Triple Layered Fibonacci Caesar Cipher Hybrid

Shivam Sharma
Faculty of Engineering and Technology
Manav Rachna International Institute
of Research and Studies
Faridabad, India
shivam808sharma@gmail.com

Shubh Gaur
Faculty of Engineering and Technology
Manav Rachna International Institute
of Research and Studies
Faridabad, India
shubhgaur37@gmail.com

Harsh Vardhan
Faculty of Engineering and Technology
Manav Rachna International Institute
of Research and Studies
Faridabad, India
harshhh.vardhan@gmail.com

Pronika Chawla
Faculty of Engineering and Technology
Manav Rachna International Institute
of Research and Studies
Faridabad, India
pronika.fet@mriu.edu.in

Madusudhan Mishraa
Faculty of Engineering and Technology
Manav Rachna International Institute
of Research and Studies
Faridabad, India
mcmishra8@gmail.com

***Abstract*— The amount of people that use computer, laptops and mobile platforms daily is staggering. Out of those people, everyone needs internet one day or another which can lead to the discovery of a wonderful amount of content at the disposal of their fingertips. The biggest drawback of using these devices and the internet is threat of a cyberattack. Here we try to build an encryption algorithm that's never been attempted before where we remove the drawbacks of their former selves and improve upon it by combining them. This encryption algorithm will certainly help protect more people from the threat of cyberattacks and secure their privacy and data.**

***Keywords*— cryptography, cryptanalysis, encryption, decryption.**

I. Introduction

Cryptography is the art of hiding messages/data/information from any third party without the proper authentication or authorization. These kinds of techniques have been used since ancient times like this Caesar Cipher used by emperor Julius Caesar to hide messages sent and received by the royalty from opposing kingdoms, bandits, spies and also people within the kingdom of Rome that were conspiring against him. In the modern world cryptographic techniques have been used in the cyberspace that is the digital world which is paramount to the way the world works and the biggest stepping stone to the future. Ancient techniques like the Caesar cipher are obsolete in the modern era as it is very easily broken by simple brute forcing techniques. These kinds of ciphers could be deciphered easily even by humans so they are absolutely useless against the humongous processing power that the machines of today possess. These days the ciphers we use are So complex that they can't be deciphered by a human Through normal means as it could take an indefinite amount of time Which could be larger than the span of multiple human lives. Some examples are-DES dddes aes rsa blowfish etc.

Ciphers are basically reversible techniques implemented through different mathematical functions to achieve

Cryptographic algorithms are generally classified into two types –

A. Symmetric Key Algorithms

This kind of encryption involves using a single key for the process of encrypting a message, which is used for both encryption and decryption. It generally gives a smaller or the same length of text when compare to the original plain text file. This technique is old but it is fast and is also used to transfer large amount of data and it also works on low utilization of resources.

This kind of method is the inverse of the technique known as asymmetric encryption which uses different keys for the process of encryption and decryption. Symmetry Ki cryptography is based on a single share key that all parties are aware of and can use to encrypt and decrypt data.

It is the simplest kind of encryption technique and is also known as secret key cryptography or private key cryptography. Its most common example would be DES, DDDES, AES.

This technique provides less security as there is only one key in use and if it falls into the hands of an attacker, it could ruin the whole encryption process and could result in catastrophic damage to an organization's assets, intellectual properties and so on.

It only provides us with confidentiality for the given data but it cannot provide us the assurance of integrity and authenticity.

In specific cases it is very useful for example someone wants to connect to network that is close via VPN, so during the establishment of the connection the client and server will have to exchange keys, if the keys are symmetric the process will be much faster and smoother and the transfer of data will become much easier and the same cannot be said for a symmetric key encryption.

B. Asymmetric Key Algorithms

This kind of encryption involves using two keys for the purpose of encryption or decryption, It is also called public key encryption because the key is used for encryption or the public key and the private key. We use a public key for encrypting a message and a private key for decrypting a message.

It generally gives larger or the same length of text as the original plain text. This technique is modern what is it is not fast and cannot be used to transfer huge amounts of data as it has higher utilisation of resources.

This metal is the inverse of the technique that is symmetric key encryption which uses the same key for the process of encryption and decryption. A symmetric key cryptography is based on two keys where each person has a separate private key to decrypt messages and public year to encrypt those messages. It is a more complex and newer technique and its most common examples are RSA, Diffie Hellman key exchange and DSA.

This method of encryption provides us with confidentiality, integrity and authenticity for the given data. This method is very useful for broadcasting on multicasting secret messages over a network as only those people who have the proper private key can decrypt the message in corrupted by other specific public key.

II. Literature Review

A. *Caesar Cipher*

Caesar Cipher is a basic encryption technique used since the ancient and medieval times. It is an old school encryption technique that was famous for its usage by Emperor Gaius Julius Caesar of the ancient Roman Empire. He is known for leading the Roman armies in many wars and governed the nation as a dictator. He was secretly dispatched during his reign.

The technique that we use in Caesar cipher is a simple monoalphabetic substitution that comprises of shifting our given string of characters (the plain text) by a certain number of times through a fixed key generally provided by the user. We basically perform a shift with each individual character by N number of times assuming N is the key.

The formula used for the Encryption would be –`

CT = (PT + key) (mod 26)

CT is the Cipher Text alphabet number

PT is the Plain Text alphabet number (provided by the user).

Key is the number of shifts needed to be done (provided by the user).

Mod 26 is used to take the remainder after dividing the key by 26.

The process for decryption is similar to encryption except the fact that it is the opposite of it. We shift the characters back by the number of characters as given in the key for each character in the string and get the desired information, that is, the plain text which we hid successfully.

The formula used for decryption would be –

PT = (CT – key) (mod 26)

CT is the Cipher Text alphabet number.

PT is the Plain Text alphabet number.

Key is the number of shifts needed to be reversed (provided by the user).

