Vigenère Cipher

Polyalphabetic Substitution Cipher

Keane J. Moraes

History

The polyalphabetic cipher is arguably the brainchild of Italian architect **Leon Battista Alberti**. In his paper, *De Cifris* in 1466, he describes his cipher - the Alberti cipher that uses two copper disks, the outer one being stationary and the inner one being mobile.[1]Each disk had 24 equal cells containing letters of the old alphabet. By picking a letter on the movable disk to serve as an index key, we align it with the first letter of the plain text on the stationary disk and encode using this index key. Every 3 to 4 words, we change the index key. This cipher is **immune** to brute force attacks and frequency analysis.

Johannes Trithemius expanded this 'species' of cipher by his *tabula recta* in his post-mortem publication *Polygraphiae libri sex* in 1508. He used a square table (26x26) with letters of the alphabet as shown below.

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

David Kahn, in his book *The Codebreakers*, noted that "The great advantage of this procedure over Alberti's is that a new alphabet is brought into play with each letter. Alberti shifted alphabets only after three or four words.[1]""

Before Vigenère, a man named **Giovan Battista Bellaso**, who served as secretary in the suite of Cardinal Rodolfo Pio di Carpi publicized his booklet in 1553 named *La cifra del. Sig. Giovan Bastista Belaso.*, where he suggested the idea of using an easily changed 'key' - which he called a countersign. The key letter that is used to encrypt the corresponding plain text will be the column value of the tableau and the plain text letter will be the row value of the tableau. This melded the best of both worlds of its predecessors - the mixed alphabet of Alberti and the letter-by-letter encipherment of Trithemius.[1]. Whereas Alberti and Trithemius used a fixed pattern of substitutions, Bellaso's scheme meant the pattern of substitutions could be easily changed, simply by selecting a new key.

Blaise de Vigenère was a diplomat and translator who worked in Rome in 1549 where he read the works of Trithemius, Belaso and Alberti's manuscripts. In 1585, *Traicte des Chiffres* described his autokey cipher which used the plain text as the key along with a priming keyword. However, in the 19th century, the **invention of Bellaso's cipher was misattributed to Vigenère**. This is how the modern Vigenère cipher came to be misnamed. However, the 'Vigenère' cipher was considered to be unbreakable for nearly 300 years. It was dubbed the '*le chiffre indéchiffrable*' by the French.[2] It was not until 1863 when Kasiski published the first method of cracking the cipher.

[1]Kahn, David (1999). "On the Origin of a Species". The Codebreakers: The Story of Secret Writing. Simon & Schuster. ISBN 0-684-83130-9.

[2]Singh, Simon (1999). "The Anonymous Codebreaker". The Code Book (Teenage Version). Anchor Books, Random House. pp. 63–78. ISBN 0-385-49532-3.

Encryption and Decryption Algorithms

In both encryption and decryption, we will take the key to be in uppercase. This is because we need the physical letter number of the key at a certain itre position to act as the shift regardless of how we convert it into that standard format (0-25). As long as we convert the ASCII character into standard format, the case of the key is impertinent.

Encryption

Encryption of the plain text is done by shifting each letter of the plain text to the right by the corresponding letter number of the given key at that same itre position. Any numbers that exceed 26 are wrapped around. We will perform a letter-by-letter encryption.
We must take into account the conversion of our lowercase character $\big(a(97)-z(122)\big)$ or uppercase$\big(A(65)-Z(90)\big)$ into the standard $(0-25)$ format else the modular arithmetic will not work with a modulo of 26. Additionally, we must convert the key from uppercase ASCII to standard format.
Let E_t be the cipher text
P_t be the plain text having length 'n'
K_t be the key
$E_t = E_1E_2E_3 \dots E_n$ where E_i is the i^{th} character of the cipher text
$K_t = K_1K_2K_3 \dots K_m$ where m is the length of the key
i is the **itre position** of the plain text where $1 \leq i \leq n$
j is the **itre position** of the key where $1 \leq j \leq m$
The i^{th} character of the cipher text is -
If the letter is between 65 and 90 (Uppercase),

$$E_i = \Big[\big((P_i - 65) + (K_j - 65)\big) \mod 26\Big] + 65$$

If the letter is between 97 and 122 (Lowercase)

$$E_i = \Big[\big((P_i - 97) + (K_j - 65)\big) \mod 26\Big] + 97$$

Once again, note that the ± 65 and ± 90 is simply to bring the ASCII characters down to standard format, perform the operation necessary and then raise them back up to be converted back into characters.

