1.1 Lab on encryption using binary/byte addition

Under this encryption algorithm, the key entered is added character by character (byte by byte) to the data to be encrypted. Here addition modulo 256 is used, i.e. so that any carry-overs are ignored. The key is applied cyclically (as under the Vigenère encryption algorithm and also with the Exclusive-OR), i.e. once all the characters (bytes) of the key have been used, the algorithm reverts to the first character until the text has been completely encrypted.

To decrypt the text, the characters of the key have to be subtracted from the encrypted text modulo 256.

If one knows the characters which occur most frequently in the plaintext, it is then possible to work out the key with the aid of a computer (and hence also the plaintext) (see Automatic analysis, Byte Addition).

The key used for Binary Addition is entered in the Key entry dialog.

This encryption algorithm can be easily broken with a Ciphertext-Only attack (see Automatic analysis, Byte Addition). An example of this will be found in the Examples chapter.

1. Open the file CrypTool-en.txt under C:\Program Files (x86)\CrypTool\examples.

2. Click “Analysis\Tools for Analysis\Histogram”.

Page | 1
We can see from the histogram that the character which occurs most frequently is the letter E. This is true of many German and English texts. This information will be used later on during our attack.


4. Enter 12 34 AB CD as the key and click Encrypt.

The encrypted message shows up:
5. cipher text only attack will be performed. Choose from menu “Analysis\Symmetric\Ciphertext-only\Byte Addition”.

We are told that key length is calculated to be 4. The commonest character is E with hexadecimal value of 45. If we look at the plaintext, the most frequently character is e with hexadecimal value of 65. We enter into the Expected most common character field in the Byte-by-byte Addition Analysis box 20 (=65-45).

6. Click “Continue”, CrypTool has been able to find the key. The only information was needed to do this was the fact that the character which occurred most frequently in the plaintext was the lower case letter e.

7. Click the “Decrypt” button shows the plaintext.
8. If the text is compressed prior to encryption then we will not be able to draw any conclusions from the frequency distribution of the characters in the text about the frequency distribution of the compressed text, since the compression process not only reduces size of a file but alters the frequencies of the individual characters so that they no longer reflect the frequencies of the characters in the original text. To compress the document, we make startingexample-en.txt active again. And select “Indiv. Procedure\Tools\Compress\Zip”, the rate of compression is displayed.

9. Click “OK”, the compressed document is shown.

10. Click “Analysis\Tools for Analysis\Histogram” to see its histogram. The compression produces a quite different histogram profile from the one previously obtained for the uncompressed document. The characters are much more evenly distributed than in the unencrypted document.
11. Make the compressed document the active window once again and the encrypt it using the same key 12 34 AB CD.

12. Click “Encrypt”.

13. We invoke the analysis again by choosing from menu “Analysis\Symmetric\Ciphertext-only\Byte Addition”.

CrypTool returns an incorrect key length of 12.

Given this key length, it is not possible to find the correct key either.

14. We will check whether it is possible to arrive at a readable version of the text document from the compressed and then encrypted document. We will provide the key and then unzip.

We will make the compressed and encrypted document the active window again. Choose from menu “Encrypt/Decrypt\Symmetric\Byte Addition”.

15. Enter **12 34 AB CD** as the key and click **Decrypt**.
16. Choose from menu “**Indiv. Procedure\Tools\Compress\UnZip**”, and the original text is displayed.