Mod 26 is used to take the remainder after dividing the key by 26.

For Example:

Plain Text: HelloWorld

Key : 3

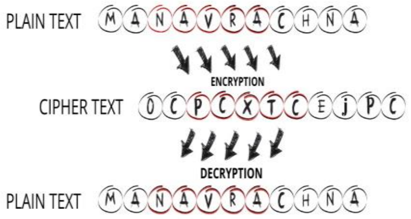
Cipher Text: KhoorZruog

The encryption/decryption of the letter is performed by first converting the letters into numbers, according to the scheme A=0, B =1, C =2,…..,X=23, Y=24, Z=25 and performing a modular arithmetic.

The Algorithm can be represented as follows:

Encryption: C = E(K,P) = (P + K)mod26

Decryption: P = D(K,C) = (C - K)mod26



Another use of Caesar cipher is the ROT 13 which means shift or rotate the alphabets by 13, generally used in learning around the world, though, it has a wide variety of uses where we could see it in different pop culture even going as far as concealing offensive language or other antics such as hiding a response or solving a riddle. The repositioning of the alphabets or units of the strings is deliberately chosen to be 13 to fulfil a few objectives. Since shifting the units of the string twice brings it back to the original text which is useful for easily deciphering the text without building a separate algorithm or program for the decryption process. A function by similar name is also found in python which further illustrates the fame and variety that Caesar cipher has garnered over a course of a long time.

B. *Playfair Cipher*

This is a stronger encryption technique when compared with Caesar cipher which is just a simple mono alphabetic substitution cipher, this cipher encrypts a pair of alphabets called diagraphs.

It was invented by Charles Wheatstone, who is also known for his work with the famous Wheatstone bridge used to measure electrical resistance with a current that is not known.

Wheatstone Bridge is generally taught to high school students and has a very important role in understanding the fundamentals of electrical engineering. It was invented in 1854 but still isn't known by the name of Charles Wheatstone this is because Lord play fair brought it out to the world and made it famous in the world of cryptographic techniques.

In this we create a 5×5 matrix Which we call the grid of letters. We can create a simple matrix with alphabets from a to Z wherein we skip one particular alphabet as we have to create a 5X5 matrix which has 25 characters so generally we skip an alphabet or we place two alphabets in a single cell. We can also take a key from the user or assume one and inserted alphabets in the beginning of the matrix and then insert the remaining alphabets in the matrix to improve its complexity so it can't be easily broken by using simple brute forcing techniques or other methods to decipher or decrypt the message.

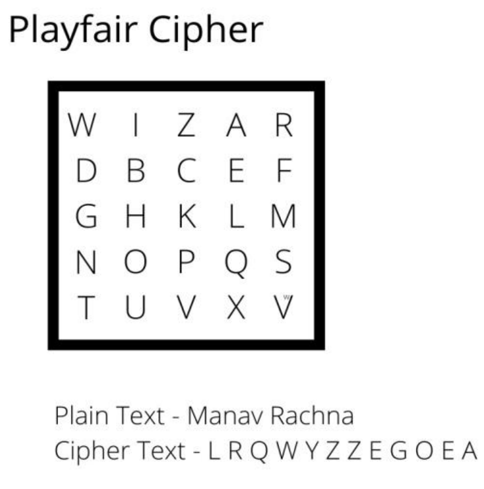
We then divide the plain text into pairs and start encrypting them through the matrix that we just created. We take each pair and encrypt it according to certain set of rules. If both the letters lie in the same column, we shift them by one character to downwards.

If they're in the same row we save them by one character right and if both these conditions are not true then they must form the corners of a rectangle or a square and we encourage them bye replacing them with the opposite corners of the given rectangle or the square. If the number of alphabets aren't even in number then we add a letter with it so we can form a pair for example we can add X to a single character to form a pair, also we use this method in case we have two same letters in a pair and we replace one of them with x as this improves the quality of encryption and makes the ciphertext much more difficult to decipher.

In case of decryption, we do the opposite of encryption as usual. First, we take the ciphertext And divided into pairs. Now we take each pair and decrypted according to the same set of rules that we used to encrypt the plaintext. If both the letters line the same column, we shift them One character upwards.

If they're in the same row we shift them one character to the left. If none of these conditions occur, they form the corners of a rectangle or a square and we decrypted by taking the opposite corners of the same rectangle or the square formed by the two characters. After that we check the plaintext that we deciphered for replacement characters like X and replace it with the appropriate character or none depending on the plaintext as we use X to either replace to same characters in a pair or to form a pair if there is only a single character present in a pair.

The cryptanalysis of this technique suggest it is much better and complex than simple mono alphabetic substitution ciphers like Caesar cipher and is also stronger than other polyalphabetic substitution cycles like the Vigenère cipher.



Playfair Cipher

W	I	Z	A	R
D	B	C	E	F
G	H	K	L	M
N	O	P	Q	S
T	U	V	X	V

Plain Text - Manav Rachna
Cipher Text - L R Q W Y Z Z E G O E A

C. *Vigenère cipher*

The Vigenère cipher belongs to the class of polyalphabetic Ciphers. Another example of it would be the enigma machine. The first mention of this exemplary cipher was made in 1553. It went on to be indecipherable for another 300 years through which it garnered the reputation of the indecipherable cipher.

This cipher particularly requires a secret key to be used effectively. We start by taking the plaintext from the user followed by the key. Then we created 26 x 26 matrix (Vigenère square, Vigenère table or tabula recta).

