Code Break

Team:
Naveen Edala - AP18110010120
Purna Sai Madala - AP18110010371
Prabhu Avula - AP18110010409
Bhavana Yedlapalli - AP18110010523
Sai Mounika - AP18110010647
Amey Sadasivuni - AP18110010420
Krishna Kusam - AP18110010415
Steve Varghese - AP18110010389
Abhiram Namburu - AP18110010200
Manan Verma - AP18110010324

Table of Contents

Acknowledgment

“We would like to thank each of our team members for their hard work and sacrifice throughout this project. We also thank our professor, Dr. Ashok Kumar Pradhan, without whose guidance, mentorship, and help, this project would not have been possible.”

- Team Code Break

Abstract

AIM:

Develop an application to break the secret key of a given secret message sent via a set of encryption algorithms.

Proposed Application:

In this fast and busy world, people and world governments prioritize security in all aspects of life. It is very important to encrypt our data before sending it to someone else. Likewise, it is also important to decrypt the data sent from the transmitter, at the receiver, to get the original message. Code Break does just that. This application is designed to decrypt a message sent by any encryption algorithm by breaking the secret key, and hence, the message itself.

Software Used:

Python

Introduction

People spend so much time of their lives on the internet. Whether it's for storing our personal information, watching entertainment, doing business transactions, or doing our jobs, our society relies increasingly on an online presence.

This increased dependence on the internet means that information security is more important than ever. The stakes are very high at the moment. Users are urged to know that their sensitive data is kept confidential, and readily available to intended readers.

Data encryption is just one weapon in cybersecurity, but it's one of the oldest and most used. And since no discussion about data encryption is complete without talking about ciphers and algorithms, let us do so. As people got better at cracking codes, the encryption had to become more sophisticated so that the messages could be kept secret, and secure.

Here are a few more reasons why we must encrypt our data when we are online:

1. Authentication: Having an encrypted public key ensures the authenticity of the information being shared online.
2. Privacy: Encrypting your data ensures its security when being sent online. This means that your information cannot be read by anyone other than you, and the intended receiver.
3. Security: Encrypted data is far more secure from falling prey to security breaches, whether the data is in transit or at either terminal point.

Now, there are several ways to encrypt your data. The main algorithms used worldwide are the Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), and RSA (Rivest-Shamir-Adleman). These are some of the most widely used and highly secure algorithms involved in encrypting data on a day to day basis.

Now that we have established the importance of data encryption, we must also understand the need and importance of data decryption. Once the encrypted data is sent from the transmitter, it arrives in its encrypted form at the receiver terminal. However, we cannot interpret the encrypted message because it looks like gibberish. This is where the process of decryption proves useful. Decryption ensures that the encrypted message is systematically deconstructed and the original message is retrieved and is ready to be interpreted.

Project Body Report

Algorithms Used:

Data Encryption Standard (DES):
The predecessor to the Advanced Encryption Standard, the DES algorithm was the benchmark for encryption standards since its introduction in 1971 till 2002. Developed by IBM researchers, it is based upon the Feistel Cipher, nicknamed LUCIFER. The algorithm takes the input plain text in blocks of 64 bits and converts them into the ciphertext using 48-bit keys. It is a symmetric key algorithm. The same key used while encrypting the plain text must also be used while decrypting the ciphertext to retrieve the original message.

Triple Data Encryption Standard (3DES):
The Triple Data Encryption Standard Algorithm is based upon the block cipher. Named after the DES algorithm, it uses the cipher technique in triplicate. That means a total of three keys are used to convert the plain text to ciphertext and vice versa. Usually, keys 1 and 3 are used to encrypt the message, whereas key 2 is used to decrypt the message.

Advanced Encryption Standard (AES):

The need for AES arose for two reasons. First, the key used in DES is small in size and hence, it was vulnerable to an exhaustive key search attack. Now, 3DES solved this drawback, however, it was too slow. AES is an iterative cipher. Based on the 'substitution-permutation' concept, it is six times faster than the 3DES method. It takes the plain text, breaks it into blocks of 128 bits, and produces the ciphertext by converting the bits using 128 bits, 192 bits, and 256 bits keys. This ensures a higher and secure mode of encryption and speed.

