ADEEN AYUB

APHICHAYA PIYAPINANSOOK

RONY XAVIER

CS6903 PROJECT 1

CRYPTANALYSIS BASED ON UNIT FREQUENCY CHARACTERS IN THE KEY

This team consists of Adeen Ayub, Aphichaya Piyapinansook, and Rony Xavier. We are submitting one cryptanalysis approach as follows:

CRYPTANALYSIS APPROACH FOR TEST1

The cryptanalysis approach we used in our program utilizes the fact that seven of the alphabets ('b', 'j', 'k', 'v', 'q', 'x', 'z') have the average frequency of one, which means that their corresponding ciphertext will be same and repeated. For example, the letter j has an average frequency of 1 and the key value would be k(j,1), for any randomly chosen key, Its ciphertext will always be identical.

We pre-process the known plaintexts to map out occurrences of all the unit frequency characters. In other words, we collected and saved all the known unit frequency character positions that help us determine which one of the five plaintexts is given as ciphertexts.

The decryption scheme will work regardless of the scheduling algorithm used in the encryption scheme. The Scheduling algorithm does not play a crucial role in this decryption scheme due to the weakness that several alphabets only have an average frequency of 1. This results in those letters being mapped to the same number.

IMPLEMENTATION FOR TEST1

As stated above, the decryption scheme preprocesses the known plaintexts to map the occurrences of all the unit frequency characters using the method GENERATE_UNIT_FREQ_CHAR_POSITIONS.

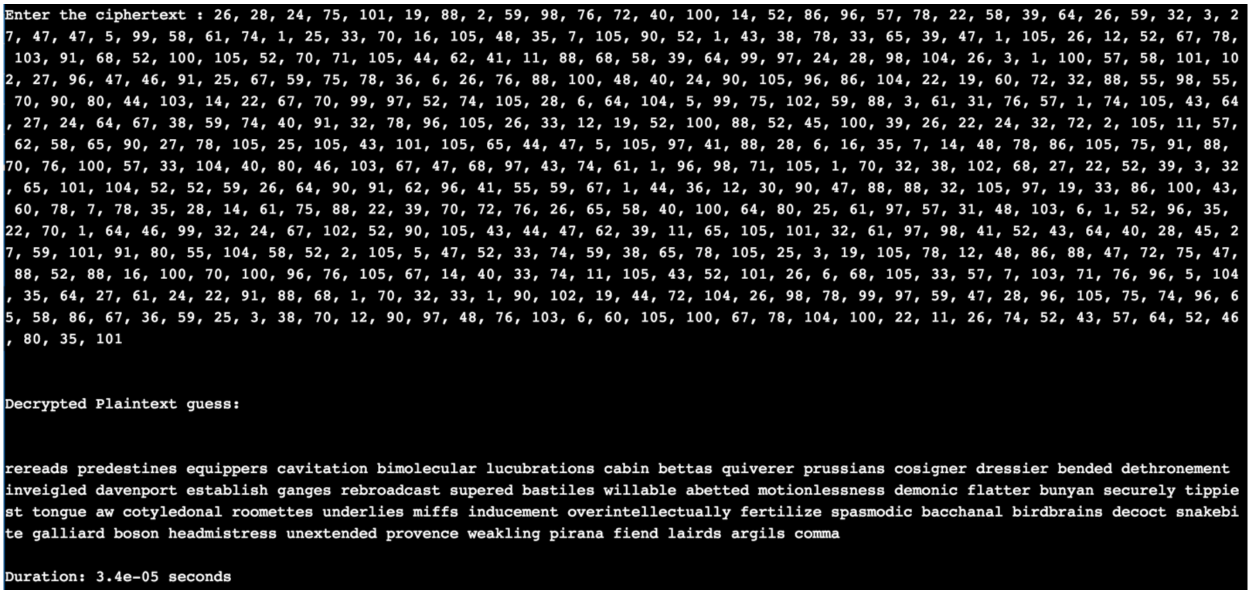
```
procedure GENERATE_UNIT_FREQ_CHAR_POSITIONS
    PLAINTEXT_LIST ← List of known plaintext strings
    UNIT_FREQUENCY_CHARS ← List of unit frequency chars { 'b', 'j', 'k', 'v', 'q', 'x', 'z' }
    for each string plaintext in the vector PLAINTEXT_LIST do
        for each character char in vector UNIT_FREQUENCY_CHARS do
            for each character letter in string plaintext do
                if char == letter do
                    insert into vector positionlist index of letter in string plaintext
                end if
            end for
            insert into map unitFreqCharPositions vector positionlist at key char
            clear vector positionlist
        end for
        insert into vector unitFreqCharPositionsList map unitFreqCharPositions
        clear map unitFreqCharPositions
    end for
    return unitFreqCharPositionsList
end procedure
```