We check our plaintext and the key and if our key is not of the same length as the plain text we repeat the characters of the key until it is of the same length as the plaintext. The 26 x 26 matrix that we create contains all the alphabets from a to Z in the first row and column. After that, we start mapping the letter according to the letters in the rows and columns. First, we check the alphabet in the plain text and then we check the alphabet in the key and accordingly, we select the specific character in the cell mapped to the letters found in the plain text and the cipher text.
We check to see in the Matrix where is the element is found and that element replaces the given alphabet in the plain text. We continue this process till we come to the end of the plaintext.
In 1553 its working was shown by Giovan Battista Bellaso who also invented it. It was named after Blaise de Vigenère after Credit went to him wrongfully and not to its original inventor who did so three centuries ago
The cipher became indecipherable because it used a key which could be a phrase or a word and could be easily replaced which made the cryptanalysis process difficult in the mediaeval times when it was used by the rulers and nobles as there was no specific study related to cryptography at that time.

III. Proposed algorithm

In our approach, we are trying to combine three encryption algorithms together to generate a **hybrid cipher** which will not be easy to crack because not only we are combining the three algorithms but also modifying the working of two encryption algorithms used and using the third encryption as it is in our hybrid cipher.

The three encryption algorithms used in the proposed approach are:

Vigenère Encryption

Caesar Cipher used in conjunction with Fibonacci Series. [MODIFIED]

Playfair Cipher 6x6. [MODIFIED]

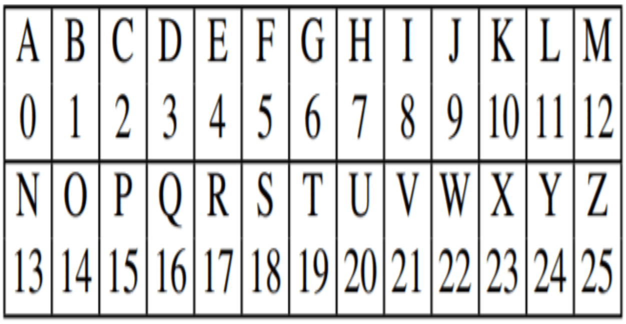
In our approach, the text will be encrypted one by one using each of the encryption algorithms mentioned above and in the same order as above.

By modifying the existing algorithms of Caesar cipher and Playfair cipher, we are trying to overcome their limitations which ensures that the cipher text generated is highly enforced and not easy to crack using conventional methods.

The limitations of Caesar cipher and Playfair cipher are summarized below: -

In Caesar cipher, a numeric key(n) is required in the range (0-25) and then that numeric key is used to shift each character by that number of places to the right using the alphabetic table.

ALPHABETIC TABLE



A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Now,

Encryption with key n, can mathematically be written as:

$$E_n(x) = (x + n) mod\ 26$$

Similarly,

Decryption with key n, can mathematically be written as:

$$D_n(x) = (x - n) mod\ 26$$

Note:

X represents each character from the message taken one at a time.

MOD 26 is taken so that (x-n) and (x+n) always remain in the range of 0-25.

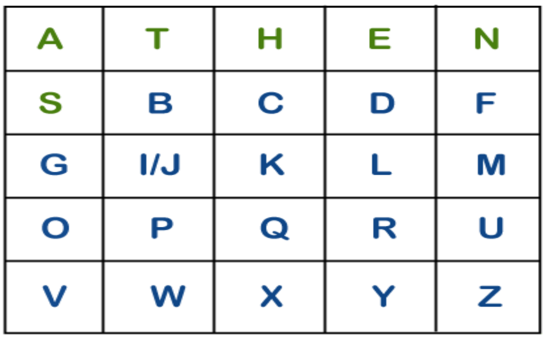
Using the old traditional Caesar cipher technique is obsolete as one can easily decipher it using brute-forcing, as there are not a wide range of keys available. At any time, key will be in the range of 0-25 only and the same key will be used to shift each character of the plain text.

In Playfair Cipher, the message is encrypted using digraphs approach (taking two characters together at a time) instead of a single letter using a 5 X 5 matrix generated using the key, commonly known as the Playfair matrix.

In this technique, a 5 X 5 matrix (Playfair matrix) consisting of alphabets is generated using the key provided such that no character is repeated. As there are only 25 cells in a 5 X 5 matrix so J and I are considered equal and reside in the same column and J is always considered as I or vice versa.

For Example,

If the key is **ATHENS**, then Playfair matrix would look like the following:



A	T	H	E	N
S	B	C	D	F
G	I/J	K	L	M
O	P	Q	R	U
V	W	X	Y	Z

A. LIMITATION 1:

This is one of the limitations of this technique, as J and I are considered equal, there is no way to find out during decryption whether the character that has just been decrypted was J or I which leads to inconsistency.

If we consider J as I in the Playfair matrix, then after decryption wherever J was present in the plain text, I would be there.

If we consider I as J in the Playfair matrix, then after decryption wherever I was present in the plain text, J would be there.

After constructing the Playfair matrix, the next step is to split the plain text in pairs of two, such that if any letter appears twice (side by side), put X at the place of second occurrence and also, if a single character remains at the end, pair it with X.

For Example, JAZZ can be split as:

First pair: JA

Second Pair: ZX

Third Pair: ZX

Digraph: JA ZX ZX

B. LIMITATION 2:

This arises the second limitation of Playfair cipher,

If X is present side by side in the text that we want to encrypt, then we won’t be able to pair it with another X as it will generate the same result and will also lead to the violation of the rule that no two characters can be together.

For Example,

HEXXXO

Splitting into pairs:

First Pair: HE

Second Pair: **X**X (Rule Violation)

Third Pair: **X**X (Rule Violation)

Fourth Pair: **X**O

Now, when we decrypt the cipher text that was generated using this technique, we need to remove all the occurrences of X from the result, so that we get the final plain text because “X” is the character that we appended for the scenario explained above.

C. LIMITATION 3:

This gives rise to the third limitation of Playfair Cipher,

Suppose X was present in the string that we wanted to encrypt.

After encryption, if we try to decrypt then we must remove all the occurrences of X from the result we got as “X” is also the character that we use for appending to letters for making digraphs.

This would lead to loss of information as there is no way to know which X was used for appending and which X was already present in the ciphertext.

