# Vigenère cipher

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

The Vigenère (French pronunciation: ​[viʒnɛːʁ]) cipher has been reinvented many times. The method was originally described by Giovan Battista Bellaso in his 1553 book La cifra del. Sig. Giovan Battista Bellaso; however, the scheme was later misattributed to Blaise de Vigenère in the 19th century, and is now widely known as the "Vigenère cipher.

Though the cipher is easy to understand and implement, for three centuries it resisted all attempts to break it; this earned it the description le chiffre indéchiffrable (French for 'the indecipherable cipher'). Many people have tried to implement encryption schemes that are essentially Vigenère ciphers. Friedrich Kasiski was the first to publish a general method of deciphering a Vigenère cipher.

One of the main problems with simple substitution ciphers is that they are so vulnerable to frequency analysis. Given a sufficiently large cipher text, it can easily be broken by mapping the frequency of its letters to the know frequencies of, say, English text. Therefore, to make ciphers more secure, cryptographers have long been interested in developing enciphering techniques that are immune to frequency analysis. One of the most common approaches is to suppress the normal frequency data by using more than one alphabet to encrypt the message. A polyalphabetic substitution cipher involves the use of two or more cipher alphabets. Instead of there being a one-to-one relationship between each letter and its substitute, there is a one-to-many relationship between each letter and its substitutes.

**Vigenère Cipher Algorithm:**

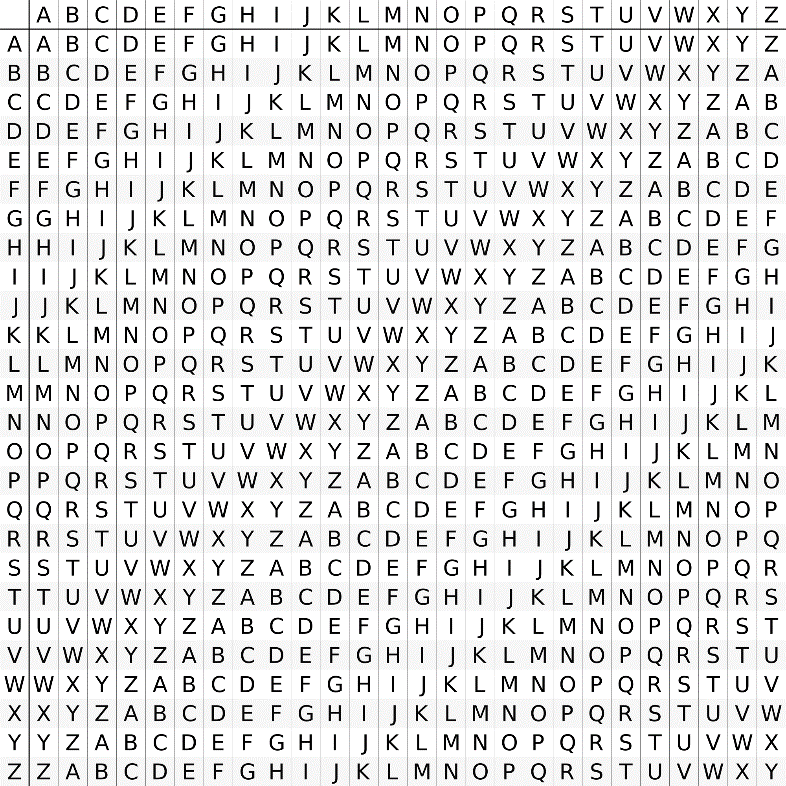
Vigenère can also be viewed algebraically. If the letters A–Z are taken to be the numbers 0–25, and addition is performed modulo 26, then Vigenère encryption E using the key K can be written,

C_i = E_K(M_i) = (M_i+K_i) \mod {26}

And decryption D using the keyK,

M_i = D_K(C_i) = (C_i-K_i) \mod {26},

Where M = M_1 \dots M_n is the message, C = C_1 \dots C_n is the cipher text and K = K_1 \dots K_n is the key obtained by repeating the keyword \lceil n / m \rceil times, where mis the keyword length.



The Vigenère square or Vigenère table, also known as the tabula recta, can be used for encryption and decryption.

