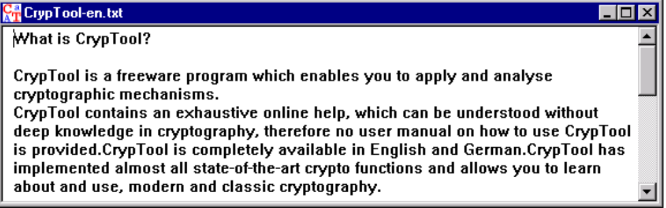
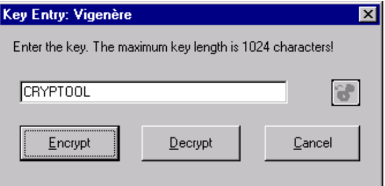
|  |
| --- |
| **Lab Session 03** |

**OBJECT: Implementation of Vigenere Cipher and Atbash.**

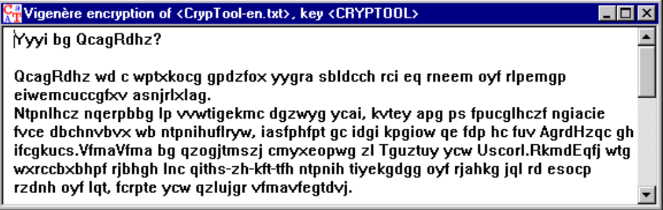
First of all, to acquaint ourselves with the Vigenère encryption algorithm we will open a document, encrypt it and then decrypt it again. We will then try to get the computer to work out the key with which a plaintext is encrypted. To this end we open a document which contains a part of the CrypTool Help text ”Introduction to CrypTool”. The file name of the document to be opened is CrypTool-en.txt. This file is opened via the menu selection File \ Open**.**



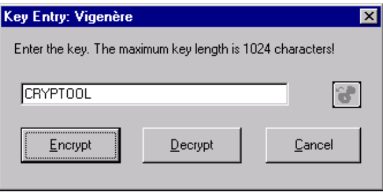
We now want to encrypt this document using the Vigenère encryption algorithm. To do this, we select Encrypt/Decrypt \ Classical \ Vigenère, following which this dialog box appears. As key, we enter CRYPTOOL.



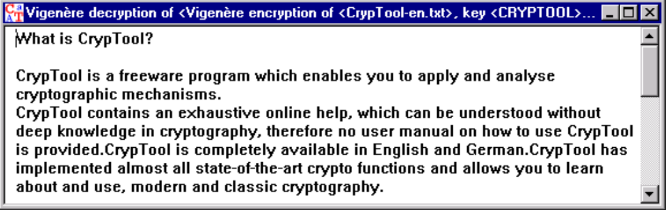
Clicking on the Encrypt button opens a new window that contains the encrypted text.



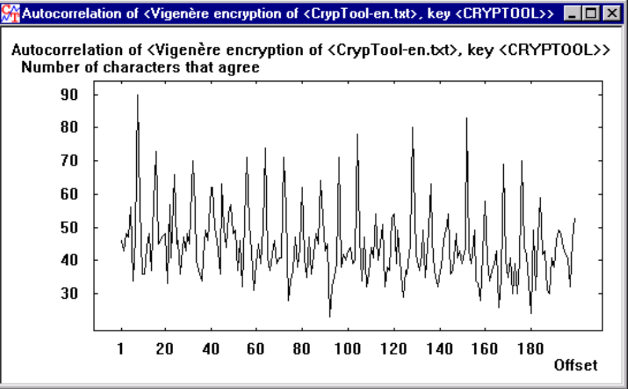
The plaintext version of this encrypted document can now be obtained simply by selecting Encrypt/Decrypt \ Classical \ Vigenère. In the dialog box which then appears we enter the key with which the document was encrypted (CRYPTOOL). This time we do not want the text to be encrypted, but instead to be decrypted. Therefore the Decrypt field must be selected. This is done by clicking on the field itself or on the radial button to the left of the field.



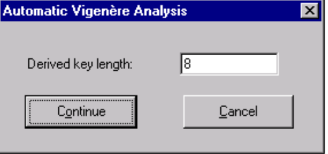
Clicking on the Decrypt button instructs CrypTool to decrypt the text. The plaintext appears immediately.



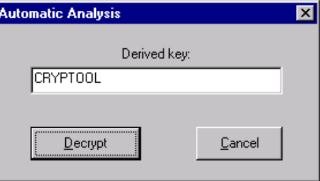
We now wish to find the key with which the document was encrypted. Restore the window containing the encrypted document (via a mouse click) and select Analysis \ ciphertext only\ Vigenère, following which the text will automatically be analyzed. First of all a new, autocorrelation window opens.



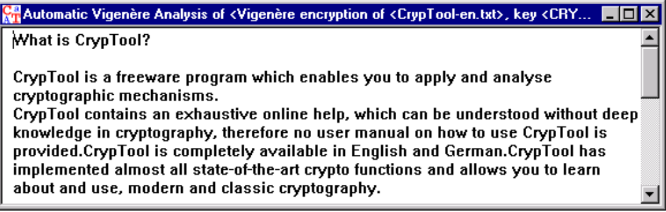
It is possible to work out the key length from the regular peaks in the autocorrelation. It is correctly calculated at 8 characters.



When the Continue button is clicked, another dialog box opens to display the key that has been computed.



Clicking on Decrypt closes this dialog box and opens another window which contains the unencrypted text**.**

****

As you can see, the distinction between upper and lower case and the formatting (blank characters, line breaks etc.) have been retained. Special characters such as umlauts, hyphens and slashes were kept on in the encrypted document. Normally in a classical encryption algorithm all formatting is removed and only the letters are encrypted, but to make the text easier for you to read and compare, CrypTool retains the formatting in the encrypted text (and hence also in the decrypted text). By disabling the formatting options accessed via Options \ Text Options, it is possible to disable retention of formatting and, by checking the relevant boxes again, to reinstate formatting.

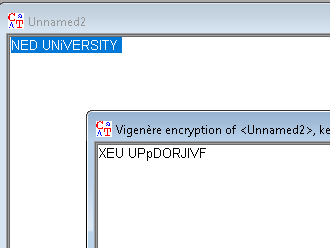
**EXERCISE:**

1. **Explain the working of the Vigenère cipher and the term polyalphabetic substitution. Encrypt the text "NED University" with a key "Karachi". what is the result.**

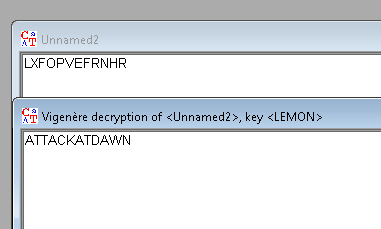
The Vigenère cipher is a **polyalphabetic substitution cipher**, which means it uses multiple substitution alphabets for encryption, rather than a single one like the Caesar cipher.

* **Encryption Process**:
  1. Choose a **keyword** (e.g., "KEY").
  2. Repeat the keyword to match the length of the plaintext (e.g., "HELLO" becomes "KEYKE").
  3. For each letter in the plaintext, shift it by the position of the corresponding letter in the keyword:
     + **H (plain)** and **K (key)**: Shift H by 10 (K is the 11th letter), resulting in **R**.
     + **E (plain)** and **E (key)**: Shift E by 4 (E is the 5th letter), resulting in **I**.
     + Continue this for the entire plaintext.

This generates a ciphertext that changes its letter-shifting pattern based on the keyword.



1. **Decrypt the text "LXFOPVEFRNHR" using the key "LEMONLEMONLE" and explain the process.**



**Steps to Decrypt:**

**1. Align the Ciphertext and Key:**

The key is repeated to match the length of the ciphertext.

Ciphertext: L X F O P V E F R N H R

Key: L E M O N L E M O N L E

**2. Decryption Process:**

**For each letter in the ciphertext, subtract the corresponding letter from the key (in terms of their position in the alphabet) to get the plaintext letter. Here's how it works:**

* Convert both the ciphertext letter and the key letter to their numerical position (A = 0, B = 1, ..., Z = 25).
* Subtract the key letter's position from the ciphertext letter's position.
* If the result is negative, wrap around by adding 26 (since there are 26 letters in the alphabet).
* Convert the result back to a letter.

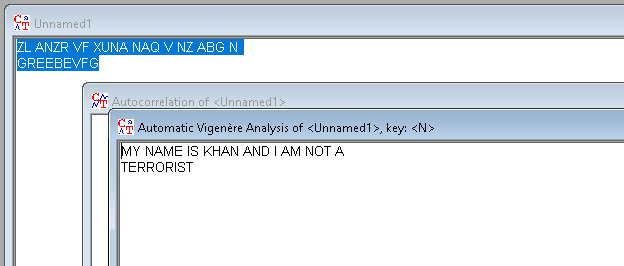
**Decryption for each letter:**

1. **L** (ciphertext) - **L** (key):
   * L = 11, L = 11 → (11 - 11) = 0 → **A**
2. **X** (ciphertext) - **E** (key):
   * X = 23, E = 4 → (23 - 4) = 19 → **T**
3. **F** (ciphertext) - **M** (key):
   * F = 5, M = 12 → (5 - 12) = -7 → (26 - 7) = 19 → **T**
4. **O** (ciphertext) - **O** (key):
   * O = 14, O = 14 → (14 - 14) = 0 → **A**
5. **P** (ciphertext) - **N** (key):
   * P = 15, N = 13 → (15 - 13) = 2 → **C**
6. **V** (ciphertext) - **L** (key):
   * V = 21, L = 11 → (21 - 11) = 10 → **K**
7. **E** (ciphertext) - **E** (key):
   * E = 4, E = 4 → (4 - 4) = 0 → **A**
8. **F** (ciphertext) - **M** (key):
   * F = 5, M = 12 → (5 - 12) = -7 → (26 - 7) = 19 → **T**
9. **R** (ciphertext) - **O** (key):
   * R = 17, O = 14 → (17 - 14) = 3 → **D**
10. **N** (ciphertext) - **N** (key):
    * N = 13, N = 13 → (13 - 13) = 0 → **A**
11. **H** (ciphertext) - **L** (key):
    * H = 7, L = 11 → (7 - 11) = -4 → (26 - 4) = 22 → **W**
12. **R** (ciphertext) - **E** (key):
    * R = 17, E = 4 → (17 - 4) = 13 → **N**

**Result:**

The decrypted plaintext is: **"ATTACKATDAWN"**.

1. **Using brute force decrypt the text "ZL ANZR VF XUNA NAQ V NZ ABG NGREEBEVFG" using Vigenère cipher where the key is a single letter.**



1. **How would you define the Vigenère cipher with respect to Caesar cipher where the key length is 1.**

The **Vigenère cipher** can be seen as a generalization of the **Caesar cipher**, particularly when the key length is greater than 1. However, when the key length of the Vigenère cipher is **1**, it becomes functionally identical to the Caesar cipher. Here's why:

**Vigenère Cipher with Key Length 1:**

* **Key length of 1** means that the same letter in the key is used repeatedly for the entire plaintext.
* The encryption process involves shifting each letter of the plaintext by a constant number of positions in the alphabet, which is determined by the single letter in the key.

**Caesar Cipher:**

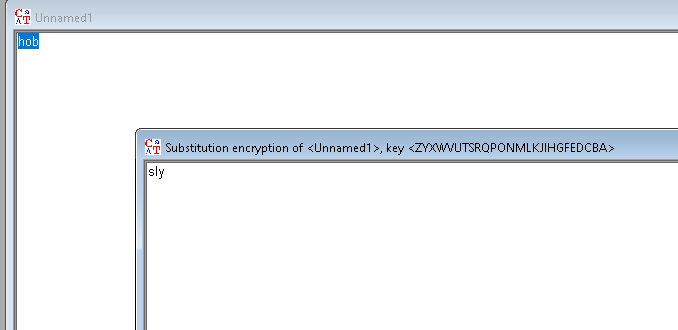
* The Caesar cipher also shifts every letter in the plaintext by a fixed number of positions, typically 3 (but it can be any fixed value from 1 to 25).

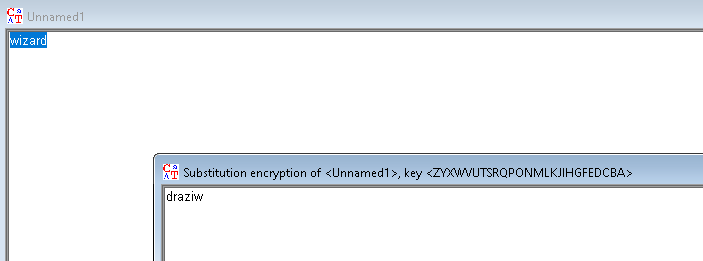
1. **What is atbash substitution, how does it work? Is the atbash a weak or strong cipher and why?**

The **Atbash cipher** is a simple, ancient **monoalphabetic substitution cipher** where each letter of the alphabet is mapped to its reverse counterpart. The first letter (A) is substituted with the last letter (Z), the second letter (B) is substituted with the second-to-last letter (Y), and so on. The **Atbash cipher is weak** because it offers no cryptographic complexity, has no key, and is easily broken through simple analysis of the letter frequencies or recognition of the cipher itself. While it’s interesting from a historical perspective, it provides very little security in modern cryptography.

1. **A few English words 'Atbash' into other English words. For example,"hob"="sly","hold"="slow","holy"="slob","horn"="slim","zoo"="all","irk"="rip","low"="old","glow"="told", and "grog"="tilt." Some other English words Atbash into their own reverses, e.g.,**

**"wizard" = "draziw. Prove this using cryptool and state the advantage or disadvantage of this?**





While the Atbash cipher's ability to turn some words into other English words or their reverses is a curious and fun feature, it doesn't improve its cryptographic strength. The cipher remains easily breakable and has limited practical use in modern encryption, though it retains value in historical studies and recreational cryptography.

1. **If we go in to the atbash menu and select key entry: remaining characters are filed in ascending order, what is the effect of choosing a key 'LEMON' in one instance and in another instance the key 'ZXC'. Which key is stronger?**

The key **'LEMON'** is **stronger** than **'ZXC'** because:

1. It uses more letters, increasing the complexity of the substitution alphabet.
2. The letters in **'LEMON'** are spread across the alphabet, leading to more disruption in the sequential order of the remaining characters.
3. The overall substitution is less predictable, making it more resistant to basic cryptographic attacks like frequency analysis.

However, it’s important to note that **keyed Atbash ciphers (or any monoalphabetic substitution cipher) are still inherently weak** by modern cryptographic standards. Even with a longer, well-distributed key like 'LEMON', they can be broken relatively easily with techniques like frequency analysis or pattern recognition.