For Example,

PT: HEXO (to be encrypted)

Key: YU

Ciphertext generated after encryption,

CT: DFTR

Now, if we try to decrypt it, we will get the result as HEXO and according to the rule we must also remove X from it as X is used for appending, so the final result would be **HEO** which is not the result we expected as the plaintext contained X in it.

So, the modifications proposed by us aim to correct these limitations as well as create a brand-new three-layered **hybrid encryption technique** which could be used in the future for protecting digital data confidentiality.

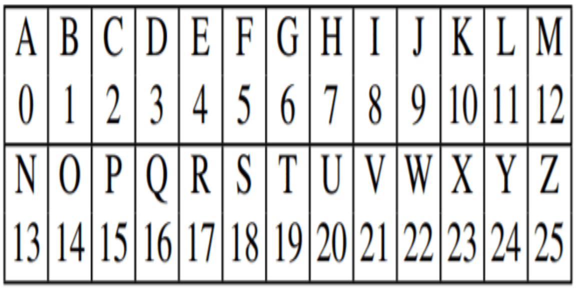
D. Modified Caesar Cipher using Fibonacci Series

In this technique, characters in a string are shifted based on their position and the corresponding number in the Fibonacci series.

Only alphabets can be encrypted using the following technique.

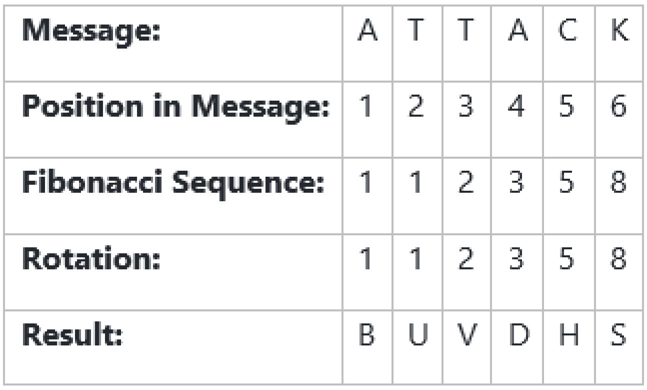
Alphabetic Table is used to rotate the character which is shown below:

ALPHABETIC TABLE



A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Working:



Message:	A	T	T	A	C	K
Position in Message:	1	2	3	4	5	6
Fibonacci Sequence:	1	1	2	3	5	8
Rotation:	1	1	2	3	5	8
Result:	B	U	V	D	H	S

In the above depiction we are encrypting ATTACK using the proposed approach.

We first mark the positions of each character in the message and then based on that position find the Fibonacci term corresponding to that position and shift that character by the value of that number to the right.

The Mathematical formula can be summarized as below:

CT=(X+Fibonacci(i)) MOD 26

where,

X= Character

I= Position of X in the string

Fibonacci(n)=nth Fibonacci Number

Using this approach to encrypt text using the traditional Caesar cipher technique, we do not require a key and it also overcomes the limitation of original Caesar cipher algorithm which was that it shifts each character in the string by the same amount, but here, every character in the string to be encrypted is shifted by a different number which is given by the Fibonacci sequence, thus, enforcing it.

E. Playfair Cipher 6x6

The original Playfair encryption used a 5 X 5 matrix generated using an alphabetic key due to which we had to consider I and J to be same and there was no way to distinguish I and J from each other after decryption.

In this modified approach, instead of using 5 X 5 matrix we are using a 6 X 6 matrix to so that there are enough places for all the alphabets to fit in. There are in total 36 places in the matrix, out of which 26 are occupied by alphabets and the remaining 10 are occupied by numbers from 0-9.

Also, we have modified the pairing approach in which the character used for pairing is always a number which can be decided using the key which resolves the limitations 2 and 3 of the original Playfair technique as explained above.

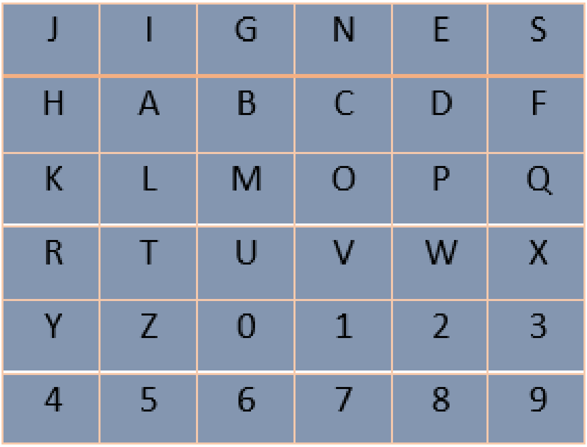
WORKING EXAMPLE

Encryption

Step 1: Creating a Playfair matrix using the key.

Key: JIGNESH

The Playfair matrix according to the given key can be constructed as below:

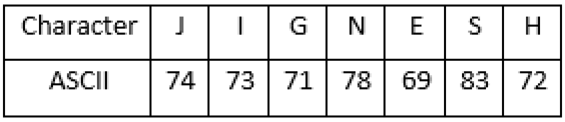


J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Step 2: Calculate the pairing character.

The pairing character can be calculated by summing up the ascii values of each character in the key take the one's digit of the sum obtained as the pairing character

For Example:



Character	J	I	G	N	E	S	H
ASCII	74	73	71	78	69	83	72

Sum=74 + 73 + 71 + 78 +69 +83 +72

Sum=520

Pairing Character= Sum MOD 10

Pairing Character= 520 MOD 10

Pairing Character= 0

Step 3: Split the plaintext into pairs.