**Source Code:** Language used is Python and framework used is webpy

**Python code:**

import web

urls = (

"/", "Encrypt",

"/decryption", "Decrypt",

"/favicon.ico","icon",

)

render = web.template.render('templates')

app = web.application(urls, globals())

def ptoc(i,j):

num1 = ord(i)-97

num2 = ord(j)-97

return(chr(((num1+num2)%26)+97))

def ctop(i,j):

num1 = ord(i)-97

num2 = ord(j)-97

return(chr(((num1-num2)%26)+97))

def vigencrypt(plain,k):

plaintext = plain.lower()

key = k.lower()

plaintext = plaintext.replace(' ','')

key = key.replace(' ','')

cipher = ''

j = 0

keylen = len(key)

for i in plaintext:

if(j>keylen-1):

j= 0

cipher = cipher + ptoc(i,key[j])

j = j+1

return cipher.upper()

def vigendecrypt(cipher,k):

ciphertext = cipher.lower()

key = k.lower()

ciphertext = ciphertext.replace(' ','')

key = key.replace(' ','')

plain = ''

j = 0

keylen = len(key)

for i in ciphertext:

if(j>keylen-1):

j= 0

plain = plain + ctop(i,key[j])

j = j+1

return plain

class Encrypt:

def GET(self):

return render.encryption()

def POST(self):

pk = web.input(plaintext='',key='')

if (pk.plaintext=='') or (pk.key==''):

return render.encryption(error='You cannot leave any field empty')

return render.encryption(error='',ciphertext = vigencrypt(pk.plaintext,pk.key))

class Decrypt:

def GET(self):

return render.decryption()

def POST(self):

ck = web.input(ciphertext='',key='')

if (ck.ciphertext=='') or (ck.key==''):

return render.decryption(error='You cannot leave any field empty')

return render.decryption(error='',plaintext = vigendecrypt(ck.ciphertext,ck.key))

class icon:

def GET(self): raise web.seeother("/static/favicon.ico")

if \_\_name\_\_ == "\_\_main\_\_":

app.run()

**HTML Templates:**

***Encryption Template***

$def with(error='',ciphertext='')

<html>

<head>

<title>Vigenere Ciphere</title>

<link href="static/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<h1>VIGENERE CIPHER</h1>

<blockquote>The Vigenère cipher is a method of encrypting alphabetic text by using a series of different Caesar ciphers based on the letters of a keyword. It is a simple form of polyalphabetic substitution</blockquote>

<h2>Encryption</h2>

<form method="POST" role="form" class="form-horizontal">

<div class="form-group">

<label class="control-label col-sm-2" for="usr">Plain Text:</label>

<div class="col-sm-10">

<input type="text" class="form-control" placeholder="Enter Plain Text" name="plaintext">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="usr">Key:</label>

<div class="col-sm-10">

<input type="text" class="form-control" placeholder="Enter Key" name="key">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<input type="submit" class="btn btn-success" value="Encrypt">

</div>

</div>

</form>

$if error:

<div class="alert alert-danger">$error</div>

$if ciphertext:

<div class="alert alert-info">Cipher text : <strong style="color:black;"> $ciphertext</strong> </div>

<hr>

<a href="/decryption">Vigenere Ciphere Decryption Click Here</a>

</body>

</html>

***Decryption Template***

$def with (error='',plaintext='')

<html>

<head>

<title>Vigenere Ciphere</title>

<link href="static/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<h1>VIGENERE CIPHER</h1>

<blockquote>The Vigenère cipher is a method of encrypting alphabetic text by using a series of different Caesar ciphers based on the letters of a keyword. It is a simple form of polyalphabetic substitution</blockquote>

<h2>Decryption</h2>

<form method="POST" role="form" class="form-horizontal">

<div class="form-group">

<label class="control-label col-sm-2" for="usr">Cipher Text:</label>

<div class="col-sm-10">

<input type="text" class="form-control" placeholder="Enter Cipher Text" name="ciphertext">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="usr">Key:</label>

<div class="col-sm-10">

<input type="text" class="form-control" placeholder="Enter Key" name="key">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<input type="submit" class="btn btn-success" value="Decrypt">

</div>

</div>

</form>

$if error:

<div class="alert alert-danger">$error</div>

$if plaintext:

<div class="alert alert-info">Plain Text : <strong style="color:black;"> $plaintext</strong></div>

