Michael Warren

CSCI 55500

9/12/16

Cryptography – First assignment

Start of assignment

**Assumptions made**

* There was one-to-one mapping of ciphertexts to ciphers the instructor listed on the assignment
* If there were more ciphers mentioned (there were), then there is one-to-one mapping between ciphertexts and subset of ciphers. That means, assumption that there were unused ciphers was made
  + Since there were only two monoalphabetic substitution ciphers, if one ciphertext C used regular substitution cipher, there exists exactly one ciphertext different from C that used affine cipher. (Affine cipher is substitution cipher.)
* There are only two distinct one-character words in the English language.
* If a ciphertext C contains only one or two distinct cipher-character words, that ciphertext was monoalphabetic substitution cipher.

**Progress**

* The night after the assignment opened, I took notes on cryptoanalysis of affine and substitution ciphers.
* I also, because there was much mention of integer operations modulo 26, and matrices (via Hill and Permutation ciphers), wrote up UML diagram for IntegerGroup,IntegerRing that would power the computations.
  + For this reason, and operator overloading, I chose C++
* I thought first thing to do would be to write code to strip out, and add back in, any and all non-alphabetic characters from a string (including whitespace), so I worked on UML diagram for PunctuationOccurrence,PunctuationOccurrenceList, based on LinkedList that I wrote years ago.
* Main difficulty: remembering how inheritance and copy constructors worked in C++, and spending time from many days testing and debugging my code piece-by-piece.

Mid-assignment

**Assumptions made**

* My printouts of the ciphertexts were accurate (my printout for ciphertext6.txt wasn’t).
* abs(IOC – 0.065) <= 0.01 implies substitution or permutation cipher

**Progress**

* After finally wrestling with the codebase for IntegerGroup,IntegerRing, defining behavior, and including all necessary operators (the whole reason these two classes were even written in the first place), I finally move on to devising algorithm for cracking of substitution cipher and affine cipher.
* I had no detailed algorithm for this, only a couple of high-level steps based on what was said in class.
* Upon writing CodeBreaker, with functions that tally characters in text, sort the tally table, form array of characters ranked from most frequent to least frequent, and compute indexOfCoincidence for given ciphertext, I realized I had enough tools to begin trying to decipher the ciphertexts
* I searched, literally by hand, my printout of the ciphertexts for number of distinct single-ciphercharacter words.
  + Based on my assumptions, I inferred that ciphertext2.txt, ciphertext5.txt were substitution ciphers.