PT: HEXXXG

Pair 1: HE

Pair 2: X0

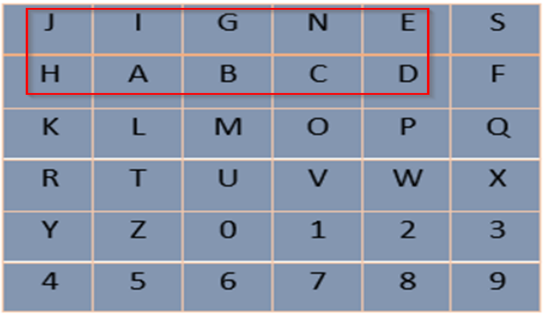
Pair 3: X0

Pair 4: XG

We can clearly see that none of the pairs are violating any of the pairing rules of the Playfair technique.

Step 4: Using the plaintext pairs generated above to generate pairs of ciphertext.

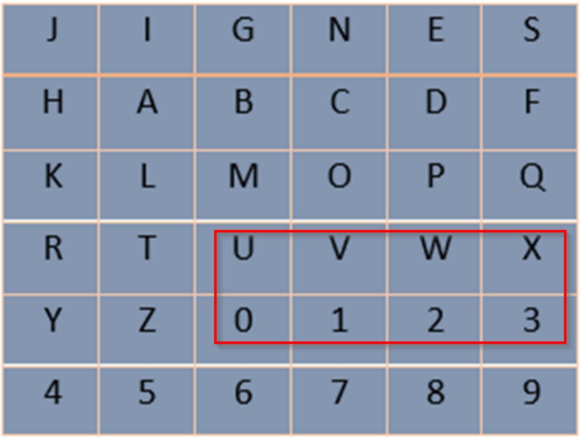
Pair 1: HE



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 1 CT: DJ

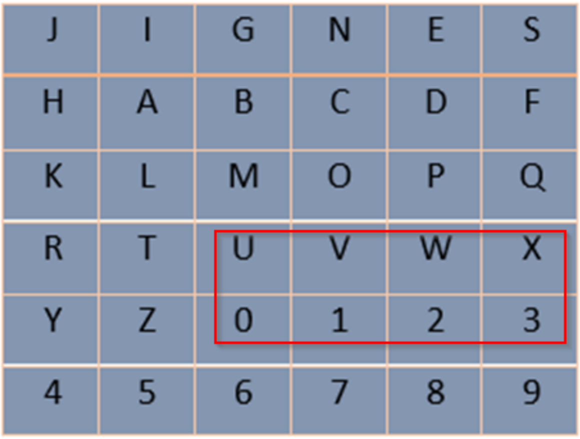
Pair 2: X0



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 2 CT: U3

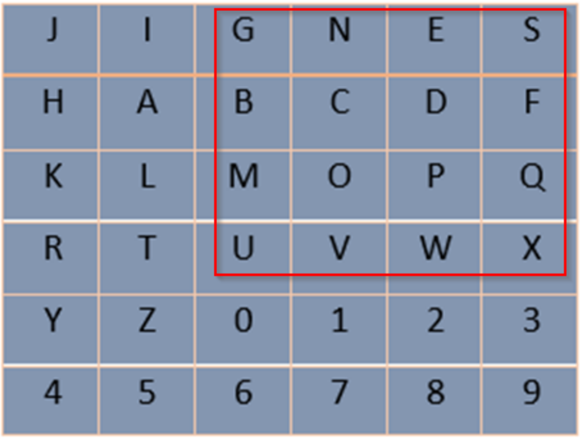
Pair 3: X0



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 3 CT: U3

Pair 4: XG



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 4 CT: US

Step 5: Combining all the cipher text pairs together to generate ciphertext.

CT=Pair 1 + Pair 2 + Pair 3 + Pair 4

CT= DJ + U3 + U3 + US

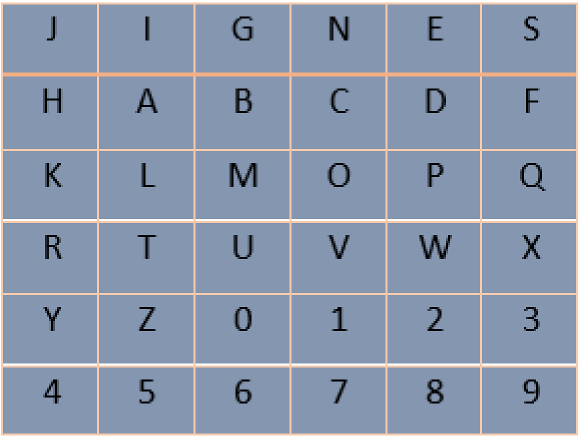
CT= DJU3U3US

Decryption

Step 1: Creating a Playfair matrix using the key.

Key: JIGNESH

The Playfair matrix according to the given key can be constructed as below:



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Step 2: Calculate the pairing character.

The pairing character can be calculated by summing up the ascii values of each character in the key take the one's digit of the sum obtained as the pairing character

For Example:

Character	J	I	G	N	E	S	H
ASCII	74	73	71	78	69	83	72

Sum=74 + 73 + 71 + 78 +69 +83 +72

Sum=520

Pairing Character= Sum MOD 10

Pairing Character= 520 MOD 10

Pairing Character= 0

Step 3: Split the ciphertext into pairs.

CT: DJU3U3US

Pair 1: DJ

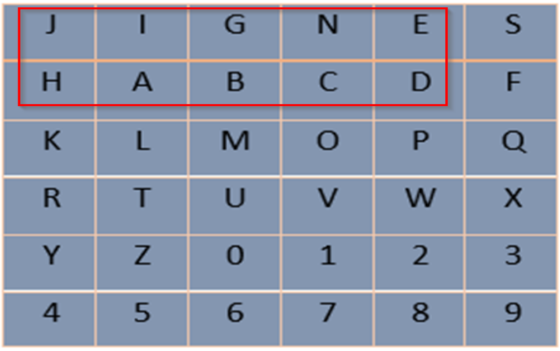
Pair 2: U3

Pair 3: U3

Pair 4: US

Step 4: Using the ciphertext pairs generated above to generate pairs of plaintext.