<hr>

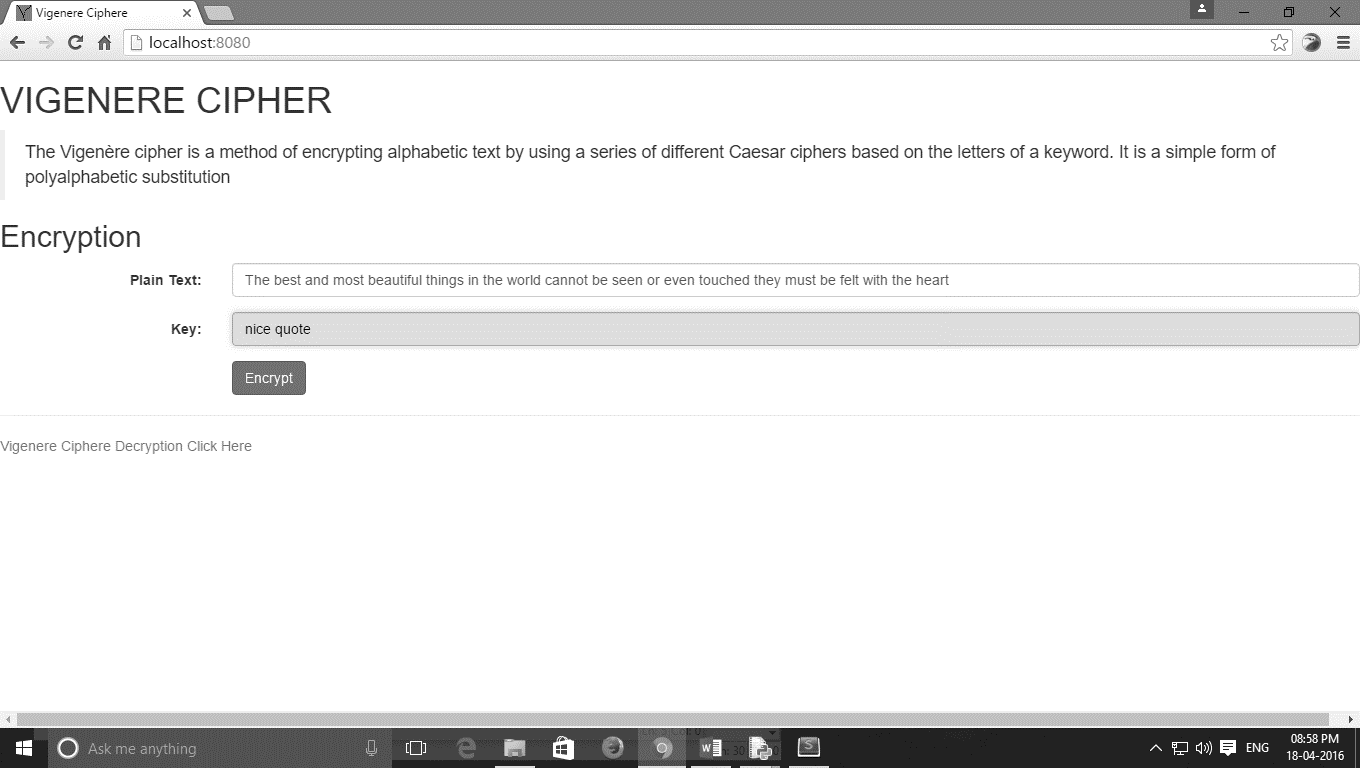
<a href="/">Vigenere Ciphere Encryption Click Here</a>

