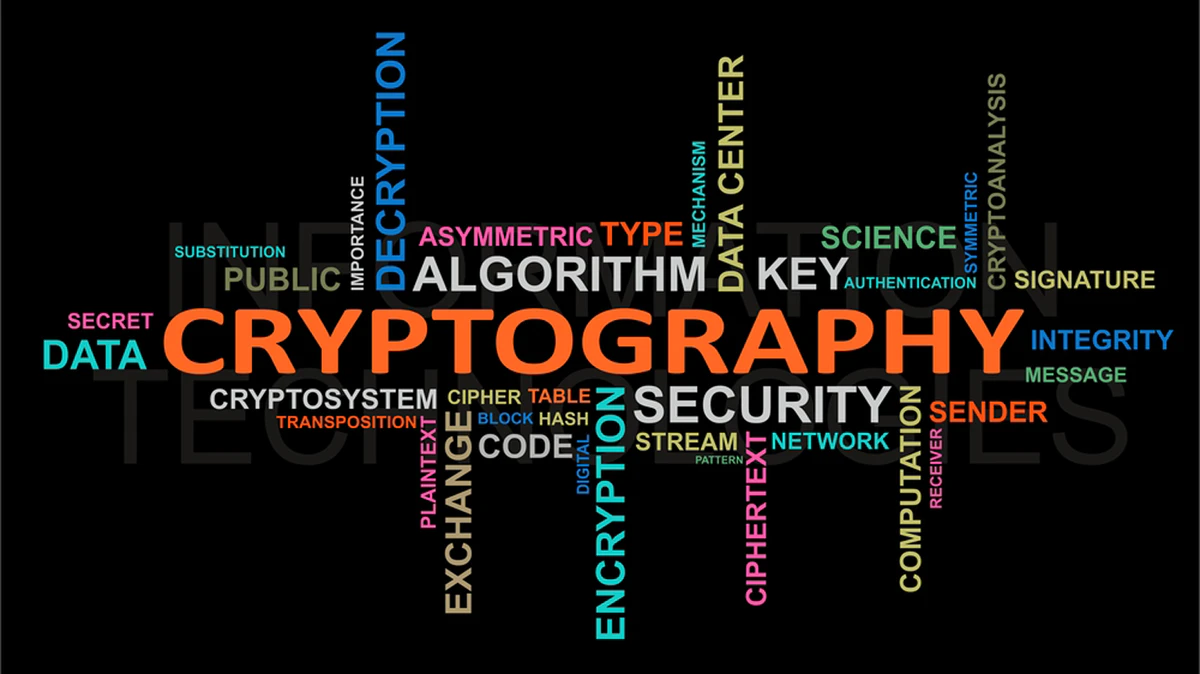
AND **ANSHUL SATIJA 2K19/EP/018**

# TABLE OF CONTENTS

1. Introduction
2. What is Cryptography
3. Terminologies Related
4. Types of Cryptography
   1. Symmetric Key
   2. Asymmetric Key
   3. Hash Functions
5. Caesar Cipher
6. Cryptanalysis of Caesar
7. ROT13 Algo
8. Assembly Program Logic and Code
9. Learning and Future Scope

**What is Cryptography?**

In the modern era, the security of our data is a major concern of us. Everyday we hear about breaches of our data in some or the other way.

This increases the importance of security methods exponentially, for this cryptography is one of the simplest and most effective methods to secure our data. 

It is an ancient art and science of writing in secret messages.

Cryptography comes from Greek word “**Crypto**“ means hiding and “**Graphy**” means writing.

It is the art of achieving security by encoding messages to make them non readable.

Earlier cryptography was effectively synonymous with encryption but nowadays cryptography is mainly based on mathematical theory and computer science practice.

Modern cryptography concerns with:

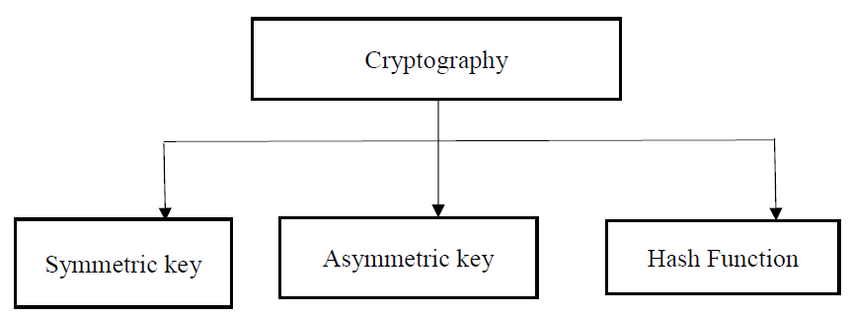
1. **Confidentiality** - Information cannot be understood by anyone
2. **Integrity** - Information cannot be altered.
3. **Non-repudiation** - Sender cannot deny his/her intentions in the transmission of the information at a later stage.
4. **Authentication** - Sender and receiver can confirm each.