Now unless m is as large as n, we will have a problem because at one point in our loop the itre position will exceed m as our loop is terminated only when the itre position exceeds n. So, we need a **correction mechanism** to cycle m back to 1 as soon as it exceeds m so that the key is repeated. Once again, **modular arithmetic** comes to our rescue.

To those reading the documentation before the code, this will be left as an open problem. The note below explains the code implementation of the formula followed by a brief explanation.

NOTE
The value of m in the formula ranges between 1 to m however this would result in an OutOfBoundsException since in code, the value must be between 0 to $m-1$. So in the code, the formula would be :

```
1         j = (++j)%m;
2
```

Keep in mind that the formulas stated are to give the reader a general overview of the underlying process used in the code instead of translating written code into mathematical dictum. This is evident in the use of 1 to n or 1 to m as itre positions instead of 0 to $n-1$ or 0 to $m-1$. Hence one must take these formula simply as supplements to learning rather than as inflexible axioms.

Decryption

Decryption of the plain text is done by shifting each letter of the cipher text to the left of the corresponding letter number of the key at that same itre position. Any numbers that are below 0 are wrapped around.

Let E_t be the cipher text having length 'n'

P_t be the plain text

K_t be the key

$P_t = P_1P_2P_3 \dots P_n$ where P_i is the i^{th} character of the plain text

$K_t = K_1K_2K_3 \dots K_m$ where m is the length of the key

i is the **itre position** of the cipher text where $1 \leq i \leq n$

j is the **itre position** of the key where $1 \leq j \leq m$

The i^{th} character of the plain text is -

If the letter is between 65 and 90 (Uppercase),

$$P_i = \Big[\big((E_i - 65) - (K_j - 65)\big) \mod 26\Big] + 65$$

If the letter is between 97 and 122 (Lowercase)

$$P_i = \Big[\big((E_i - 97) - (K_j - 65)\big) \mod 26\Big] + 97$$

Variants

There are several variants of the Vigenère cipher. Although Blaise de Vigenère did not come up with the cipher that bears his name, his Autokey cipher comes under variants of the cipher.

- Autokey Cipher
- Beaufort Cipher
- Trithemius Cipher
- Beale Cipher
- Gronsfeld Cipher

The Autokey, Beaufort, Trithemius and Gronsfeld are discussed below. Running Key cipher is in a separate documentation.

Autokey Cipher

This was Vigenère's actual invention. It involves using a priming keyword concatenated with the plain text to generate a keystream. The example in the History section shows the working of this cipher. This cipher is stronger than 'Vigenère' because there is no key repetition and is immune to many of the cryptanalysis methods (i.e. Kasiski Examination) that rely on predicting the length of the key to break the cipher. By nature of being a polyalphabetic substitution cipher, it is also immune to frequency analysis.[3]Additionally, since we add the priming key to the start of the plain text, we don't need to add our correction mechansim as the key length will always be greater than the plain text length.

For eg.: Plain Text : Meet me around the corner so that we can ...

Key : Dukemeetmearoundthecornersothat ...

where 'Duke' is the priming keyword.

Beaufort Cipher

Beaufort Cipher was invented by Sir Francis Beaufort and differed from the Vigenère in the tabula recta. There is a slight change in the encryption and decryption algorithms. This is not to be confused however with the **'Variant Beaufort Cipher'** which uses the the Vigenère decryption to encrypt the plain text and the Vigenère encryption to decrypt. There also exists a German varietal of the cipher where the key is subtracted from the plain text or cipher text depending on whether you're encrypting or decrypting. The algorithms for encrypting and decrypting of the original Beaufort cipher respectively are stated below:

[3]https://crypto.interactive-maths.com/autokey-cipher.html

$$E_i = \Big[\big((K_j - 65) - (P_i - 65)\big) \mod 26\Big] + 65$$

$$P_i = \Big[\big((K_j - 65) + (E_i - 65)\big) \mod 26\Big] + 65$$

The most noteworthy application of this cipher is during WWII when the US Navy used a handheld portable multi-rotor variant of the cipher. The interested reader will find links for information on this in the Further Reading section.