</body>

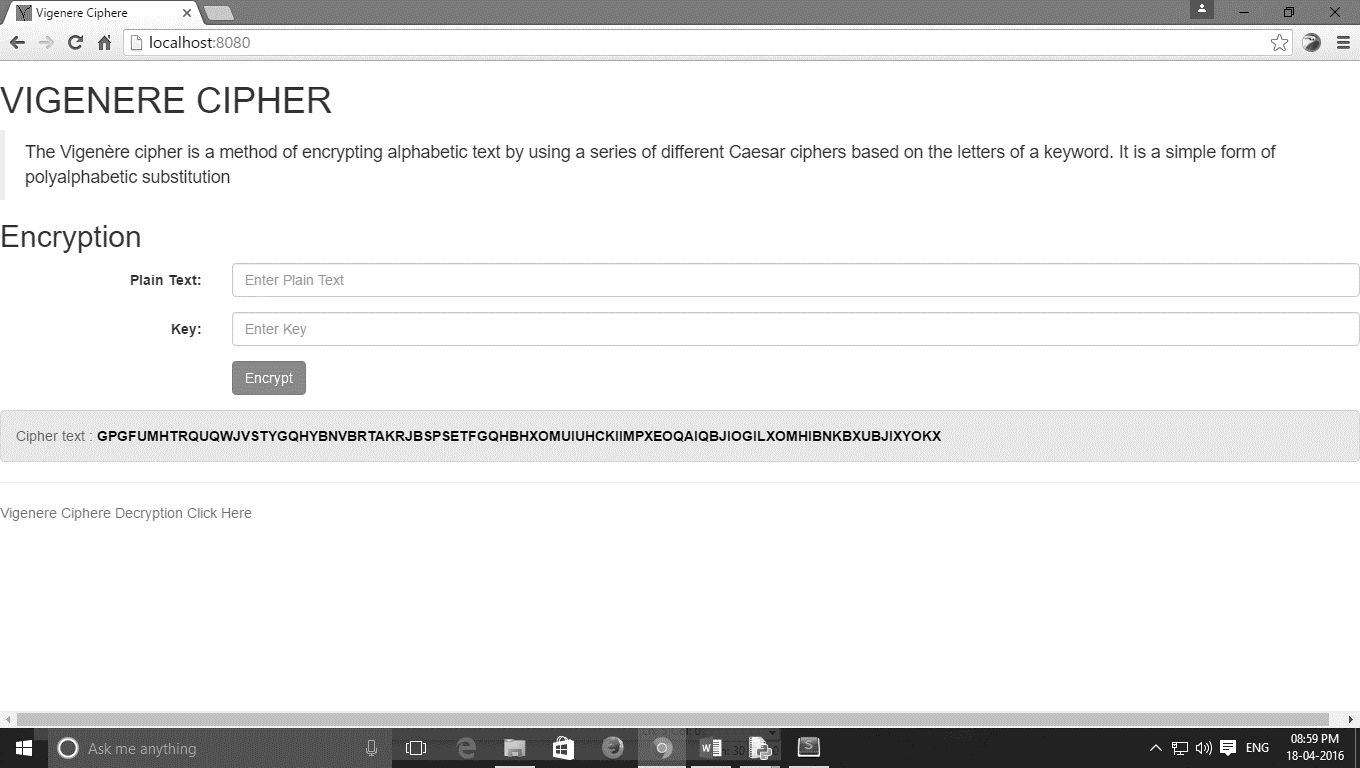
</html>

**Screenshots of the working program:**

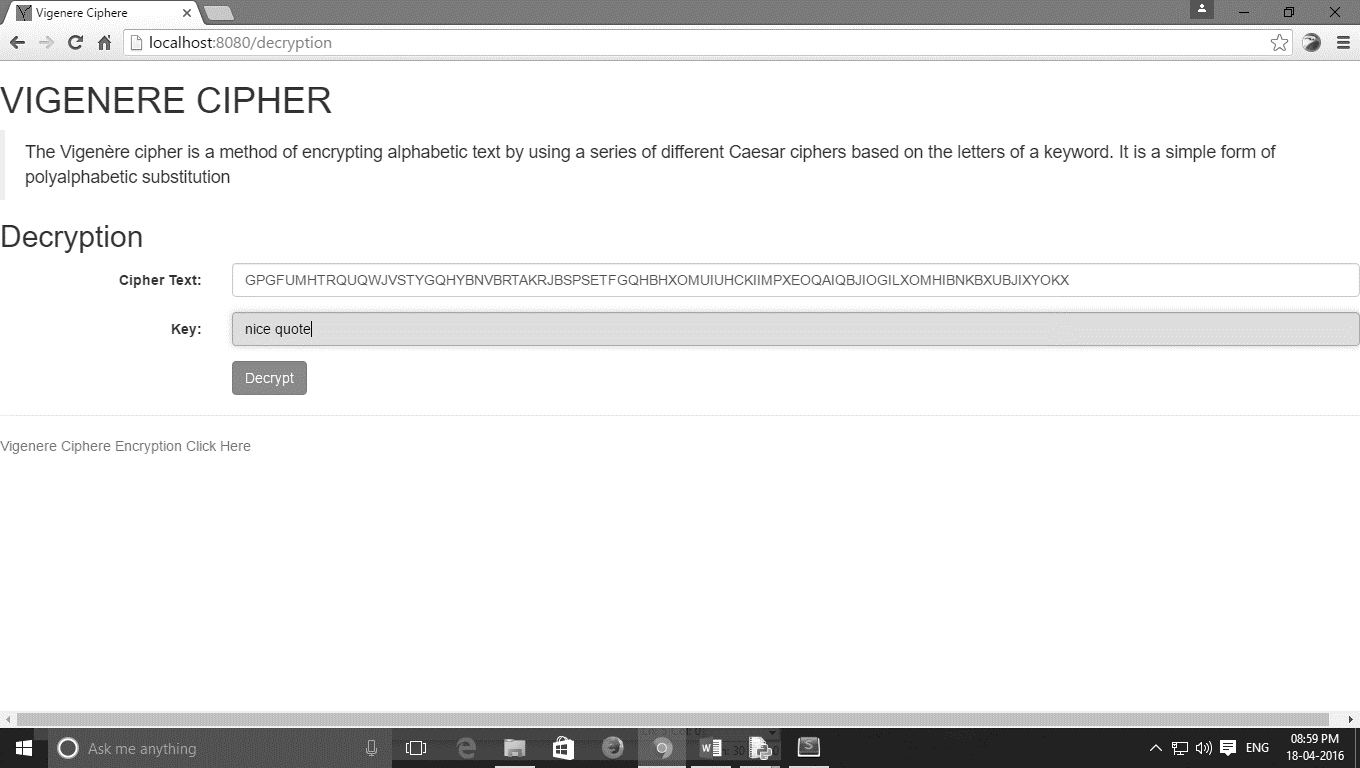
**Encryption**



