**CSE539 Applied Cryptography – Project 1b**

Name : Amaresh Meher Basanth Talluri

ASU ID : 1207301416

**One Time Pad**:

This is an type of stream cipher which takes advantage of the concept of XORing. Here we use a key of the same length of the message, and XOR it with the message to get the ciphered text. we can get back the message by only XORing the message back with the key.

In this way the sender and receiver of the data can decide a common key, and transfer message by encrypting them with this key, and only the receiver can decrypt them since only he has they key (other then the sender). This is a symmetric encryption technique.

**Steps to execute the project**:

1. Create a QT project and add all the '.c' files from the project into it.
2. Save the files 'words', 'cipher000' and 'cipher001' in the qt's project folder.
3. Build the project and run it.
4. You should get the output folders 'pad', 'msg1' and 'msg2' created in the qt's project folder.

**Algorithm** :

The first thing the code does is to read both the cipher texts (as Strings) from the files and XOR the two strings. This gives us the resultant cipher.

Initially the complete message string is filled with a special character, in this case '&'. The purpose of this is to know exactly when to end the algorithm. When all the characters (I.e. '&') are replaced then we know that the message has been generated. Here decryption is done in two phases. First all the text is decrypted and then all the special characters are decrypted.

The first step is to pick up a word from the word list. In this code I am using brute force, where I check with all the words from the list to check if its present in message. The following steps explain what checks are performed before a word is assumed to be in the message.

* First I XOR the two cipher texts of the messaged to get the pad.
* Then I loop through each position in the pad to XOR with the guessed word (from the word list)
  + But before XORing I check if that part of the message contains a '&'. Meaning it hasn't been decrypted yet. If it has been decrypted, I loop to the next position.
  + If that part is not been decrypted yet, I XOR it with the guessed word. And I check if the result is a readable text. If it is not a readable text I continue the loop to the next position in the pad. But if it is a readable text, I perform a few checks on it.
    - I first check if this readable text is of the format 'abc\*' or 'abc\*de\*' (where \* is the special charactor ' ', '!', ',', '\n'). If it is not I perform the next check. But if it is
      * I consider the word before the \* I.e. abc has a word similar to it in the list. If there is a similar word, then I add the guessed word to the message, or else I just continue the looping operation.
    - The next check is whether the readable text has a special charecter in between ( the first check here is for ' ', and this is become it is the most common and could occur after other special charecters e.g a, b). If there is I consider the word after the special charecter (eg. 'abc de', I consider 'de').
      * with this 'de' I check if it's the last word in the message and also if there is a word 'de' in the word list, if there is I add the guessed word to the message and return from the function.
      * But if there is no exact word 'de', I check if there are any similar word, I loop through each word and Xor it back with the pad and try to get back the guessed word.
        + I check if the word obtained in readable (else continue the loop)
        + if it is I check if the initial word obtained is a subset of this word ( example initial is 'the' now we got 'the p') if it is not I continue the loop, but if it is

if the resultant word is of the type 'the p' I check for all the words starting from p, and start of the operation again.

if the resultant word is of the type '...them...' then I check if it's the last word and add it or I check if the next character is a space(of type '...them ...' and add it.

* + - But if there are no similar words (i.e. of the type '...abc...'), I check if it's the last word and add the guessed word to message or I check if the next character is a space and add it to the message.

By performing all these check we cover every possible outcome of the resultant text when XORed with the guessed word. After this we get the message with all the words but no special characters.

For the special characters I again XOR through the message with all the four possible special characters and see if I get a readable text. if it is readable, I updated the message with the special character.

**References** :

1. http://travisdazell.blogspot.com/2012/11/many-time-pad-attack-crib-drag.html
2. http://stackoverflow.com/questions/12671510/xor-on-two-hexadeicmal-values-stored-as-string-in-c
3. http://stackoverflow.com/questions/12671510/xor-on-two-hexadeicmal-values-stored-as-string-in-c
4. http://stackoverflow.com/questions/3381614/c-convert-string-to-hexadecimal-and-vice-versa
5. http://stackoverflow.com/questions/3381614/c-convert-string-to-hexadecimal-and-vice-versa