Programming for Cryptologists 101

Julian Bhardwaj

September 17, 2011

Abstract

This document is intended as a brief introduction to the use of computers as electronic aid in Cryptanalysis. It was written as a companion to the National Cipher Challenge, run by the University of Southampton. This document assumes no prior programming experience but it designed for someone with a basic understanding of Classical Cryptanalysis. It is not intended for this document to provide a exhaustive guide, rather 'starting points' for a beginner to be able to learn independently.

1 Introduction

Cryptography and Cryptanalysis have always been and still are practices of Mathematicians. Charles Babbage, renown for his work breaking Vigenère's Auto Key cipher and his work on the Difference and Analytical Engines, was a Mathematician. Yet with the boom of technology with the Computer Age, Computer Science has progressed Cryptography and Cryptanalysis simultaneously. Therefore, even when tackling Classical Ciphers, a basic understanding of Computer Use and Programming can greatly aid a Cryptanalyst. This said, being able to program isn't essential! However, the follow guide seeks to try and provide starting points for those wishing to develop Computing skills relating to Cryptography. This is not intended an an exhaustive tutorial and is deliberately vague in parts! I firmly belief that Programming is best self taught through experimentation and vast use of search engines such as Google for following up topics/error messages you might encounter. The majority of citations in this document are for 'Wikipedia' such to allow quick and easy access to a multitude of online sources relating to the topic.

2 Entry Level:

You don't need to be an Elite Programmer to take advantage of a few simple time-saving features that Computers have to offer.

Substitution

Use of a Word Processing Package (for example Microsoft Word, or Notepad++ [1]) can aid decrypting Monoalphabetic Substitution Ciphers [2]. You can use this function (Ctrl-R or Ctrl-F in some packages) to replace all instances of one letter with another letter (for example, replacing all A's with d's).

By convention, plaintext (decrypted) is denoted with lowercase letters and ciphertext (encrypted) is denoted with uppercase letters. It is important to stick to this when Find-and-Replacing and to ensure that the 'Match Case'/'Case Sensitive' option is ticked. Otherwise, as you are replacing ciphertext letters with plaintext letters, you would inadvertently start overwriting plaintext letters with different plaintext letters.

NB: for competitors of the National Cipher Challenge, if you are using a package like Microsoft Word which has 'Auto-Correct' functionality, it is safest to turn this off first. Harry likes to throw the occasional spelling mistake into the Challenges as examples of cipher-clerk error, and you must submit your decryption with mistakes intact! Punctuation isn't important however, neither is upper/lowercase (See Rule 9 [3])

[1] http://notepad-plus-plus.org/

[2] http://en.wikipedia.org/wiki/Substitution_cipher#Simple_substitution

[3] http://www.cipher.maths.soton.ac.uk/rules

Transposition

When decrypting Transposition Ciphers [4], it is often more accurate and quickly to use a Word Processing Package, even a simple one like Notepad. To do this effectively however, you must be using a Mono-spaced Font [5] so that each character is exactly the same size. This allows you to line up rows of text in columns. Transposition Cipher decryption becomes even easier when using a Spreadsheet Package (See Foundation Level).

Frequency Analysis

One of the most basic skills of Cryptanalysis, developed originally by the Arabs, is Frequency Analysis[6]. This can be vastly aided using Computers. The most simplest approach is to use the 'Find' facility of a Word Processor to tell you the number of occurrences of a particular letter within a selection of text. 'Word Count' can then also be used to provide the total number of letters within the text. The 'frequency' of each letter can therefore be deduced more accurately from this.

3 Foundation Level:

Cryptanalysis can be further sped up with a few tricks using a Spreadsheet package like Microsoft Excel. Whilst Spreadsheet packages are designed primarily for Financial and Database purposes, the String Manipulation and Lookup features provided can be of great use to a Cryptanalyst.

String Manipulation

It is possible to take advantage of the 'cell' (rows and columns) layout of a Spreadsheet package to split a (reasonable sized) text such that each letter takes up one cell. This is particularly useful for Transposition Ciphers where given say a block length of 5, you can first insert line breaks every 5 letters into your text and copy this into your Spreadsheet package. You can then split each row of 5 letters into separate cells by using a String Indexing function, for example MID in Microsoft Excel. By copy-and-pasting of your formulae, you can quickly iterate through the rows this way. Transposition is then as simple as rearranging the columns.

```
LEFT(text,num_chars)
MID(text,start_num,num_chars)
RIGHT(text,num_chars)
```

Text The Cell Reference or String for the text you are selecting from.