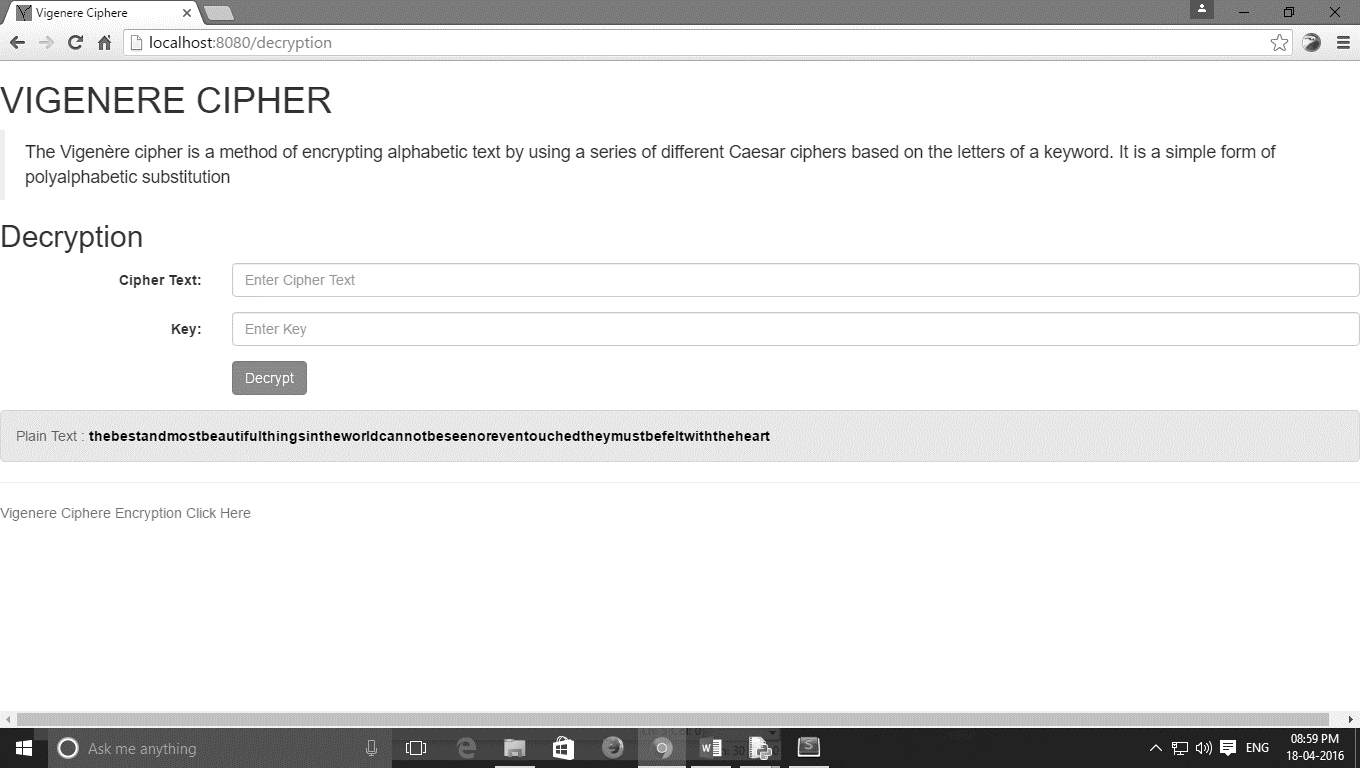
On clicking the “Encrypt” button…



**Decryption**



On clicking the “Decrypt” button…



We get back the original plain text back!