**Abstract**

When you hear the word "encryption," you might think about modern computers and things like email and online bank accounts. But did you know that encryption has been around for thousands of years? This project is about the Caesar cipher, a simple type of encryption that replaces each letter of the alphabet with another letter, and demonstrate how a modern computer can crack this ancient code in just a few seconds.

**Introduction**

Back in the days of Julius Caesar there was a need for secrecy. People were constantly challenging Rome’s power, which meant there were a lot of battles to be fought. The Roman army needed to be able to communicate with each other to make sure everyone was on the same page. Now if the enemy intercepted these messages, they would know exactly what the Roman’s plans were. This is why a way to send secret messages was important. Julius Caesar came up with a way to do this. Caesar would give his generals a certain number, or cipher, before they left for battle. Then when a message would need to be sent each letter in the message would then be incremented by that number to a new letter. So if the cipher was 2, and the current letter was originally A, then the encrypted letter would be C. Therefore if the encrypted message was intercepted, it would look like a bunch of mixed letters than meant nothing. Since the general receiving the message had the cipher, he could then receive the message, and decrement the letter by the cipher to get the original message back. This is just the general way Caesar’s cipher was used. There have been reports of the encrypted message taking each corner letter and so on to encrypt it, as well as many other ways to encrypt a message. These basic encryption methods are the foundation of cryptography and are the first things they study. However in this report we are using the basic form of Caesar’s Cipher and using the cipher to transform our messages.

**Technical Approach**

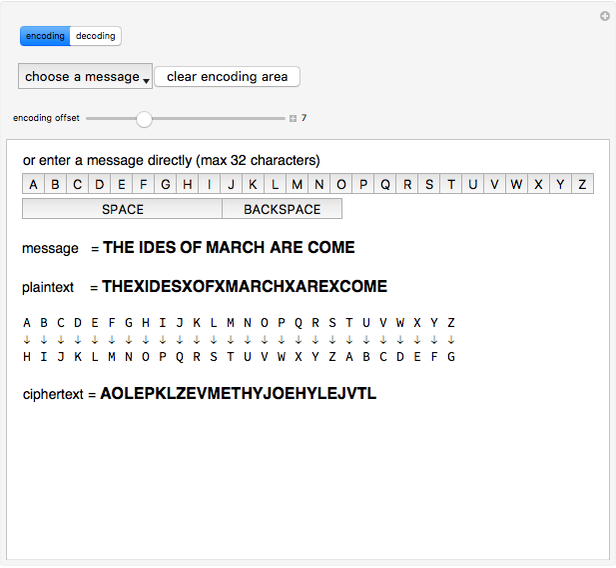
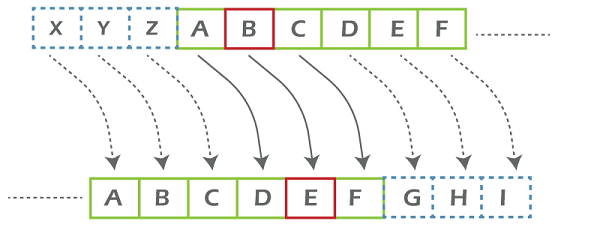
A cipher is a method that encrypts or disguises text. The undisguised text is called “plaintext”, and the disguised text is called “cipher text”. The process of transforming plaintext to cipher text is called “encryption”. The reverse is called “decryption”. In the above scheme, each letter of the cipher text substitutes for a letter of the plaintext, this is called “substitution cipher”. Mathematically, Caesar’s cipher can be described as follows:

message letter + 3 places = encrypted (disguised) letter encrypted (disguised) letter – 3 places = message letter

In this project, we generalize Caesar’s cipher, making it flexible enough to handle any number of shifts between 1 and 25, that is

message letter +*r*places = encrypted (disguised) letter encrypted (disguised) letter –*r*places = message letter

where 1<= r <=25.



**IMPLEMENTATION AND TESTING**

The implementation of the generalized Caesar’s cipher consists of three components, including a function that randomly generates an integer between 1 and 25 used as the shift, the encryption function that encrypts each letter of an input message according to the specified shift, and the decryption function that recovers the encrypted letter back to its original format. We name the random number generation function as myRand(), which is based on the rand() function provided by the system. myRand() can control the range of the random number generated. It takes two integer inputs, indicating the lower bound and the upper bound of the random integer, respectively. Suppose the upper bound is ub, and the lower bound is lb, then the following statement is used

lb + rand() % (ub – lb + 1)

One more detail has to be taken care of such that when every time running the function, it will indeed generate a different number. This would need the srand() function to be called before calling the rand() function, in which srand()randomize the seed.