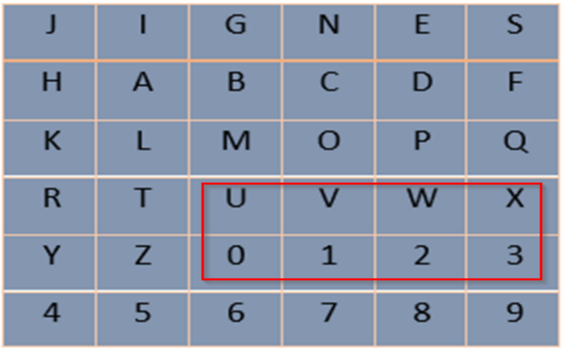
Pair 1: DJ



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 1 PT: HE

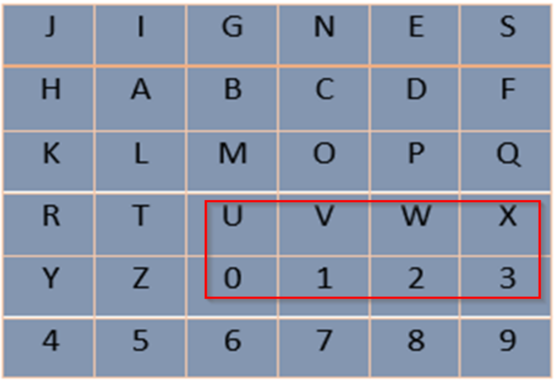
Pair 2: U3



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 2 PT: X0

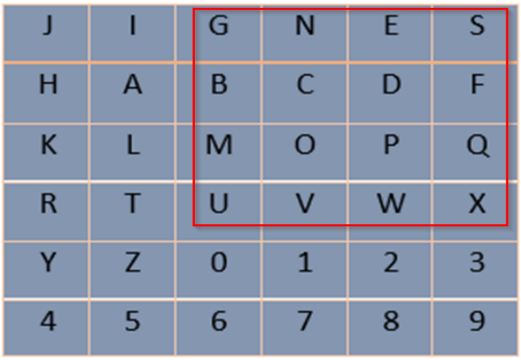
Pair 3: U3



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 3 PT: X0

Pair 4: US



J	I	G	N	E	S
H	A	B	C	D	F
K	L	M	O	P	Q
R	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

Pair 4 PT: XG

Step 5: Combining all the plaintext pairs together and remove the pairing character from the result to generate plaintext.

Result=Pair 1 + Pair 2 + Pair 3 + Pair 4

Result= HE + X0 + X0 + XG

Removing all the 0's from the result to get PT

PT= HEXXXG

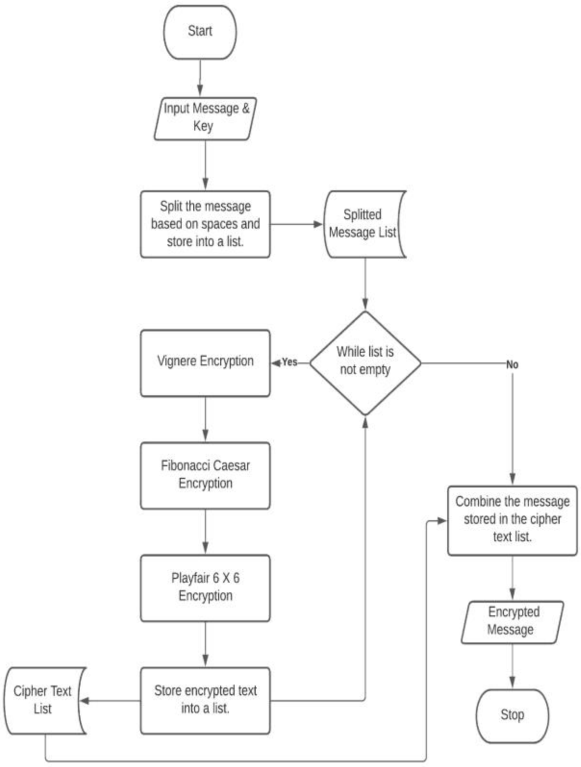
Therefore, our modified approach for Playfair cipher is able to handle all the limitations of the original Playfair encryption technique.

After overcoming the limitations of the encryption techniques discussed before, now we are ready to consolidate the algorithms that we had discussed to create a brand-new hybrid encryption and we will call it **3LFibCaesar**.

IV. WORKING

We will proceed according to the flowcharts given below for encryption as well as decryption.

A. *Encryption Flowchart:*



Start
Input Message & Key
Split the message based on spaces and store into a list.
Splitted Message List
While list is not empty
Yes
No
Vignere Encryption
Fibonacci Caesar Encryption
Playfair 6 X 6 Encryption
Store encrypted text into a list.
Cipher Text List
Combine the message stored in the cipher text list.
Encrypted Message
Stop

Let us walk through the steps mentioned in the flowchart using a simple example:

Step 1: Input message to be encrypted and the key used to encrypt it.

Message: MANAV RACHNA

Key: MRIIRS

As Fibonacci Caesar does not require a key, this key will be used for Vigenère and Playfair 6 X 6 encryption.

Step 2: Split the message based on spaces and store into a list.

Split 1: MANAV

Split 2: RACHNA

List: [“MANAV”, ”RACHNA”]

Step 3: Iterate on the split list and encrypt each word in the order as shown in the flowchart.

Vigenère Encryption

For 0th List Element:

Text: MANAV

Key: MRIIRS

Cipher Text Generated: YRVIM

For 1st List Element:

Text: RACHNA

Key: MRIIRS

Cipher Text Generated: DRKPES

Step 4: Encrypt each cipher text generated in the previous step using Fibonacci Caesar encryption.