**Some Terminologies Related to Cryptography**

* **Encryption** - It is the process of transforming information so it is unintelligible to anyone but the intended recipient.
* **Decryption** - It is the process of transforming encrypted information so that it is intelligible again.
* **Plaintext** - the message to be transmitted or stored.
* **Cipher text** - the disguised message or encrypted message.
* **Cipher** - Algorithm used for encryption and decryption.
* **Protocol** - an algorithm, defined by a sequence of steps, precisely specifying the actions of multiple parties in order to achieve an objective.
* **Alphabet** - a collection of symbols, also referred to as characters.
* **Character** - an element of an alphabet.
* **Bit** - a character 0 or 1 of the binary alphabet.
* **String** - a finite sequence of characters in some alphabet

**TYPES OF CRYPTOGRAPHY**

Cryptography has basically three types.



**Symmetric Key Cryptography:**  
It is an encryption system where the sender and receiver of messages use a single common key to encrypt and decrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange keys in a secure manner. The most popular symmetric key cryptography system is Data Encryption System(DES).

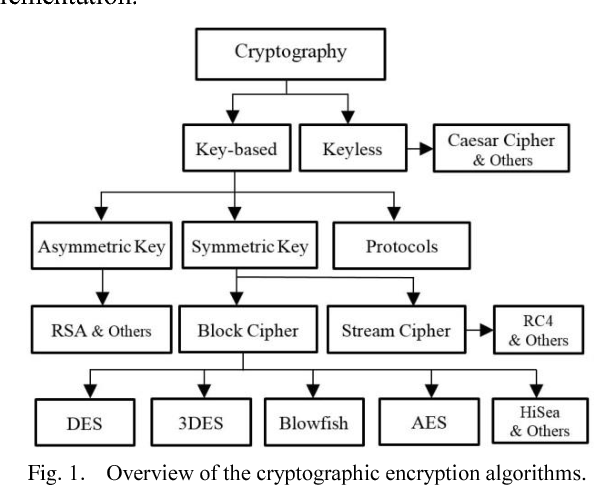
**Asymmetric Key Cryptography:**  
Under this system a pair of keys is used to encrypt and decrypt information. A public key is used for encryption and a private key is used for decryption. Public key and Private Key are different. Even if the public key is known by everyone the intended receiver can only decode it because he alone knows the private key.

**Hash Functions:**  
There is no usage of any key in this algorithm. A hash value with fixed length is calculated as per the plain text which makes it impossible for contents of plain text to be recovered. Many operating systems use hash functions to encrypt passwords.

**DIFFERENT STANDARDS AND**

**ALGORITHMS TYPES**

In this project we are going to use **Symmetric Key Cryptography.**



**CAESAR’S CIPHER**

**(SUBSTITUTION CIPHER)**

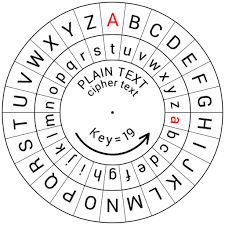
The method is named after Julius Caesar, who used it to communicate with his generals.

It is also known as the shift cipher, Caesar’s code or Caesar shift.

It is one of the simplest and most widely known encryption techniques.

The Caesar cipher method is based on a mono-alphabetic cipher and is also called a shift cipher or additive cipher.

Caesar ciphers is a weak method of cryptography. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.

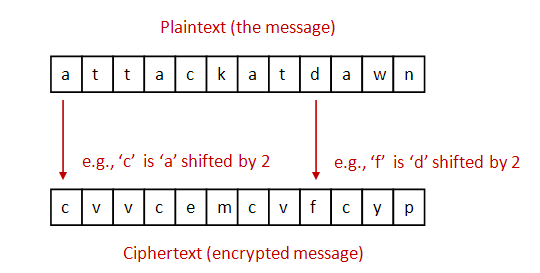


Encryption of a letter x by a shift n can be described mathematically as,

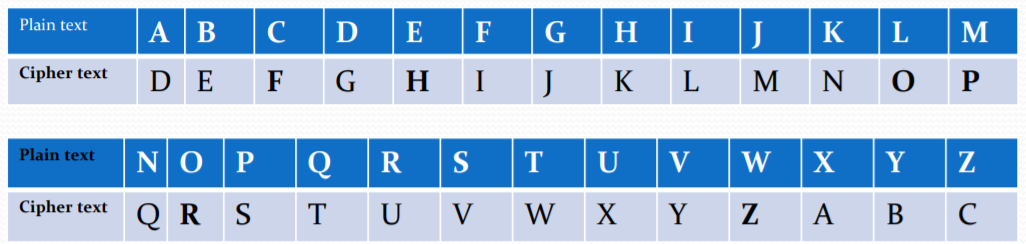
En(x) = (x+n) mod 26

Example: Encryption of a letter A by a shift 2 is, En(x) = (A+2) mod 26 = (0+2) mod 26 = 2

Encrypted letter for A is C .



**Example for KEY = 3**

****

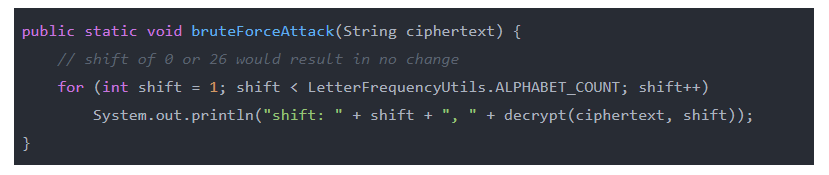
* Plain text is = “**WELCOME**”

Cipher text is = “**ZHOFRPH**”

* Plain text is = “**MICRO**”

Cipher text is = “**PLFUR**”

**CRYPTANALYSIS**



Mallory knows that Bob will be sending his secret food of choice to Alice encrypted using Caesar's cipher. She also happens to know that Bob is allergic to nuts, and wants to maliciously modify his message to Alice to say his food preference is in fact the deadly peanut.

In order to decipher the message, she first needs to determine the shift value used. Thankfully (for her), the English alphabet has only 26 characters, which means the key space is 26. This means the shift value could be any integer between 0 and 26, but since 0 and 26 would result in no change of plaintext it is realistic to consider the possible keys as {1, 2, … , 25}.

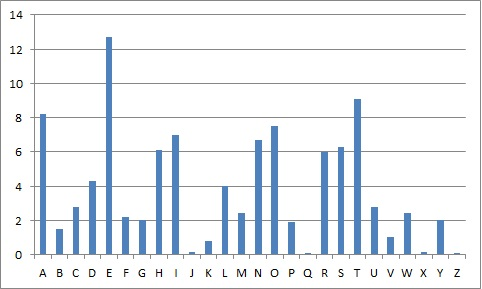
**Brute Force**

Brute force attacks are usually the simplest to implement, but have a very low computational efficiency. This, however, is not so bad in the case where our key space is so small.

Mallory needs to apply every possible shift value to each character in the ciphertext until she finds the value that successfully transforms it into the original plaintext.

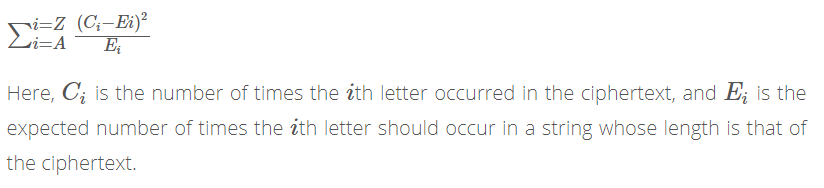
The requires human intervention in that for every possible shift value the decryption is applied and the result printed. The user (Mallory) must then look at each result and decide which is the original plain text.

**Frequency Analysis**

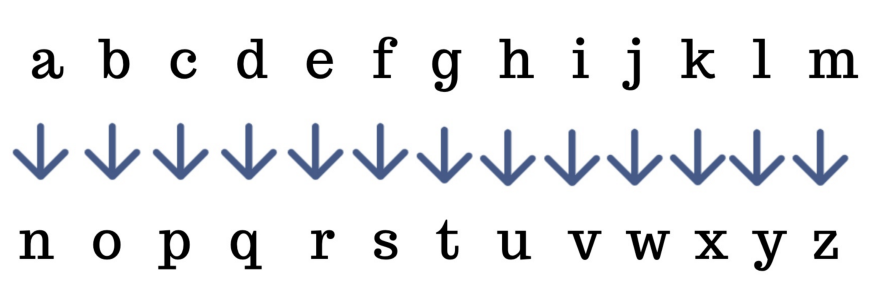


By comparing the frequency of characters that we'd expect to find in the language used with those in the ciphertext, we are able to calculate a best guess at the shift value used.

