Cryptography Homework № 1

Saeed Hedayatian, 97100292 April 27, 2020

Problem 1[1]

In this problem we are asked to break the two-time pad. In order to do this we are given 11 cipher texts, all of which are encrypted using one key. Our approach here, is to try and guess as many bits of the key as possible. We use the property of XOR here, if we XOR two given cipher texts c_1 and c_2, we will obtain the XOR of their corresponding messages, m_1, m_2. From this, and using properties of ascii standard, we will try to find bits of key. So first, we are going to XOR all 11 given cipher texts with each other to get $m_{ij} = m_i \oplus m_j$. Now let us consider the kth character of m_{ij}. If m_i's kth character is space character, $' \, '$, and if m_j's kth character is not a space, then we can understand this by looking at the seventh bit of m_{ij}'s kth character(Note that we can't know whether m_i has a space or m_j. We only know that exactly one of them does). If we found a space at the kth position of m_{ij}, we call $m_{ij}[k]$ a "hit". It is highly unlikely that more than three messages have a space as their kth character. Thus, by counting the number of hits among $m_{ij}[k]$ for $1 \leqslant j \leqslant 11, j \neq i$, we get a measure of how likely it is that m_i's kth character is space. If there are more than six hits, we assume that m_i's kth character is indeed a space, and obtain kth character of the key by XOR.

$$key[k] = c_i[k] \oplus m_i[k] = c_i[k] \oplus space.$$

Using this method, if we decrypt the target message, we will obtain the following message.
"Thm secuet mes-age is Whtn usa|w wsstreíÌ cipher nevir use the key more than once"
It is not hard to guess that the original message should be *"The secret message is: When using a stream cipher, never use the key more than once"*.
We can further test this, by finding the correct key, using the plain text that we guessed above, and decrypting all 11 messages using it. All of them look like a normal english text. So, we have successfully guessed the key. Other messages' decryptions are below.
We can factor the number 15 with quantum computers. We can also factor the number 1
Euler would probably enjoy that now his theorem becomes a corner stone of crypto -
The nice thing about Keeyloq is now we cryptographers can drive a lot of fancy cars
The ciphertext produced by a weak encryption algorithm looks as good as ciphertext
You don't want to buy a set of car keys from a guy who specializes in stealing cars
There are two types of cryptography - that which will keep secrets safe from your l
There are two types of cyptography: one that allows the Government to use brute for
We can see the point where the chip is unhappy if a wrong bit is sent and consumes
A (private-key) encryption scheme states 3 algorithms, namely a procedure for gene
The Concise OxfordDictionary (2006) deï¬-nes crypto as the art of writing o r sol

[1]See `TwoTimePad.java`.

Problem 2[1]

We can use sum of distances between the frequency of each letter in a given cipher text(monograms) and that of a normal english text, as a measure of how close the text's letters are to an ordinary english text. Obviously, this test does not check whether a given cipher text is meaningful or not, it is not a good measure compared to using bigram and trigrams etc. But this test is perfect to determine whether a given cipher text comes from a transposition cipher(in which only the order of letters are changed and not the letters themselves) or not(it could, for example, be a substitution cipher). More formally we calculate

$$\sum_{\beta \in \{a,\ldots,z\}} (F(\beta) - E(\beta))^2.$$

Where $F(\beta)$ and $E(\beta)$ are β's frequency in the cipher text, and the expected frequency of β respectively. This value turns out to be 0.00052 which is very small, indicating that this cipher text probably belongs to some sort of a transposition cipher.

Now, to break the columnar transposition cipher, if the length of keyword used to encrypt the message is not too long(maximum of 8), one way of breaking the cipher is to just try all possible permutations of columns to decrypt the cipher text, and see which one of the results looks like an ordinary english text. In my code, this is done using the function `EnglishLike` which uses bigrams and trigrams to estimate how likely it is for a given string, to be meaningful english. We try all possible key lengths, from 2 to 7 and save all strings that "look like" ordinary english in a file. There were around 80 candidate texts after this stage. After looking at some of the results, we can see that at the end of most of the calculated plain texts, there are a couple 'x' characters, which is unusual. We can guess that probably, character 'x' was used as a padding in case the length of the message was not a multiple of key length. Thus, we further limit our search to only the strings that end with at least two 'x' characters. This limits the number of matches to only 23. After examining them, the decrypted message is found to be the following string(All 23 candidate strings are provided at "result.txt" file located beside the source code).

"*theobscuritywassodifficulttopenetratethat*
mrlorrypickinghiswayoverthewellwornturkeycarpetsupposedmissmane
ttetobeforthemomentinsomeadjacentroomuntilhavinggot
pastthetwotallcandleshesawstandingtoreceivehimbythetablebetw
eenthemandthefireayoungladyofnotmorethanseventeen
inaridingcloakandstillholdingherstrawtravellinghatbyitsribboninher
handashiseyesrestedonashortslightprettyfigure
aquantityofgoldenhairapairofblueeyesthatmethisownwithaninquiringlookan
daforeheadwithasingularcapacityrememberinghowyoung
andsmoothitwasofriftingandknittingitselfintoanexpressionthatwasno
tquiteoneofperplexityorwonderoralarmormerelyofabrightfixedattention
thoughitincludedallthefourexpressionsashiseyesres
tedonthesethingsasuddenvividlikenesspassedbeforehimofachild

[1] See `Problem2.cpp` and `results2.txt`.

whomhehadheldinhisarmsonthepassageacrossthatvery channelonecoldtimewhenthehaildriftedheavilyandthe searanhighthelikenesspassedawaylikeabreathalongthesurfaceofthe gauntpierglassbehindherontheframeofwhichahospital processionofnegrocupidsseveralheadlessandallcrippleswereoffering blackbasketsofdeadseafruittoblackdivinitiesofthe femininegenderandhemadehisformalbowtomissmanettexxx” (The plaint text is part of “A Tale of Two Cities” by Charles Dickens.)

Problem 3[1]

Similar to what we did in the previous problem, we use the frequency of letters in the given cipher text to check whether given cipher text is a substitution cipher or not. Because in a substitution cipher, the order of alphabet is changed, this time we calculate the sum of squares of letter frequencies in the cipher text, and compare it to that of a normal english text. More formally we calculate

$$\sum_{\beta \in \{a \dots z\}} F(\beta)^2.$$

For a normal english text, or for a cipher text obtained from any substitution or transposition cipher, this value should be near 0.065. But for our cipher text this value is about 0.0495 which shows significant deviation from the expected 0.065. Thus, a different type of cipher was used here.

From the discussion above, we conclude that whatever cipher was used, it completely changes the frequency of letters in a text. We suspect that this might be a Vigenere cihper and we try to test this theory.

If a vigenere cipher with a keyword of length d was used, characters at $i \times d + j$ positions(for $0 \leqslant j < d$) together form a cesar cihper with a fixed key. A cesar cipher can easily be broken by testing all possible 26 values of key and calculating the distance between single letter frequencies and normal english letter frequencies each time we use a specific key and selecting the most probable value for the key(This process is exactly like what we did in problem 2 to check if the given cihper text was from transposition cipher). Using the method we just said, we find the most probable key and decrypt the message using that key, for all keys of length $2 \cdots 9$. After looking at the 8 candidate answers, we see that indeed a vinegere cipher with a keyword of “selma” was used to encrypt the following message, which is part of Martin Luther King’s “I have a dream...” speech.
ihaveadreamthatonedaythisnationwillriseupandliveoutthetrue meaningofitscreedweholdthesetruthstobeselfevidentthatallmenarecreated equalihaveadreamthatonedayontheredhillsofgeorgiathesonsofformerslaves andthesonsofformerslaveownerswillbeabletositdowntogetheratthetableofbrotherhood

[1] See `Vigenere.cpp` and `results3.txt`.

Problem 4[1]

We know the primitive characteristic polynomial of the LFSR. Let $key = (s_0, s_1, \cdots, s_{50})$ be the initial value stored in LFSR. We can define a transition matrix A for the LFSR such that for all $n \in \mathbb{W}$,

$$A^n \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{50} \end{bmatrix} = \begin{bmatrix} s_n \\ s_{n+1} \\ \vdots \\ s_{n+50} \end{bmatrix}.$$

Where s_n is the nth bit that the LFSR outputs. In order to create this matrix A, we use the characteristic polynomial $x^{51} + x^9 + 1$. We have

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}.$$

The first 50 rows of A are similar to an identity matrix(if we don't consider the first column entries) and in the last row, all entries are zero except for $A_{51,1}$ and $A_{51,10}$ which are set to one.
Now, let us consider the cipher text C given to us. Each character in C is an 8-bit, alpha-numeric ascii character. Using the properties of ascii codes, it can be seen that the first (most significant) bit in all alpha-numeric ascii characters is 0. Thus, by considering the first bit of the ith character in C, we can obtain the $(8 \times i)$th output bit of LFSR. But the nth output bit of the LFSR is just a linear combination of the 51 bits of key. This linear combination is determined using the first row of $A^n \times key$ if we consider key as a 51×1 column matrix as can be seen below.

$$A^n \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{50} \end{bmatrix} = \begin{bmatrix} & Row1 & \\ & Row2 & \\ & \vdots & \\ & Row51 & \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{50} \end{bmatrix} = \begin{bmatrix} s_n \\ s_{n+1} \\ \vdots \\ s_{n+50} \end{bmatrix}$$

Hence,

$$s_n = \begin{bmatrix} \text{Row 1 of } A^n \end{bmatrix} \cdot \begin{bmatrix} s_0 & s_1 & \dots & s_{50} \end{bmatrix}.$$

Thus, for every character in C we can obtain a linear equation in fifty-one unknowns, $s_0, \dots, s_{50}$. By considering all characters in C, we can create a system of linear equations with unknowns $s_0, \dots, s_{50}$. By solving this system, we obtain key, the initial state of the LFSR. From here, all we have to do is to decrypt the message using key. The result of decryption is the following message.

[1] See `LFSR.java`.

“ hedweltatadistanceofthreequartersofanhourfromthecityfarfrom anyhamletfarfromanyroadinsomehiddenturnofaverywildvalleynoone knewexactlywherehehadthereitwassaidasortoffieldaholealairthe rewerenoneighborsnotevenpassersbysincehehaddweltinthatvalley thepathwhichledthitherhaddisappearedunderagrowthofgrassthe localitywasspokenofasthoughithadbeenthedwellingofahangman”

Which is part of *Les Misérables* by Victor Hugo.