Fibonacci Caesar Encryption

For 0th List Element:

Text: YRVIM

Cipher Text Generated: ZSXLR

For 1st List Element:

Text: DRKPES

Cipher Text Generated: ESMSJA

Step 3: Encrypt each cipher text generated in the previous step using Playfair 6 X 6 encryption.

Playfair 6 X 6 Encryption

For 0th List Element:

Text: ZSXLR

Key: MRIIRS

Cipher Text Generated: 1RUPIZ

For 1st List Element:

Text: ESMSJA

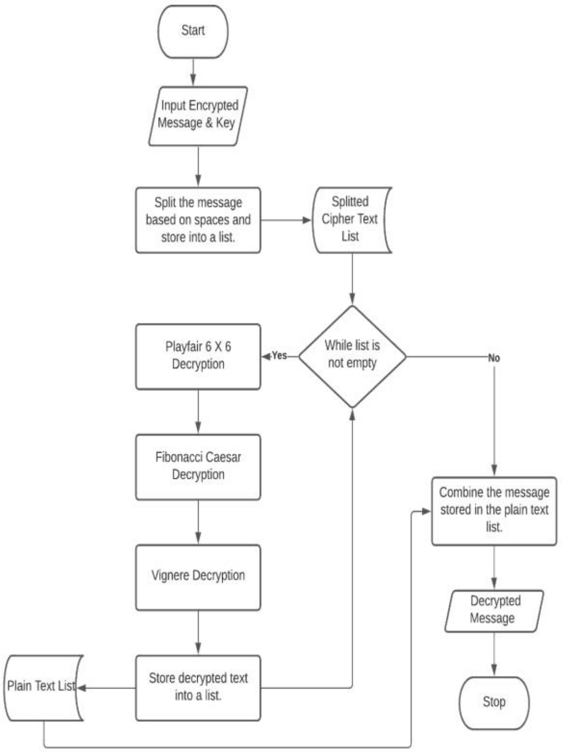
Key: MRIIRS

Cipher Text Generated: FIRAOM

Step 3: Combine the cipher texts generated in the previous step to get the final encrypted cipher text.

Cipher Text= 1RUPIZ FIRAOM

B. Decryption Flowchart:


Start
Input Encrypted Message & Key
Split the message based on spaces and store into a list.
Splitted Cipher Text List
While list is not empty
Yes
No
Playfair 6 X 6 Decryption
Fibonacci Caesar Decryption
Vignere Decryption
Store decrypted text into a list.
Plain Text List
Combine the message stored in the plain text list.
Decrypted Message
Stop

Let us walk through the steps mentioned in the flowchart using the example used in the encryption phase:

Step 1: Input message to be encrypted and the key used to encrypt it.

Encrypted Message: 1RUPIZ FIRAOM

Key: MRIIRS

As Fibonacci Caesar does not require a key, this key will be used for Vigenère and Playfair 6 X 6 encryption.

Step 2: Split the encrypted message based on spaces and store into a list.

Split 1: 1RUPIZ

Split 2: FIRAOM

List: [“1RUPIZ”, ”FIRAOM”]

Step 3: Iterate on the split list and decrypt each word in the order as shown in the flowchart.

Playfair 6 X 6 Decryption

For 0th List Element:

Encrypted Text: 1RUPIZ

Key: MRIIRS

Plain Text Generated: ZSXLR

For 1st List Element:

Encrypted Text: FIRAOM

Key: MRIIRS

Plain Text Generated: ESMSJA

Step 4: Decrypt each plain text generated in the previous step using Fibonacci Caesar decryption.

Fibonacci Caesar Decryption

For 0th List Element:

Encrypted Text: ZSXLR

Plain Text Generated: YRVIM

For 1st List Element:

Encrypted Text: ESMSJA

Plain Text Generated: DRKPES

Step 3: Decrypt each plain text generated in the previous step using Vigenère decryption.

Vigenère Encryption

For 0th List Element:

Encrypted Text: YRVIM

Key: MRIIRS

Plain Text Generated: MANAV

For 1st List Element:

Encrypted Text: DRKPES

Key: MRIIRS

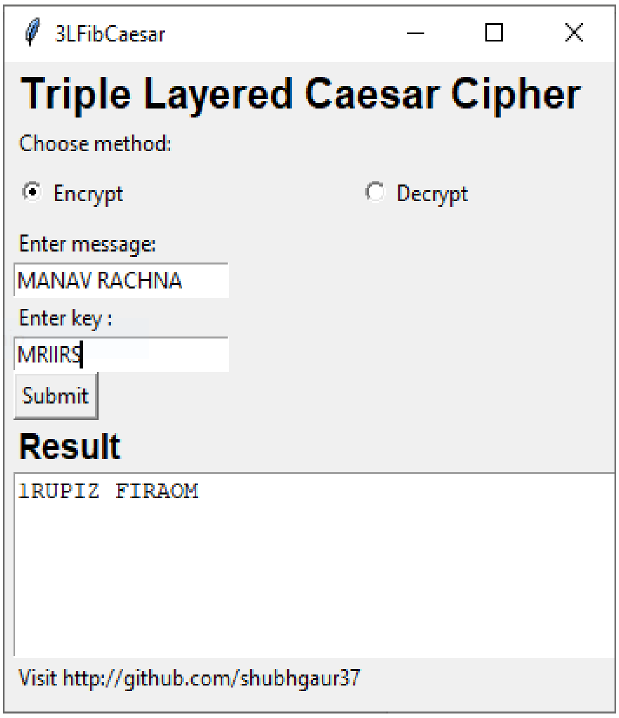
Plain Text Generated: RACHNA

Step 3: Combine the plain texts generated in the previous step to get the final decrypted plain text.