Rivest-Shamir-Adleman (RSA):

Considered by many as a very secure encryption method, the RSA algorithm was invented by Rivest, Shamir, and Adleman in 1978. By implementing two sets of keys (Public and Private), its encryption methods are very hard to crack. It is popular exponentiation in a finite field over integers including, but not limited to, prime numbers. The integers used here are large enough to make sure that it is difficult to crack. Its use of methods such as modulus generation, public and private key sets generation, encryption, and decryption formulas ensure that it is very complex and the ciphertext is very secure and tough to crack.

Ciphers Used:

Playfair Cipher:

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In Playfair cipher, unlike traditional cipher, we encrypt a pair of alphabets(digraphs) instead of a single alphabet. It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

Hill Cipher:

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

Feistel Cipher:

Feistel Cipher is not a specific scheme of a block cipher. It is a design model from which many different block ciphers are derived. DES is just one example of a Feistel Cipher. A cryptographic system based on a Feistel cipher structure uses the same algorithm for both encryption and decryption.

Vigenere Cipher:

The Vigenère cipher is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword. It employs a form of polyalphabetic substitution.

Block Cipher:

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

Stream Cipher:

A stream cipher is a method of encrypting text (to produce ciphertext) in which a cryptographic key and algorithm are applied to each binary digit in a data stream, one bit at a time. This method is not much used in modern cryptography.

Caesar Cipher:

The action of a Caesar cipher is to replace each plaintext letter with a different one a fixed number of places down the alphabet.

Rail Fence Cipher:

The rail fence cipher is an easy-to-apply transposition cipher that jumbles up the order of the letters of a message in a quick convenient way. It also has the security of a key to make it a little bit harder to break.
The Rail Fence cipher works by writing your message on alternate lines across the page and then reading off each line in turn.

Row Transposition Cipher:

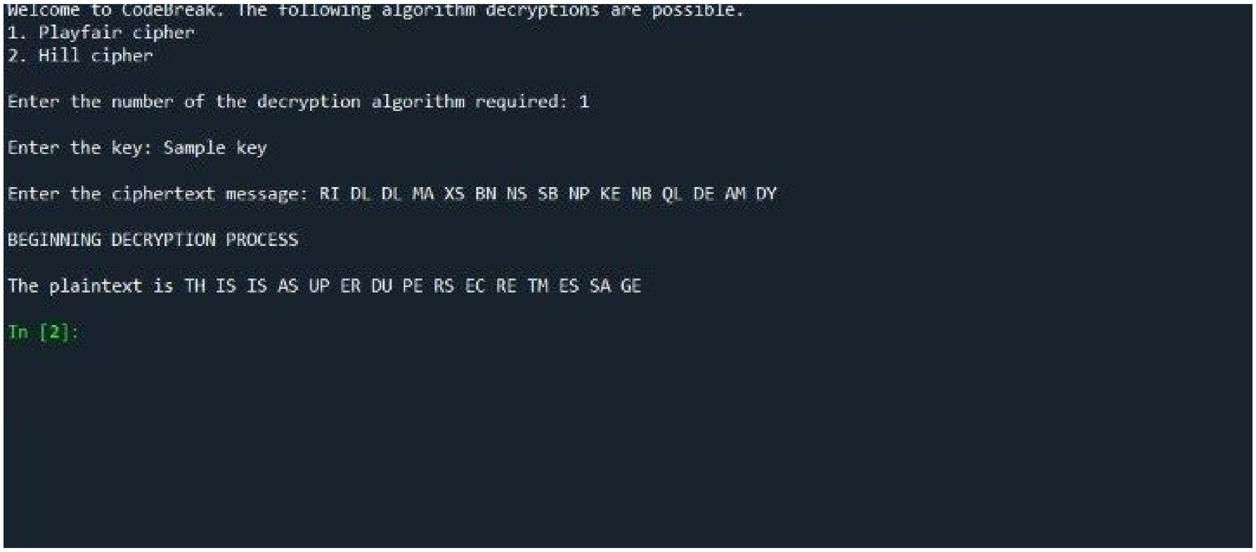
Row Transposition Cipher is a cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a ciphertext. In this process, the actual plain text alphabets are not included.