Start_num The index of the first character you wish to retrieve. This only applies to the MID function, LEFT always starts at the first character and selects to the right of this and RIGHT always starts at the last character and selects to the left of this.

Num_chars The number of characters to retrieve.

At times, especially when applying more mathematical transforms like an Affine Shift [7], you might want to convert each letter to a number. Normally in Classical Cryptography, the Latin alphabet letters are assigned numbers from 0-25 (or 1-26). However, computers by default use a different encoding, normally one called ASCII. The mapping of each character/symbol to an ASCII number can be seen here: `http://www.asciitable.com/`. Most programming languages and spreadsheet packages provide a simple function to convert a character to and from its ASCII character code, for example in Microsoft Excel:

```
CODE(text)
```

Text The character you wish to convert into a number.

[4] http://en.wikipedia.org/wiki/Transposition_cipher

[5] http://en.wikipedia.org/wiki/Mono-spaced_font

[6] http://en.wikipedia.org/wiki/Frequency_analysis

[7] http://en.wikipedia.org/wiki/Affine_cipher

```
CHAR(number)
```

Number The number you wish to convert into a character.

The uppercase letters have ASCII codes from 65-90, so by using the formula:

```
=CODE(TEXT) - 65
```

You can get a number between 0-25 for each letter. Alternatively, you can use a 'Lookup' function to convert a letter to any other character (number or letter) you like, detailed below.

Database/Lookup Functions

The concept of using a 'Lookup' function is that you have a table of values being mapping onto alternate values. Microsoft Excel provides a series of Lookup and Database functions which are designed for extracting particular items of data out of bigger tables/databases, however they work quite well for even a simple 2 by 26 table of values. Whether you need the VLOOKUP and similar HLOOKUP function depends on which way round your table is orientated.

```
VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)
HLOOKUP(lookup_value,table_array,row_index_num,range_lookup)
```

Lookup_value They 'key' value to be found in the first column/row of your table.
Table_array The range of cells which form your table.
Col_index_num/Row_index_num The column/row index number that you wish to be returned.
Range_lookup TRUE/FALSE whether you want to look for an exact match (TRUE) or approximate match (FALSE) through use of Wildcards (*/?)

Microsoft Excel also provides other Database functions which can be used as alternatives to the aforementioned Lookup functions.

Transposing Text

Most Spreadsheet packages provides multiple copy-and-paste functions, amongst which the 'Transpose' option will allow you to flip rows and columns. This is particular useful if you are tackling a cipher which includes a Caesar Square[8] element.

4 Intermediate Level:

One way of yielding the power of a Computer is through use of 'scripts' or 'macros'. Microsoft Excel provides functionality for both use of macros and a more extensive programming language known as 'VBA' (Visual Basic for Applications[9]). Alternatively, scripting languages such a Python[10] and PHP[11] can provide a simply environment for learning the basics of Programming and being able to pick up a few tricks to aid your Cryptanalysis. The rest of this section focuses on Python as it is both very powerful and versatile, but simple to get started with (PHP requires a web server to run on). Additionally, Python provides an interactive GUI which lets you instantly run commands without the need to write a full program and compile it.

[8] A 'Square' transposition, attributed to Julius Cesaer as the 'Caesar Square', simply reads off the message in columns instead of rows.

[9] http://en.wikipedia.org/wiki/Visual_Basic_for_Applications

[10] http://www.python.org/

[11] http://www.php.net/

Getting Started with Python

There are myriad guides and tutorials for Python 101 accessible through your search engine of choice[12], so instead of reproducing these this section aims to provide a few basic pointers relevant to Cryptanalysis. Firstly, if you are not running a Linux system which ships with Python or are running Windows/Mac, you'll need to download Python here: http://www.python.org/download/. Python 3 isn't used that widely yet, so in the interest of being able to access a plethora of compatible tutorials, I would recommend Python 2. At time of writing, the majority of readers will want to download the 'Python 2.7.2 Windows Installer' Binary. The official documentation and tutorials for this are available here: [13]. You'll want to start by playing around with the 'IDLE' - the Interactive Python GUI which lets your execute commands 'on-the-fly'.

The first concept you will want to be familiar with is that of a 'variable'[14]. Try running the following commands:

```
plaintext = "this is a plaintext message."
plaintext2 = "this is another plaintext message."
print plaintext
print plaintext + plaintext2
```

String Manipulation

Experiment with the functions demonstrated with the following lines to explore the String Manipulation functions of Python, which are detailed extensively here: http://docs.python.org/library/string.html.

```
pt = "hello, this is a rather boring sample message"
len(pt)
pt[0]
pt[45]
print pt[44]
pt.upper()
words = pt.split(" ")
print words
```

Note: The fourth line above should give you an 'IndexError: string index out of range'.

Iteration

Iteration[15], otherwise known as 'looping', is one of the core paradigms within all programming languages. The concept is that you tell the computer to execute a number of commands either a set number of times, or until a certain condition is met. Particularly common loop structures are the 'For' Loop[16] and the 'While' Loop[17].Python makes it particularly easy to loop through an List[18] of variables, shown in this example:

```
words = [‘the’,‘cat’,‘sat’,‘on’,‘wall’]
for word in words:
        print "word: " + word
```

Note the colon after the first line of the For statement, and the indentation of the commands to be looped.

Another similar example is such:

[12] http://www.google.co.uk

[13] http://docs.python.org/

[14] http://en.wikipedia.org/wiki/Variable_(computer_science)

[15] http://en.wikipedia.org/wiki/Iteration#Computing

[16] http://wiki.python.org/moin/ForLoop

[17] http://wiki.python.org/moin/WhileLoop

[18] http://docs.python.org/tutorial/datastructures.html

```
cipher = "SRTHKBLCOSKEMDFGKGOIW"
plaintext = ""
key = 5
for letter in cipher:
        plaintext += decrypt(letter, key)
print plaintext
```

Before you can run this however, you must first of defined ‘Decrypt’ as a function[19]. By building up a library of Crypto functions, saving them in a file (for example, crypto.py), and them importing them at the beginning of your session with:

```
import crypto.py
```

You will quickly be able to yield the power of Python against the ciphers you are tackling.

5 Advanced:

Most high level languages, including Python, are what is known as ‘Object Orientated’[20]. There are many tutorials available on the Internet which introduce object orientated paradigms, however the basic idea is that you define your own ‘classes’, which are a collection of variables and functions/procedures which can be called. Two very popular high level languages which are very object orientated are Java[21] and the .NET family[22] (e.g. Visual Basic, C#). The following examples are given in Java, which is a very popular first choice of language to learn (most university Computer Science courses start with Java).

Object Orientated Programming

There are innumerable tutorials available for getting starting with Java, as well as for Object Orientated Programming. However, you'll find most of the OO tutorials will go into great detail with the more complex concepts such as Inheritance[23] which you don't need to worry yourself with at this stage. The following shows an example class declaration for a Cipher class. Note, in Java a class should be defined within a file with the same name as the class name; the following class would be declared in Cipher.java

```
public class Cipher {
        //this is a comment in Java.
        String cipher;
        double letterfreqs[];
        static String engalpha = "abcdefghijklmnopqrstuvwxyz";
        static double[] engfreq =
{8.17,1.49,2.78,4.25,12.70,2.23,2.02,6.09,6.97,0.15,0.77,4.03,2.41,
        6.75,7.51,1.93,0.10,5.99,6.33,9.06,2.76,0.98,2.36,0.15,1.97,0.07};
        static String ordalpha = "etaoinshrdlcumwfgypbvkjxqz";
        static double[] ordfreq =
{12.70,9.06,8.17,7.51,6.97,6.75,6.33,6.09,5.99,4.25,4.03,2.78,2.76,
        2.41,2.36,2.23,2.02,1.97,1.93,1.49,0.98,0.77,0.15,0.15,0.10,0.07};

        public Cipher(String ct) { //Class Constructor
                this.ciphertext = ct;
                freqAnalysis();
        }
```

[19] http://docs.python.org/tutorial/controlflow.html#defining-functions
[20] http://en.wikipedia.org/wiki/Object-oriented_programming
[21] http://en.wikipedia.org/wiki/Java_(programming_language)
[22] http://en.wikipedia.org/wiki/.NET_Framework
[23] http://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)

```
        void freqAnalysis() {
                //Code to count the letter frequencies of this.ciphertext
                //into the array this.letterfreqs
                return;
        }

        public double getLetterFreq(char letter) {
                //A function to return the frequency of a particular letter
                return letterfreq[x];
        }
        public void printCipher() {
                System.out.println(ciphertext);
                return;
        }
}
```

You can then create instances of this class from your main code.

6 Next Steps

Once you have identified a starting point within this document, I recommend you follow this procedure for expanding your knowledge and experience of programming:

1. Search on the Internet for a piece of code demonstrating how to do something.
2. Adapt the code so that it does what you want it to.
3. Search on the Internet for an explanation of the inevitable Error Message you are receiving:
4. Correct your code so it runs properly.
5. Repeat

Have fun!