Plain Text= MANAV RACHNA

V. Implementation

Encryption



3LFibCaesar

Triple Layered Caesar Cipher

Choose method:

● Encrypt ○ Decrypt

Enter message: MANAV RACHNA

Enter key : MRIIRS

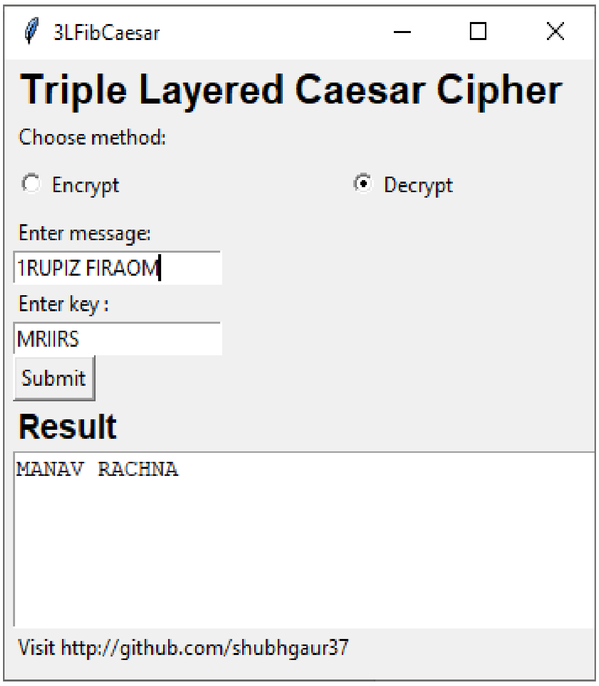
Submit

Result

```
1RUPIZ FIRAOM
```

Visit http://github.com/shubhgaur37

Decryption



3LFibCaesar

Triple Layered Caesar Cipher

Choose method:

○ Encrypt ● Decrypt

Enter message: 1RUPIZ FIRAOM

Enter key : MRIIRS

Submit

Result

```
MANAV RACHNA
```

Visit http://github.com/shubhgaur37

VI. Conclusion

Our project started from the Caesar Cipher and ended up becoming a triple layered encryption algorithm with its basic architecture being the Caesar cipher, the Playfair cipher and the Vigenère cipher. We aimed to create an encryption algorithm that would be hybrid and better suited to providing confidentiality, integrity and authenticity of the data that the user wants to encrypt.

The algorithm we created is extremely difficult to crack, albeit near impossible because of the number of permutation and combinations one would have to try to hope to crack the algorithm and get to the critical information.

Our algorithm is new and absolutely original, no one has been able to or tried to cover up the limitations of the existing encryption algorithms while turning them into a unique hybrid like this one.

VII. Future Scope

Our project aims to help people by giving them the opportunity to step into the inconceivable huge field of cryptography. This project shall be about giving people a new option to encrypt their data and store it securely.

The project shall have a huge scope of improvement, be it related to adding more cryptographic algorithms in the already existing hybrid algorithm or improvement of the UI and frontend.

It would prove to be most useful to teachers, professors, students and cybersecurity enthusiasts who aim to teach and learn the ways of cryptographic algorithms and the processes involved in it. The field of cryptography should be filled with people who are well informed and aim to improve the currently existing architectures and procedures.

Thus, our project's idea is simple but an efficient way to give back to the community with a lot of scope of improvement in the sustainable future.

ACKNOWLEDGEMENT

This research paper wouldn’t have been possible without the support of our project guide, Ms. Pronika Chawla, who helped us where we needed and acted like our guiding light. We would also like to thank the other staff at our university including the HOD, Dr. Tapas Kumar for their continuous support.

REFERENCES

[1] Han F et al (2014) A general transformation from KP-ABE to searchable encryption. Future Gener Comput Syst 30:107–115

[2] Rachmawati Dian and Candra Ade 2015 Implementation of the combination of Caesar Cipher and Affine Cipher for text data security *Informatics Research and Education Journal (JEPIN)*

[3] Ariyus D. 2008 *Introduction to Cryptography: Theory, analysis and implementation* (Yogyakarta: Andi)

[4] Basuki, Paranita and Hidayat 2016 Design of Layered Cryptography Applications Using Caesar Algorithms, Transpositions, Vigenere and Block Cipers Based on Mobile *National Seminar on Information and Multimedia Technology. STMIK AMIKOM (Yogyakarta,)*

[5] Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption: analysis of the DES modes of operation. In: Proceedings of the 38th Symposium on Foundations of Computer Science. IEEE (1997)

[6] K. Senthil, K. Prasanthi and R. Rajaram, "A modern avatar of Julius Caesar and Vigenere cipher", Computational Intelligence and Computing Research (ICCIC) 2013 IEEE International Conference, pp. 1-3, 2013.

[7] Champakamala B.S, Padmini K and Radhika D.K 2014 Least Significant Bit Algorithm for Steganography Int. J. of Advance Computer Technology

[8] Emam, Marwa M, Aly Abdelmgeid A and Omara Fatma A 2016 An Improved Image Steganography Method Based on LSB Technique with Random Pixel Selection Int.l J. of Advanced Computer Science and Applications 7 17-22

[9] Inan, Y. (2019). Analyzing the Classic Caesar Method Cryptography. 4th International Conference on Computational Mathematics and Engineering Sciences(pp. 213-220)

[10] Monika,A.,& Pradeep,M. (2012). A Comparative Survey on Symmetric Key Encryption Techniques.International Journal on Computer Science and Engineering (IJCSE), 877-882.

[11] Senthil, K.,et al. (2013). A modern avatar of Julius Caesar and Vigenere cipher. IEEE International Conference.Computational Intelligence and Computing Research (ICCIC)

[12] Verma, O.P et al. (2011). Performance Analysis Of Data Encryption Algorithms. IEEE.Delhi Technological University India

[13] Atish,J.,et al.(2015).Enhancing the Security of Caesar Cipher Substitution Method using a Randomized Approach for more Secure Communication. International Journal of Computer Applications, 129(13),6-11.DOI: 10.5120/ijca201590706

[14] Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg