University of Maryland Baltimore County

CMPE/ENEE 491/691
Hardware Security
Spring 2023

Lab 1: Vigenere Tableux Cipher

Due Date: 02/21/2023

Deliverables

1. A .zip file containing your source codes and testbenches (.v and/or .vhd files), a Makefile for building and running your simulations executable, and a short Readme describing how to run each.

In this lab, you deal with the Vigenere Tableux Cipher encryption/decryption scheme discussed in class. This homework has two parts as below and for each part, your code and testbench should be written using either VHDL or Verilog.

a) Implement the Vigenere Tableux Cipher encryption/decryption scheme and a testbench for your design either using Verilog or VHDL. Then, you should simulate your design with your testbench.

Your testbench should read an input file called "in.txt". Figure 1 shows a sample input file. As shown, the first line includes a flag showing that you want to do encryption (if flag=1) or decryption (if flag=0). The flag follows with a string which is the KEY. Note that there is a space between the flag and the KEY string. KEY can be a string with any size.

Note that Figure 1 is an example. Your code should be able to read any text included in "in.txt" and perform encryption/decryption accordingly.

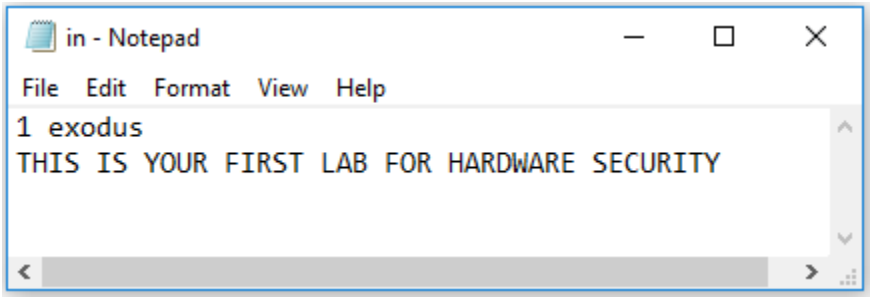

Figure 1. Input Data File

The testbench should write the output string (ciphertext when simulating the encryption design, and plaintext when simulating the decryption design) to an output data file called "out.txt".
Figure 2 shows the resulting ciphertext in the output file for the input file shown in Figure 1.

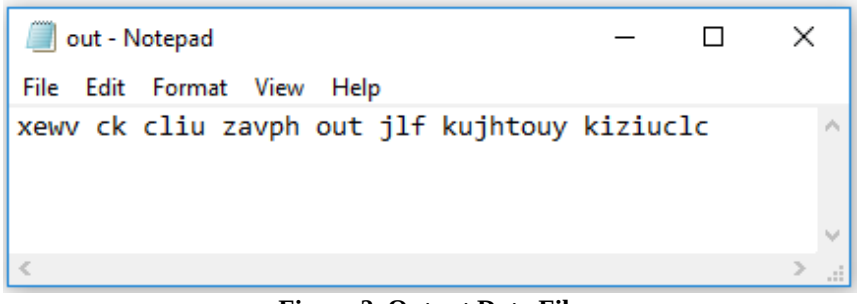

Figure 2. Output Data File

Note that the file called “table.txt” includes the table you need for encryption/decryption. You may or may not use table.txt file (based on your algorithm). It is up to you.

b) In this part of the homework, you are to break the Vigenere Tableux Cipher. For this part, your code finds the key used for encryption. In this part, the input file “plain_cipher.txt” file includes a number of (it can be one or more) plain texts and the corresponding ciphertexts. The odd lines include plaintexts and the even lines following each odd line include the related ciphertext. You know that the ciphertext was generated using Vigenere Tableux Cipher. You are to find the key used for encryption.

   Note that only one key was used for encryption. So you should make sure that your key is correct for each pair of plain/cipher texts. You need to write the key in the “key.txt” file. Figure 3 shows a sample input file and Figure 4 shows the related output file. In case, you cannot find any valid key, the output file (“key.txt”) should include: “NO VALID KEY WAS FOUND”

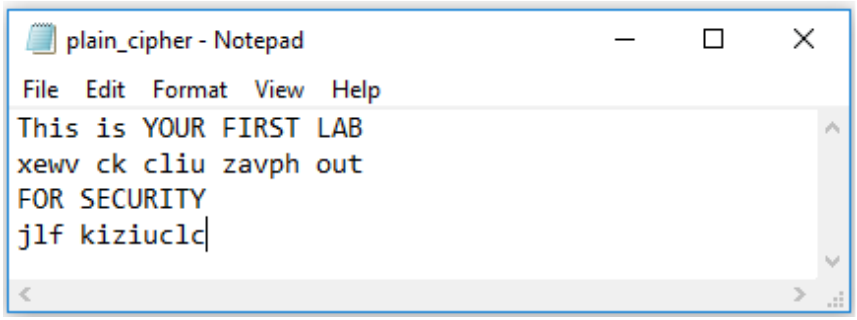

Figure 3. Input plain_cipher.txt file

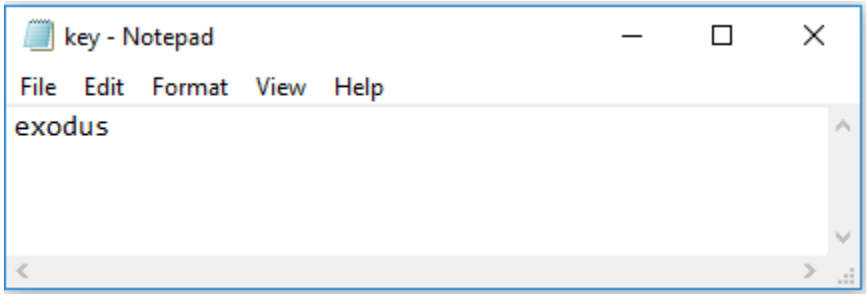

Figure 4. Output key.txt file

Note1: Your output in both parts a&b can be lowercase or uppercase. You can print as you wish.

Note2: What follows gives some examples to clarify part b.

- If your extracted key is “exodusexodus” then you should report the key as “exodus”
- If your extracted key is “exodusexodusg” then you should report the key as “exodusexodusg” (because the pattern of exodus is not repeated as the last charcter is “g”)
- If your extracted key is “exoduse”, then you should report the key as “exodus” (because we can consider that second “e” relates to the next iteration of key)

- If your extracted key is “exodusexodusexodusexodusab” then you should report the key as “exodusexodusexodusexodusab” (it is true that exodus is repeated 4 times but the last two characters are “ab”)

If you want to verify your results in part a, you can use the following website for encryption and decryption using Vigenere Tableux algorithm (please do not change the default setting of the website):

https://cryptii.com/pipes/vigenere-cipher