Code:

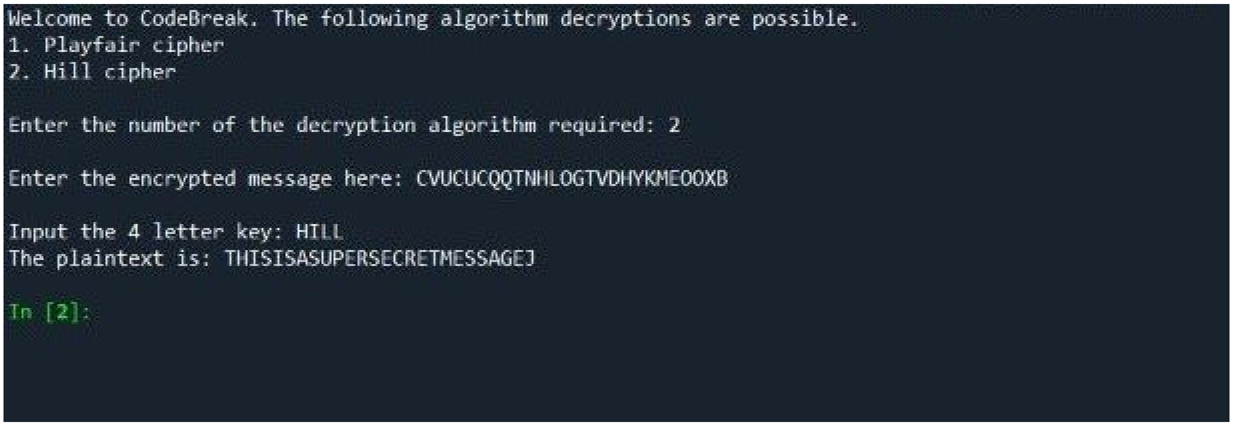
Since there are many codes we uploaded everything to this Gdrive
https://drive.google.com/drive/folders/1IutWeS7JEt-djhh0RgenKjoXRJZ_QDLB?usp=sharing
We can run main.py and use all the codes from there.

Project Evaluation

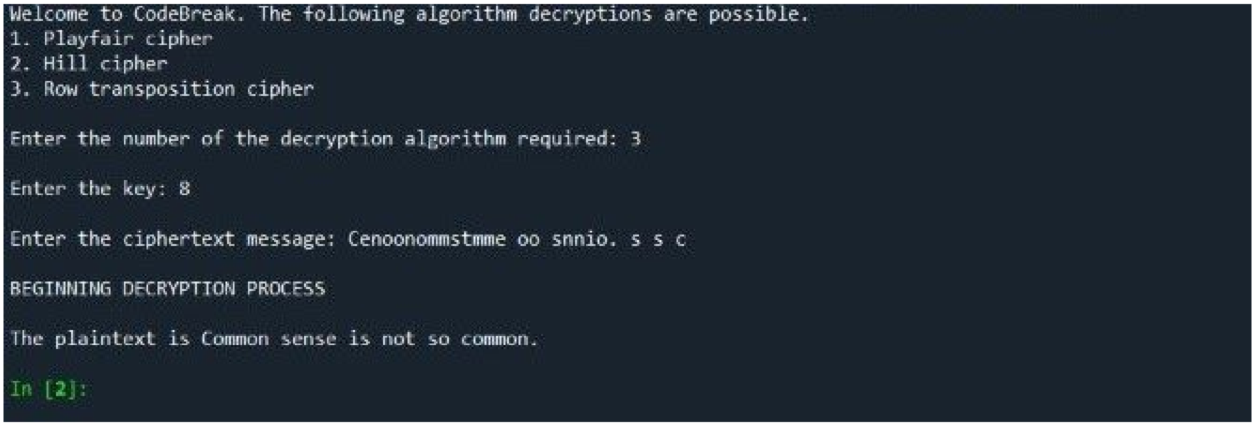
The above code was implemented and after execution, here is the output sequence:



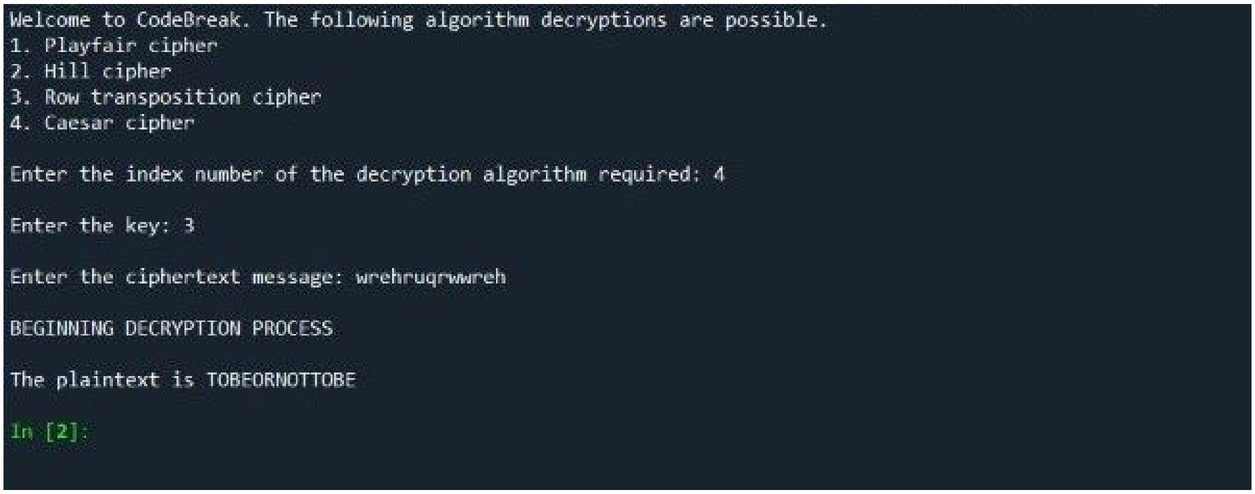
```
Welcome to CodeBreak. The following algorithm decryptions are possible.
1. Playfair cipher
2. Hill cipher

Enter the number of the decryption algorithm required: 1

Enter the key: Sample key

Enter the ciphertext message: RI DL DL MA XS BN NS SB NP KE NB QL DE AM DY

BEGINNING DECRYPTION PROCESS

The plaintext is TH IS IS AS UP ER DU PE RS EC RE TM ES SA GE

In [2]:
```



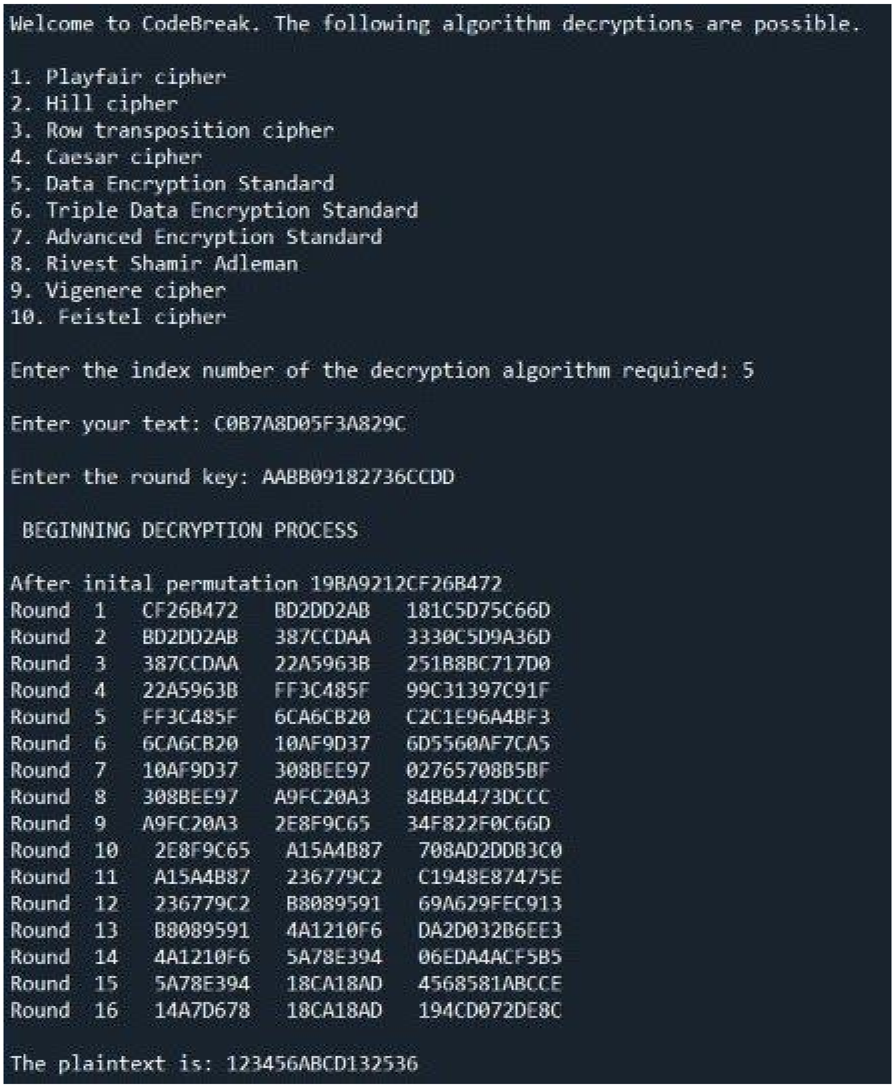
```
Welcome to CodeBreak. The following algorithm decryptions are possible.
1. Playfair cipher
2. Hill cipher

Enter the number of the decryption algorithm required: 2

Enter the encrypted message here: CVUCUCQQTNHLOGTVDHYKMEOOXB

Input the 4 letter key: HILL
The plaintext is: THISISASUPERSECRETMESSAGEJ

In [2]:
```



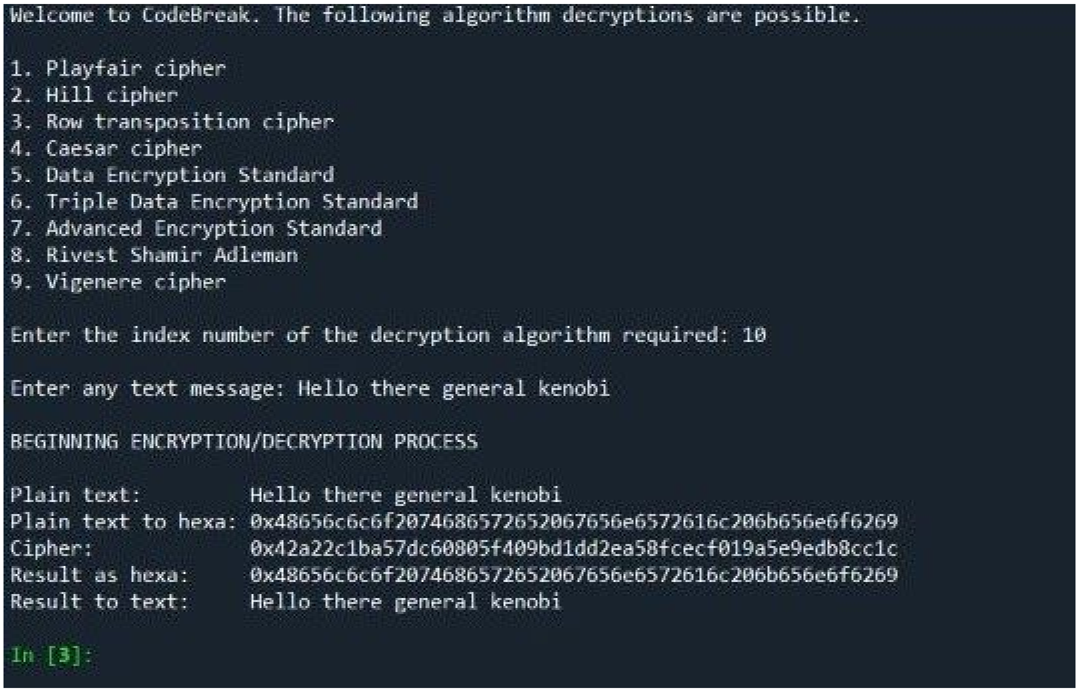
```
Welcome to CodeBreak. The following algorithm decryptions are possible.
1. Playfair cipher
2. Hill cipher
3. Row transposition cipher

Enter the number of the decryption algorithm required: 3

Enter the key: 8

Enter the ciphertext message: Cenoonommstmme oo snnio. s s c

BEGINNING DECRYPTION PROCESS

The plaintext is Common sense is not so common.

In [2]:
```



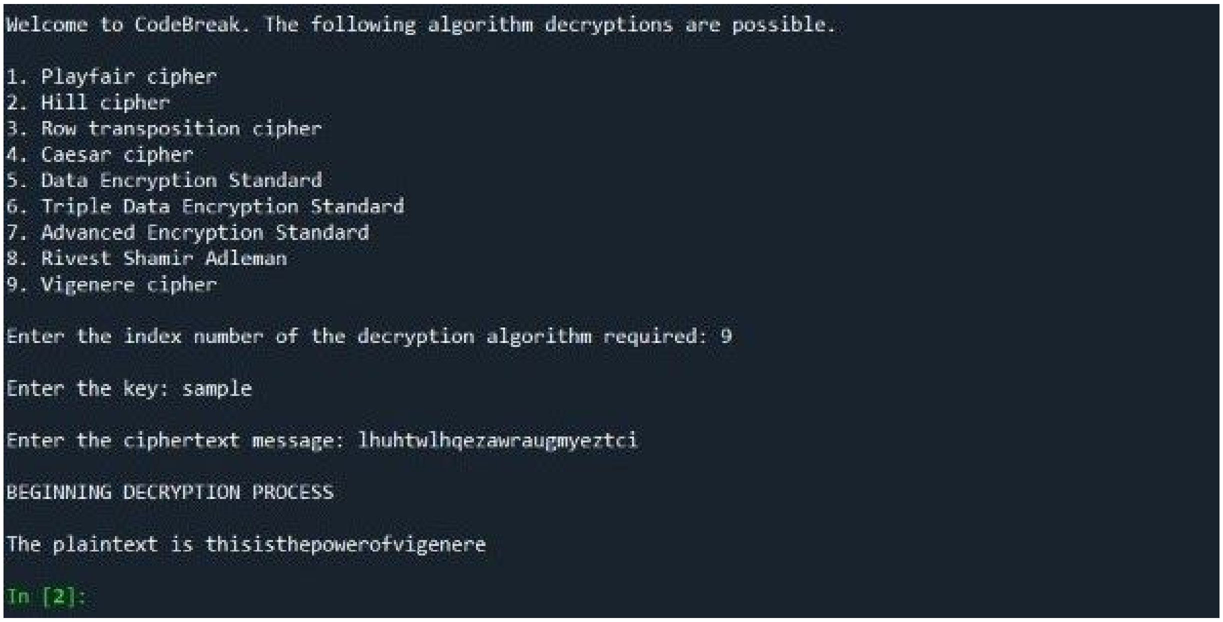
```
Welcome to CodeBreak. The following algorithm decryptions are possible.
1. Playfair cipher
2. Hill cipher
3. Row transposition cipher
4. Caesar cipher

Enter the index number of the decryption algorithm required: 4

Enter the key: 3

Enter the ciphertext message: wrehruqrwwreh

BEGINNING DECRYPTION PROCESS

The plaintext is TOBEORNOTTOBE

In [2]:
```



```
Welcome to CodeBreak. The following algorithm decryptions are possible.

1. Playfair cipher
2. Hill cipher
3. Row transposition cipher
4. Caesar cipher
5. Data Encryption Standard
6. Triple Data Encryption Standard
7. Advanced Encryption Standard
8. Rivest Shamir Adleman
9. Vigenere cipher
10. Feistel cipher

Enter the index number of the decryption algorithm required: 5

Enter your text: C0B7A8D05F3A829C

Enter the round key: AABB09182736CCDD

 BEGINNING DECRYPTION PROCESS

After inital permutation 19BA9212CF26B472
Round  1   CF26B472   BD2DD2AB   181C5D75C66D
Round  2   BD2DD2AB   387CCDAA   3330C5D9A36D
Round  3   387CCDAA   22A5963B   251B8BC717D0
Round  4   22A5963B   FF3C485F   99C31397C91F
Round  5   FF3C485F   6CA6CB20   C2C1E96A4BF3
Round  6   6CA6CB20   10AF9D37   6D5560AF7CA5
Round  7   10AF9D37   308BEE97   02765708B5BF
Round  8   308BEE97   A9FC20A3   84BB4473DCCC
Round  9   A9FC20A3   2E8F9C65   34F822F0C66D
Round  10   2E8F9C65   A15A4B87   708AD2DDB3C0
Round  11   A15A4B87   236779C2   C1948E87475E
Round  12   236779C2   B8089591   69A629FEC913
Round  13   B8089591   4A1210F6   DA2D032B6EE3
Round  14   4A1210F6   5A78E394   06EDA4ACF5B5
Round  15   5A78E394   18CA18AD   4568581ABCCE
Round  16   14A7D678   18CA18AD   194CD072DE8C

The plaintext is: 123456ABCD132536
```



```
Welcome to CodeBreak. The following algorithm decryptions are possible.

1. Playfair cipher
2. Hill cipher
3. Row transposition cipher
4. Caesar cipher
5. Data Encryption Standard
6. Triple Data Encryption Standard
7. Advanced Encryption Standard
8. Rivest Shamir Adleman
9. Vigenere cipher

Enter the index number of the decryption algorithm required: 10

Enter any text message: Hello there general kenobi

BEGINNING ENCRYPTION/DECRYPTION PROCESS

Plain text:         Hello there general kenobi
Plain text to hexa: 0x48656c6c6f207468657265206765e6572616c206b656e6f6269
Cipher:             0x42a22c1ba57dc60805f409bd1dd2ea58fcecf019a5e9edb8cc1c
Result as hexa:     0x48656c6c6f207468657265206765e6572616c206b656e6f6269
Result to text:     Hello there general kenobi

In [3]:
```



```
Welcome to CodeBreak. The following algorithm decryptions are possible.

1. Playfair cipher
2. Hill cipher
3. Row transposition cipher
4. Caesar cipher
5. Data Encryption Standard
6. Triple Data Encryption Standard
7. Advanced Encryption Standard
8. Rivest Shamir Adleman
9. Vigenere cipher

Enter the index number of the decryption algorithm required: 9

Enter the key: sample

Enter the ciphertext message: lhuhtwlhqezawraugmyeztci

BEGINNING DECRYPTION PROCESS

The plaintext is thisisthepowerofvigenere

In [2]:
```

