**Project 1b: Crib-Drag Algorithm**

Submitted by: Anshul Jain ([ajain89@asu.edu](mailto:ajain89@asu.edu))

Crib drag is the method that is used to decode the messages even without knowing the key with which they were encrypted. This can only happen if the OTP (One Time Pad) is used for encrypting multiple messages. This is the limitation of the One-Time Pad.

This project helped to understand how easy it is to uncover the encrypted messages if they are encrypted using the same Pad. The algorithm works on a very simple logic: ‘A’ XOR ‘B’ XOR ‘A’ = ‘B’.

I used the same approach in this implementation.

This project contains two parts:

1. Encrypting the message using the One-Time Pad.
   1. Let A = Message 1 in Binary format
   2. Let B = One-Time Pad in Binary format.

So Cipher text 1 = A XOR B which is also in Binary format (and same for Message 2)

1. Decrypting the messages using the Crib-Drag Algorithm

**Explanation of Crib-Drag Algorithm I implemented in this project**

**ASSUMPTIONS:**

1. The message should not contain any special character.
2. The words in the messages should match exactly with the words in the “words” file present in linux.
3. The messages should be of equal lengths.

**IMPLEMENTATION:**

The program starts with taking the cipher text’s file name as the input (cipher000, cipher001, …….) and generates the file containing the readable messages.

Let C1 = Cipher Text 1 and C2 = Cipher Text 2

C = C1 XOR C2

I have used a vector of cribs that contains all the words present in the ‘words’ file. The time to obtain the original messages depends on how the cribs are selected. For testing the algorithm, a small crib file can be used that contains the words according to the following conditions:

1. The starting word of any of the 2 messages should be present in the crib.
2. If the messages contains space at the same location, then add the next word in of any of the 2 messages in the cribs. For ex.: in “in at world”, “of programs”, either at or programs should be there in the crib.
3. If a word in message 1 ends one character before the word in message 2, then add the next word in the crib. For ex.: in “in at world the”, “of program that” the word “program” ends one character before “world”, so add the or that in the crib.

This crib file can help to get the messages within seconds. The algorithm will work with the complete words file as well but it may take some time. The program will never enter the infinite loop, it may take some time but it will return some value, maybe nothing in some cases, but it will terminate.

The implementation works on two recursive functions:

1. int decryptUsingCribMain(string, string);
2. int decryptUsingCrib(string, string, string);

**int decryptUsingCribMain(string C1xorC2, string prevStr):**

This function is called at 3 different occasions:

1. By the main() program
2. After we find that the 2 messages contains the space character ‘ ‘ at the same index.
3. When a word in message 1 ends one character before a word in message 2.

This function will then send the control to the 2nd function according to the requirement.

**int decryptUsingCrib(string C1xorC2, string HexCrib, string prevStr):**

This is the main implementation of the Crib-Drag algorithm. It adds a pad in the vector whenever it finds a new One-Time Pad. It calls itself recursively if it finds “C1xorC2 XOR HexCrib” readable. It sends “C1xorC2 XOR HexCrib” as the new HexCrib if it is readable.

**DataStructures:**

1. Map<char, vector<string> >: Maps are associative containers that store elements formed by a combination of a key value and a mapped value. In my implementation, I have used map to create a dictionary of words with first character as there key.
2. Vector<string>: Vectors are sequence containers representing arrays that can change in size. I have used vectors for storing multiple values like the complete dictionary, the cribs, the One Time Pads, Cipher Texts, etc.

During implementation of the Crib-Drag algorithm, I realized that there can be many results possible. For a simple message containing four words, I got 478 unique results and it took some time to display all those results. So in my final submission I have restricted the number of results to 1 which can be changed easily by updating the value of the global variable “MaxFound”. A part of the 478 unique messages is attached with this report.

For the following example, the algorithm works as follows:

Message 1: in the an world

Message 2: of program hello

C1xorC2 = XOR of the above 2 messages in HEX format.

decryptUsingCribMain – this will send C1xorC2 and “” to decryptUsingCrib function.

The algorithm will try to find the “in” in the cribs and will get “of” corresponding to “in”.

Now the control will be sent back to decryptUsingCribMain function as the message contain space at the same location and it will send the substring of C1xorC2 with “in” as the prevStr.

Now corresponding to “the”, it will find “program” and then finally the complete string.

This program can take multiple cipher texts as the input.

The message may not be accurate because there are many results possible for a particular message. With some assumptions, it is easy to decrypt the messages. This project has helped me to understand the limitation of using one-time pad to encrypt multiple messages.