Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NetID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Activity Cryptanalysis**

**Total Points: 50 Bonus Points**

**Part 0: Getting Ready**

By now you should have working Vignere and Frequency Analysis tools.

In this assignment are multiple files inside sample.tar.gz: [Read README for a description]

These are sample encryptions meant to compare Vigenère algorithms.

First output group:

Encrypted with KEY: wereandroid (there is a <space> in the middle)

Using Alphabet “**abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ**” (53)

This alphabet has a <space> in the middle between z and A.

plain0.txt 🡪 cypher0.txt 🡪 recover0.txt [If you decided to pass not in alphabet characters]

plain0.txt 🡪 cypher0.txt 🡪 recover0\_ignore.txt [If you decided to drop not in alphabet char in input]

Second output group:

Encrypted with KEY cypher

Using Alphabet “**abcdefghijklmnopqrstuvwxyz**” (26)

plain1.txt 🡪 cypher1.txt 🡪 recover1.txt

If you can see the message clearly then you are in line with Vigenère for the activity.

dictionary.tar.gz contains more sample files to use as a frequency analysis tools.

**If you are unable to bring a laptop to class and are unable to virtually connect to your application hardware, join the TEAMs meeting set up for this if you wish to participate.**

**Part 1: Sending, Receiving and Analyzing Ciphers**

**This activity is arranged for three people each taking turns to participate in one of three roles:**

1. **Sender: Encrypt and send a message to the Receiver using the Vigenère algorithm with any key. Share the alphabet and key you used with Receiver. Give the receiver cyphertext result. Give the cryptanalyst ONLY the cyphertext file. If you want to have a conversation over cyphertext by sending single sentences back and forth that is also acceptable so long as the cryptanalyst gets your cypher transmissions.**
2. **Receiver: Decrypt the message using the same algorithm with parameters matching the Sender’s. You can transmit another message in return and act as the role of Sender. Sending more messages using the same key will aid the Cryptanalyst.**
3. **Cryptanalyst: Using frequency analysis, chosen text and brute force attempt to determine the sender’s message.**

**You will need:**

* **Your laptop with HW applications. A compiler or development environment for the laptop to recompile if you need to update.**
* **Your version of Vignere (Caesar is just Vignere with key = <one character>**
* **Your version of Frequency Analysis.**
* **A set of plaintext files of varying sizes to encrypt and send and to decrypt.**
* **Ability to share files with one another, this can be done through OneDrive, Email or some other media (e.g. flash drives).**
* **Do not copy paste text to share with one another, this will likely mar your text files with control characters, especially if you move from Linux-> Windows or to MAC. It will make it difficult to encode and decrypt. Transmit whole text files for sharing.**

**Working in groups of three take your algorithm and create a plaintext file with a message of your choice. Encrypt this file and give it, the algorithm used and the parameters such as alphabet and keys to the Receiver. Next, give the transmitted cyphertext to the Cryptanalyst.**

**The Cryptanalyst can then ask Receiver/Sender to do any or all of the following.**

1. **Encrypt one or more of the messages for them to collect data (you can use text files provided below or make your own) using the same algorithm and provide it to the cryptanalyst. Do not tell them what you used.**
2. **The cryptanalyst will provide sender with a phrase. Generate a new encrypted message and include the “chosen text” they asked you to include and provide the cypher text to the Cryptanalyst.**

**Trade off roles and assume a different role and repeat until all have completed the role of cryptanalyst.**

**Repeat for a Caesar cypher (use only one character as key to produce Caesar output) to the Cryptanalyst role and note the difference in encryption strengths and frequency results.**

**As cryptanalyst, you can:**

1. **Use your generated frequency analysis to see if you can correlate any of the “message” cypher text transmitted to a readable plain text.**
2. **Request a “chosen text” phrase of two words or more. They will send a message with at least two of those words embedded in that message sequentially. See if you can match a cyphertext pattern to that chosen text, do any of the letters fall in the frequency pattern that results? Ask for more chosen text messages if it will help you.**
3. **Try using long words in your phrase, what about repeated words and characters?**
4. **Ask for a large file encryption. Does a frequency analysis of a dictionary compare to intercepted Vigenère cyphertext message?**
5. **Is it helpful to set your alphabet beyond the range [a-z][A-Z] and allow it to include the wider values or determine the character set from the cyphertext? What does the cyphertext reveal about the alphabet?**

**The Goal is not to break the message but to go about the process and thinking you might use to try to break into messages given what you know about the sender, cyphertext, analysis and algorithm.**

**Provide some screenshots of encryption as Sender, decryption as Receiver and frequency analysis as Cryptanalyst and discuss the following:**

**Q-1) Did chosen text help you find a pattern? If so, what “chosen text” worked well?**

**Q-2) Did frequency analysis help? If so, how did the frequency analysis help?**

**Q-3) Does this encryption method show any cryptanalytical weaknesses. What are they if so?**

**Q-4) How does Caesar differ from Vignere cryptanalytically?**

**Submission Guidelines:**

* No handwritten submission is accepted, always submit answers as text within this or similar document file with any support images embedded in the file.
* **EXCEPTION**: If asked for source code implementation you can submit those individually and as separate files in ASCII format in their original file format .cpp, .java, .py, .cs etc. or even as a .txt file will be acceptable. Do not insert code into the submission document file. It ruins spacing which makes python and some languages (perl, awk etc.) difficult to test build. Implementation languages preferred are C#, C/C++, Java and Python. If you plan to use another language, please let me know.
* Do not submit ZIP files… ever… for anything in D2L. The system is extremely unhelpful with regards to those filetypes and grading.
* You may include your freehand drawing/image and handwritten scans in the submission. However, the writing and images must be clearly legible. Though, it is best to present non-handwritten submissions, generally, as is done in the professional setting.
* If asked, show all work/calculations/graphs etc. in the determination of the problem.
* **Please complete your entire work in a single Word Document and Save the file as: yournetid\_CS3502\_Assignment01.docx (e.g. ogarcia5\_CS3502\_Assignment01.docx.) and upload your file in D2L.**
* Please observe the submission due date and time. After the due date there is a 50% penalty for the next 24 hours. Any submission after 24 hours of the due date will be graded at 0%.
* If you include a reference or an image taken from other sources, please cite them appropriately. APA is preferred but cite them so they can be found. **NOTE: verbatim copying or even paraphrasing is plagiarism so if the source used constitutes your answer rather than simply *supporting* the answer, it will be considered invalid. This is especially true of source code implementation answers.**
* If you resubmit, please make sure to attach the file again. Your latest submission before the due date will be the one graded.