Conclusion

Our application, Code Break, helps break the known cryptography algorithms (up to the key that has a max size of 128 bit) and retrieves the original message for the receiver to use.

Bibliography

- Advanced Encryption System. (n.d.). Tutorialspoint. Retrieved November 29, 2020, from https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm
- Data Encryption Standard. (n.d.). Simpli Learn. Retrieved November 29, 2020, from https://www.simplilearn.com/what-is-des-article
- Data Encryption Standard. (2013). Tutorialspoint. https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm
- Data Encryption Techniques. (n.d.). Simpli Learn. Retrieved November 29, 2020, from https://www.simplilearn.com/data-encryption-methods-article
- Triple DES. (n.d.). Tutorialspoint. Retrieved November 29, 2020, from https://www.tutorialspoint.com/cryptography/triple_des.htm
- Ceaser Cipher. (n.d.). Practical Cryptography. Retrieved November 29, 2020, from http://practicalcryptography.com/ciphers/caesar-cipher/
- Monoalphabetic Substitution Ciphers. (n.d.). Crypto Corner. Retrieved November 29, 2020, from https://crypto.interactive-maths.com/monoalphabetic-substitution-ciphers.html
- Playfair Cipher with Examples. (n.d.). GreeksforGreeks. Retrieved November 29, 2020, from https://www.geeksforgeeks.org/playfair-cipher-with-examples/
- Rail Fence Cipher. (n.d.). Crypto Corner. Retrieved November 29, 2020, from https://crypto.interactive-maths.com/rail-fence-cipher.html
- Types of Cipher. (n.d.). EDUCBA. Retrieved November 29, 2020, from https://www.educba.com/types-of-cipher/
- Vigenere Cipher. (n.d.). GreeksforGreeks. Retrieved November 29, 2020, from https://www.geeksforgeeks.org/vigenere-cipher/
- What is a Block Cipher? (n.d.). Wolf SSL. Retrieved November 29, 2020, from https://www.wolfssl.com/what-is-a-block-cipher/#:~:text=A%20block%20cipher%20is%20an,%2C%20192%2C%20or%20256%20bits

- Transposition cipher. (n.d.). Britannica. Retrieved November 29, 2020, from https://www.britannica.com/topic/transposition-cipher
- stream cipher. (n.d.). Search Security. Retrieved November 29, 2020, from https://searchsecurity.techtarget.com/definition/stream-cipher#:~:text=A%20stream%20cipher%20is%20a,much%20used%20in%20modern%20cryptography
- Hill Cipher. (n.d.). Geeks for Geeks. Retrieved November 29, 2020, from https://www.geeksforgeeks.org/hill-cipher/
- Feistel Block Cipher. (n.d.). Tutorials Point. Retrieved November 29, 2020, from https://www.tutorialspoint.com/cryptography/feistel_block_cipher.htm
- Difference between Block Cipher and Stream Cipher. (n.d.). Geeks For Geeks. Retrieved November 29, 2020, from https://www.geeksforgeeks.org/difference-between-block-cipher-and-stream-cipher/