Joey Maffiola

Cryptography and Network security

Project 1 (Cryptanalysis)

10/4/23

I performed frequency analysis using my program, which gave me my first few guesses. First, I looked for the letter that might be “I”. I did this by looking for what letter was frequently alone, which turned out to be the letter P. The next thing I did was try to search for three letter combinations that repeated a lot. I was looking for the word “the” which I was able to find out quickly. S was T, F was H, and J was E.

After this, I was stumped for a while, until I saw a word at the bottom of the page that looked alot like the word “This”. Then, I swapped out W for S, and all of a sudden the word next to it looked like it might be “website”. This is when the ball started rolling faster than before. When I put it together, it said, “This website I.C...ES.” I was able to figure out that the word next to website was “includes” which got me the pairings N→U, U→C, R→L, V→D, E→A.

Things went easily after that. I looked at chapter headings, the roman numerals, the title, and much more. “The Blue Castle” helped me figure out it was a novel written “by” someone. I wrote down on a piece of paper the name and the original letters and went down the row to figure out it was Frederick. I used diffchecker.com to check differences from time to time. During decryption I would use periods to represent parts of the key I didn't know yet, and fill the parts I did know or was testing in their respective places. What resulted was a plaintext document where only the known and currently tested letters were let through, and everything else was a period. This made it easier for me to decipher what was rubbish and what was plaintext.

While decrypting on the computer it was also helpful to use a whiteboard and color coordinate which letters I knew were correct and which letters were guesses. I was able to only use three colors; black, red, and orange. I used black for the letters I knew were correct. Red was for letters I added in secondary and didn't have to look over the same letters again. Orange was for the letters I was still figuring out if they were correct or not. When I had about 75% of the letters decrypted, I wrote down the letters that were still available and looked for them in the Ciphertext. For example, I needed to find what letter B was representing so I looked for a B in the text and saw it frequently at the top by the roman numerals and was able to use clues to realize it was X. Finally, when I got to my last two letters I wrote down both possibilities and was able to figure it out quickly with the words that were left with the wrong spelling.

After decryption the ciphertext turned out to be a novel titled The blue castle, and the key was KXWVAHYOQEZJPUGIMLTRCDSFBN. This project was stressful in the sense that I wasn't able to make meaningful progress on the decryption until I was able to get the program to import text correctly, which I struggled with. Then I had to find a way to keep the formatting intact because it was printing as a block of text with no spaces or line breaks, which made it very hard to try and decipher. Once formatting was complete, I thought that the decryption part would be the hardest part, but other than the first few letters, it was easier than I expected. It felt fulfilling to finish the project on time, because I was very concerned that this would not happen.

I have spent some time thinking about how I developed my program vs how others developed theirs, and how they would compare in terms of complexity and

efficiency. I believe my worst case complexity would be $O(n^2)$ in my decryptCipherText() method in my decrypt class. I am curious as to how my implementation stacks up against my peers. Overall, while this was a stressful project, I enjoyed it and learned a lot. I learned how insecure monoalphabetic ciphers are. They look complex on the surface, but once you find a few parts of the key, one discovery leads to another until the plaintext is revealed. Please don't make us try and crack DES next! I enjoy my sanity.