Week 1 - Classical Cryptosystems

MAT260: Cryptology

Gary Hobson

Date 5/11/2025

Introduction:

The assigned computer problems this week are from Chapter 2, Problems 1, 2, 4, and 9.

Make sure to review the example computer problems in "Appendix C.2 Examples for Chapter 2" that work similar problems to those you are assigned.

Make sure to run your code so all relevant computations/results are displayed and then export your work as a PDF file for submission.

Chapter 2 Problems:

Problem 1:I ran ciphertexts and uded the command allshift to get all possible shifts. So, the correct plaintext is "watch out for Brutus"

```
ciphertexts
allshift(ycve)
```

```
ycvejqwvhqtdtwvwu
zdwfkrxwirueuxwxv
aexglsyxjsvfvyxyw
bfyhmtzyktwgwzyzx
cgzinuazluxhxazay
dhajovbamvyiybabz
eibkpwcbnwzjzcbca
fjclqxdcoxakadcdb
gkdmryedpyblbedec
hlenszfeqzcmcfefd
imfotagfradndgfge
jngpubhgsbeoehghf
kohqvcihtcfpfihig
lpirwdjiudgqgjijh
mqjsxekjvehrhkjki
nrktyflkwfisilklj
osluzgmlxgjtjmlmk
ptmvahnmyhkuknmnl
qunwbionzilvlonom
rvoxcjpoajmwmpopn
swpydkqpbknxnqpqo
txqzelrqcloyorqrp
uyrafmsrdmpzpsrsq
vzsbgntsenqaqtstr
```

```
watchoutforbrutus
xbudipvugpscsvuvt
```

Problem 2: I ran ciphertexts and used the command allshift to get all possible shifts. So, the correct plaintext is "Eve expect eggs for breakfast"

```
allshift(lcll)
```

```
lcllewljazlnnzmvyiylhrmhza
mdmmfxmkbamooanwzjzmisniab
nenngynlcbnppboxakanjtojbc
ofoohzomdcoqqcpyblbokupkcd
pgppiapnedprrdqzcmcplvqlde
qhqqjbqofeqsseradndqmwrmef
rirrkcrpgfrttfsbeoernxsnfg
sjssldsqhgsuugtcfpfsoytogh
tkttmetrihtvvhudgqgtpzuphi
uluunfusjiuwwivehrhuqavqij
vmvvogvtkjvxxjwfisivrbwrjk
wnwwphwulkwyykxgjtjwscxskl
xoxxqixvmlxzzlyhkukxtdytlm
ypyyrjywnmyaamzilvlyuezumn
zqzzskzxonzbbnajmwmzvfavno
araatlaypoaccobknxnawgbwop
bsbbumbzqpbddpcloyobxhcxpq
ctccvncarqceeqdmpzpcyidyqr
duddwodbsrdffrenqaqdzjezrs
eveexpectseggsforbreakfast
fwffyqfdutfhhtgpscsfblgbtu
gxggzrgevugiiuhqtdtgcmhcuv
hyhhashfwvhjjvirueuhdnidvw
iziibtigxwikkwjsvfvieojewx
jajjcujhyxjllxktwgwjfpkfxy
kbkkdvkizykmmyluxhxkgqlgyz
```

Problem 4:

```
% if i -> e  & f -> d
% then E(8) = 4 and E(5) = 3
% so, a * 8 + b = 4 mod 26 & a * 5 + b = 3 mod 26
% (8a+b) − (5a+b) ≡ 4−3 mod 26 = 3 * a = 1 mod 26
% Since 27 = 1 mod 26 & 3 * 9 = 27 => a = 9
a = 9;
% so, 9 * 8+b = 4 mod 26 => 72 + b = 4 mod 26
% so, b = 4−72 = −68 mod 26
% −68 + 26 = −42
% −42 + 26 = −16
% −16 + 26 = 10 so,  b = 10
b = 10;
% Since, 9 * a^-1 = 1 mod 26
% 9 * 3 = 27 = 1 mod 26  => a^-1 = 3
a_inv = 3;  % inverse of 9 mod 26
%convert letters to numbers
disp(edsg);
```

```
edsgickxhuklzveqzvkxwkzukcvuh
```

```
y = text2int(edsg);
```

```
disp(y);
```

```
     4     3    18     6     8     2    10    23     7    20    10    11    25    21     4    16    25
```

```
plaintext_nums = mod(a_inv * (y - b), 26);
disp(plaintext_nums);
```

```
     8     5    24    14    20     2     0    13    17     4     0     3    19     7     8    18    19
```

```
% convert numbers to letters
plaintext = int2text(plaintext_nums);
disp(plaintext);
```

```
ifyoucanreadthisthankateacher
```

Problem 9:

```
% Problem 9 Code Here

% Vigenère Cipher Decryption using frequency correlation
% Based on Appendix C.2 style

% Define normalized English letter frequencies (a to z)
freqs = [0.08167 0.01492 0.02782 0.04253 0.12702 0.02228 0.02015 0.06094 ...
         0.06966 0.00153 0.00772 0.04025 0.02406 0.06749 0.07507 0.01929 ...
         0.00095 0.05987 0.06327 0.09056 0.02758 0.00978 0.02360 0.00150 ...
         0.01974 0.00074];

% Set the key length
m = 6;

% Initialize the key vector
key = zeros(1, m);

% For each position modulo m, compute the shift with best correlation
for i = 1:m
    max_corr = -inf;
    best_shift = 0;

    % Try all 26 shifts and compute correlation with English frequency
    for shift = 0:25
        vec = circshift(vigvec(ocwy, m, i), -shift);
        c = corrcoef(vec, freqs);
        if c(1,2) > max_corr
            max_corr = c(1,2);
            best_shift = shift;
        end
    end

    % Store the negative shift for decryption
    key(i) = -best_shift;
end
```

```
% Decrypt using the computed key
plaintext = vigenere(ocwy, key);

% Decrypted Plaintext:
disp(plaintext);
```

```
holmeshadbeenseatedforsomehoursinsilencewithhislongthinbackcurvedoverachemicalvesselinwhichhewasbrewingapa
```

```
% Decryption Key (letters):
disp(int2text(mod(-key, 26)));
```

```
holmes
```

```
% Decryption Key (numeric):
disp(key)
```

```
    -7   -14   -11   -12    -4   -18
```