A chi-squared test is a way of measuring how likely it is that any difference between two sets of values occurred by chance. This happens to be very useful for calculating the shift value. We can have one set of values containing the expected number of occurrences of characters for a string the length of the ciphertext, and another set containing the actual number of occurrences. We can then shove the values into the chi-squared formula:



We can call the decrypt function and make a note of the chi-square for increasing shift values. After we've tried all shifts, the most likely key length is the one that produced the lowest chi-squared value.



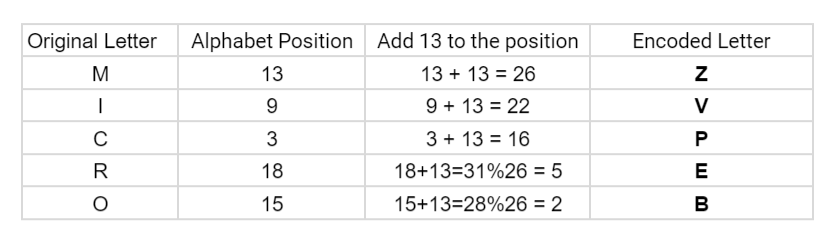
**ROT13**

ROT13 cipher(read as – “**rotate by 13 places**”) is a special case of the Caesar cipher in which the shift is always 13 (i.e. Key = 13).

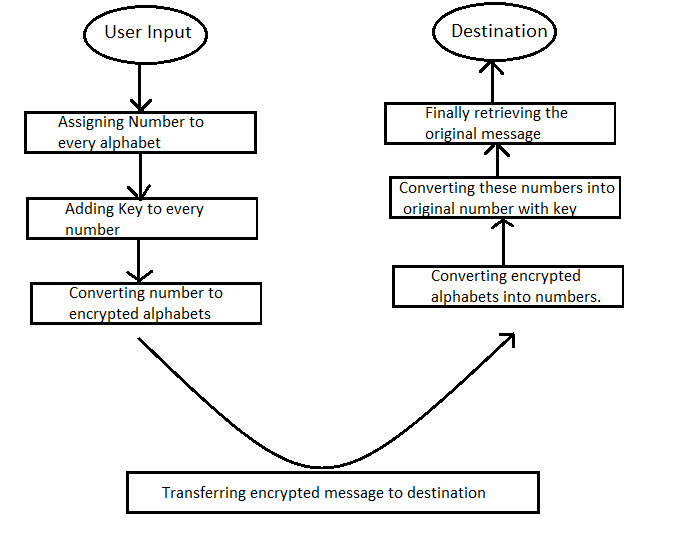
Applying ROT13 to a piece of text merely requires examining its alphabetic characters and replacing each one by the letter 13 places further along in the alphabet, wrapping back to the beginning if necessary. A becomes N, B becomes O, and so on up to M, which becomes Z, then the sequence continues at the beginning of the alphabet: N becomes A, O becomes B, and so on to Z, which becomes M. Only those letters which occur in the English alphabet are affected; numbers, symbols, whitespace, and all other characters are left unchanged. Because there are 26 letters in the English alphabet and 26 = 2 × 13, the ROT13 function is its own inverse.

ROT13 is used in online forums as a means of hiding spoilers, punchlines, puzzle solutions, and offensive materials from the casual glance.

Taking an Example of word MICRO -



**ASSEMBLY PROGRAM**



The program will print out a prompt, then read an input string from the user. Note that the "read" system call does not null-terminate the input string--therefore must use the returned byte count.

The main body of the program will then loop over the characters in the input string, converting the alphabetic characters to their ROT13 equivalents. All non-alphabetic characters (digits, punctuation symbols, etc.) must be left as-is.

Transformed input characters must be stored in successive memory locations. The program stores all transformed input characters before any characters are output.

The program works for any inputs 'A' through 'Z'. The program does not need to validate inputs, and must then output the translated string.

**CONCLUSION AND FUTURE SCOPE**

Using this shift cipher can be a fun way of encrypting messages if you're just passing notes to a friend, but given that the ciphertext has a non-uniform distribution of characters it can easily be decrypted by comparing the frequency of characters in the ciphertext with those expected in the language used.



Nothing in this world is perfect. Everything can be improved in some or the other way, and so is our project.

Future Ideas for project :

* The key will be taken from the user hence making the method of encryption different for everyone, in this way no one except the user knows the key and only he can crack the encrypted message.
* The key can be generated randomly for every word and along with the encrypted message the key for the word will also be sent.
* For special characters, a dictionary will be formed for encryption of special characters.

**THE END**