The biggest challenge in implementing either the encryption function or the decryption function is how to handle the overflow situation. For example, if the shift is 3and we need to encrypt the letter ‘Y’, it will become ‘\’ according to the ASCII table if no action is taken to deal with the overflow case. An overflow case occurs when after the shifting, the resulting encrypted or decrypted letter falls out of the range of 26characters. We use the following approach to handle overflow.

Suppose the letter to be encrypted is saved in variable “letter”, then the resulting encrypted letter “eletter” using “

shift” would be eletter = ‘A’ + (letter + shift – ‘A’) % 26

If the input letter is lower case, similar equation applies except that the ‘A’s should be changed to ‘a’s.

**Objective of the project**

**Why we choose this project?**

The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down.   
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically.

Hardware requirements

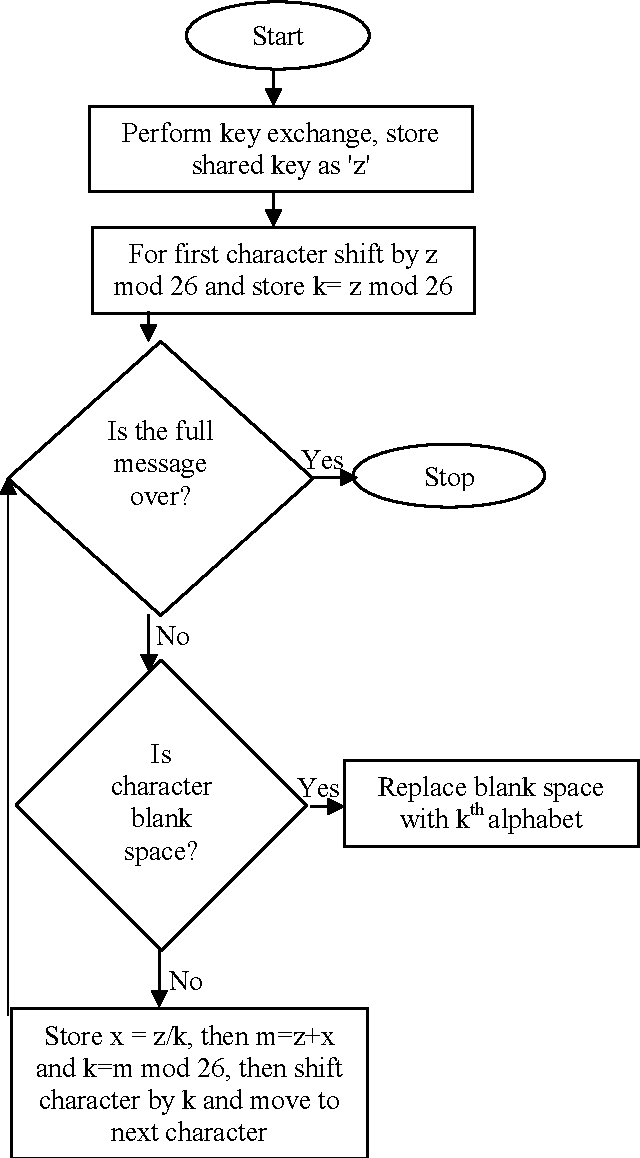
1. Computer or

Laptop

Software requirements

1. Pycharm
2. Pydroid
3. Python IDE
4. Cryptool

**Data flow diagram**



1. Shift the entire alphabet by the number you picked and write it down below your original alphabet (as shown above).
2. Pick a message to write to your friend. ...
3. Write down your encoded message using your shifted alphabet. ...
4. Give your friend the encoded message and tell them the key.

**Conclusion**

Since, we have shown that Caesar cipher being one of the simplest and widely used encryption techniques can be fortified beyond what common Caesar cipher algorithm can achieve. Many methods are used for Security purpose based on Caesar cipher algorithms. In our future work we can use various types of keys in one method also we can add more algorithms to enhance the security.

**Reference**

[1]. SomdipDey, JoyshreeNath and AshokeNath, “An Advanced Combined Symmetric Key Cryptographic Method using Bit Manipulation, Bit Reversal, Modified Caesar Cipher (SD-REE), DJSA method, TTJSA method: SJA-I Algorithm”, International Journal of Computer Applications (IJCA).

[2].“ENCRYPTION”,http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption\_perf/index.htm.

**Thank you…**