Trithemius Cipher

Johannes Trithemius first came up with the idea of using a tabula recta as a tool for polyalphabetic substitution. The cipher is rather simple and does not use a key. The shift of the cipher is the letter number of the plain text itself. Thus $A(0) \rightarrow A; B(1) \rightarrow C; C(2) \rightarrow E \ldots$ The user may add an additional shift value if desired. However, this offers lesser security owing to the lack of a key. If any third-party knows that this cipher was used to encrypt the plain text, the cipher becomes very easy to crack.

Gronsfeld Cipher

The Gronsfeld cipher or the Bronckhorst cipher, was invented around 1744 by Jost Maximilian von Bronckhorst (Count of Gronsfeld, hence the two names). It uses numbers instead of letters to encrypt the text (typically 0-6 (a-g)). Like the Trithemius, it is very easy to crack if one knows that it has been used to encrypt a text. To see the cryptanalysis in action visit Cracking the Gronsfeld Cipher

Cryptanalysis

Kasiski Examination

Freidrich Kasiski published the first general successful attack against the Vigenère in 1863, nearly 300 years after the cipher was invented. His method did not rely on prior knowledge of the key length or of the source of the plain text as did some other cryptanalysis methods. The key to cracking it is in **finding the key length**. This works because our key is repeated and given our cipher text is long enough we are bound to find the same encryption of certain words or strings. As with any cipher-text only attack, the more cipher text we have, the greater our chances of cracking it. Working through an example will better help us understand this. In the following cipher-text only attack, we will work through the steps to attain a probable key length and perform a frequency analysis to extract the key :

Cipher Text :

EVVUFGJDYPOJXSCZBGSSGDUFCSFDKFREQ,
KVRCSZVSYGHHCSZQUFVZBYSCBTFFFR,
EVVUFGJGBNWVWZUYSEPBFQFGEHEFRVV,
CWKVROSVSTCROAOALVJAZBVEGIRBP,
ZZBGSERUKRBGSSCHTQSIGYOKXSCDCEP.

STEP 1: Find repetitions
Note that Lines 1 & 3 have the repeated string "**EVVUFGJ**", a repetition of the string "**GSS**" in Lines 1 & 5 and a repetition of "**KVR**" in Lines 2 & 4.

STEP 2: Calculate distance between the repetitions.
"**EVVUFGJ**" - 63
"**KVR**" - 63
"**GSS**" - 108

STEP 3: Find the common factors of the distances.
"**EVVUFGJ**" - 7×9
"**KVR**" - 7×9
"**GSS**" - 12×9
Clearly '9' is the common factor among all of them.

STEP 4: Split the cipher text in intervals of the common factor.
Now our cipher text becomes "EVVUFGJDY POJXSCZBG ...

STEP 5: Concatenate every k^{th} letter from the intervals.
Now our cipher text is split into
1) EPSDCCYENPFOOEGSYP
2) VOSKSSSVWBRSAGSSO
3) ...

STEP 6: Perform a frequency analysis on each individual block.
Each block has been encrypted by the same letter. Thus each block is a unique Caesar cipher. Perform a frequency analysis the individual blocks to find out the letter encrypting them.

NOTE
Why do we find the common factors of the repetition distances as described in steps 2 & 3?
Suppose our key length was 'L' and the distance between repetitions was 'D', then D is most likely some integral multiple of L. This is true because if the cipher text has common repeated phrases like 'the' or 'weather' or 'troops', etc. then there is a chance that the same key letters were used to encrypt both the instances. Note that it may also be a coincidence, however, longer the length of the repeated phrases lesser the likelihood it is due to coincidence. So $D = kL$ where $k \in \mathbb{Z}$. The more number of repeated phrases that we find, the greater our chance of finding the key length.

HINT: Key is the name of a famous poet.

An interesting story is that Charles Babbage came up with a technique to crack a variant of the Vigenère cipher in 1854 after a colleague cajoled him into doing so.[4] Although he did not crack the original Vigenère cipher, his method was quite similar to the one Kasiski wrote about 9 years later. However, Babbage never published this method.

Freidman Test
This method of cracking the cipher was published by Milton Friedman in the 1920s. It employs a technique from statistical analysis to measure the index of coincidence of letters in a cipher text in order to approximate the key length. To understand this method, a starting point would be Index Of Coincidence

Further Reading

Beaufort Cipher in WWII :
"Secrecy for Sale". pp 205-210. The Codebreakers: The Story of Secret Writing by David Kahn.
Hagelin M-209
Cryptanalysis Tools :
Kasiski Analysis: Breaking the Code

[4]Singh, Simon (1999). The Code Book. Anchor Books, Random House. ISBN 0-385-49532-3.