The ciphertext is received from the user from STDIN as series of integers delimited by commas. The method DECRYPT_CIPHERTEXT shown below performs the decryption and returns an index which gives the guessed plaintext or returns ‘-1’ indicating failure to decrypt.

```
procedure DECRYPT_CIPHERTEXT
    CIPHERTEXT ← vector of integers that form the provided ciphertext
    unitFreqCharPositionsList ← GENERATE_UNIT_FREQ_CHAR_POSITIONS ()

    for each map unitFreqCharPositions in vector unitFreqCharPositionsList  do
        for each key char in map unitFreqCharPositions do
            vector positions ← unitFreqCharPositions value at key char
            for each integer position in vector positions do
                insert into vector list values of CIPHERTEXT at index position
            end for
        if length of list > 1 AND all elements of list are equal
            /* return index which is the index of guessed plaintext */
            return index of unitFreqCharPositions in unitFreqCharPositionsList
        end if
        clear vector list
        end for
    end for
    return  -1 /* indicating failure */
end procedure
```

TRIAL RUN FOR TEST1:



```
Enter the ciphertext : 26, 28, 24, 75, 101, 19, 88, 2, 59, 98, 76, 72, 40, 100, 14, 52, 86, 96, 57, 78, 22, 58, 39, 64, 26, 59, 32, 3, 2
7, 47, 47, 5, 99, 58, 61, 74, 1, 25, 33, 70, 16, 105, 48, 35, 7, 105, 90, 52, 1, 43, 38, 78, 33, 65, 39, 47, 1, 105, 26, 12, 52, 67, 78,
 103, 91, 68, 52, 100, 105, 52, 70, 71, 105, 44, 62, 41, 11, 88, 68, 58, 39, 64, 99, 97, 24, 28, 98, 104, 26, 3, 1, 100, 57, 58, 101, 10
2, 27, 96, 47, 46, 91, 25, 67, 59, 75, 78, 36, 6, 26, 76, 88, 100, 48, 40, 24, 90, 105, 96, 86, 104, 22, 19, 60, 72, 32, 88, 55, 98, 55,
 70, 90, 80, 44, 103, 14, 22, 67, 70, 99, 97, 52, 74, 105, 28, 6, 64, 104, 5, 99, 75, 102, 59, 88, 3, 61, 31, 76, 57, 1, 74, 105, 43, 64
, 27, 24, 64, 67, 38, 59, 74, 40, 91, 32, 78, 96, 105, 26, 33, 12, 19, 52, 100, 88, 52, 45, 100, 39, 26, 22, 24, 32, 72, 2, 105, 11, 57,
 62, 58, 65, 90, 27, 78, 105, 25, 105, 43, 101, 105, 65, 44, 47, 5, 105, 97, 41, 88, 28, 6, 16, 35, 7, 14, 48, 78, 86, 105, 75, 91, 88,
70, 76, 100, 57, 33, 104, 40, 80, 46, 103, 67, 47, 68, 97, 43, 74, 61, 1, 96, 98, 71, 105, 1, 70, 32, 38, 102, 68, 27, 22, 52, 39, 3, 32
, 65, 101, 104, 52, 52, 59, 26, 64, 90, 91, 62, 96, 41, 55, 59, 67, 1, 44, 36, 12, 30, 90, 47, 88, 88, 32, 105, 97, 19, 33, 86, 100, 43,
 60, 78, 7, 78, 35, 28, 14, 61, 75, 88, 22, 39, 70, 72, 76, 26, 65, 58, 40, 100, 64, 80, 25, 61, 97, 57, 31, 48, 103, 6, 1, 52, 96, 35,
22, 70, 1, 64, 46, 99, 32, 24, 67, 102, 52, 90, 105, 43, 44, 47, 62, 39, 11, 65, 105, 101, 32, 61, 97, 98, 41, 52, 43, 64, 40, 28, 45, 2
7, 59, 101, 91, 80, 55, 104, 58, 52, 2, 105, 5, 47, 52, 33, 74, 59, 38, 65, 78, 105, 25, 3, 19, 105, 78, 12, 48, 86, 88, 47, 72, 75, 47,
 88, 52, 88, 16, 100, 70, 100, 96, 76, 105, 67, 14, 40, 33, 74, 11, 105, 43, 52, 101, 26, 6, 68, 105, 33, 57, 7, 103, 71, 76, 96, 5, 104
, 35, 64, 27, 61, 24, 22, 91, 88, 68, 1, 70, 32, 33, 1, 90, 102, 19, 44, 72, 104, 26, 98, 78, 99, 97, 59, 47, 28, 96, 105, 75, 74, 96, 6
5, 58, 86, 67, 36, 59, 25, 3, 38, 70, 12, 90, 97, 48, 76, 103, 6, 60, 105, 100, 67, 78, 104, 100, 22, 11, 26, 74, 52, 43, 57, 64, 52, 46
, 80, 35, 101


Decrypted Plaintext guess:


rereads predestines equippers cavitation bimolecular lucubrations cabin bettas quiverer prussians cosigner dressier bended dethronement
inveigled davenport establish ganges rebroadcast supered bastiles willable abetted motionlessness demonic flatter bunyan securely tippie
st tongue aw cotyledonal roomettes underlies miffs inducement overintellectually fertilize spasmodic bacchanal birdbrains decoct snakebi
te galliard boson headmistress unextended provence weakling pirana fiend lairds argils comma

Duration: 3.4e-05 seconds
```

CONCLUSION

The decryption scheme could effectively decrypt ciphertexts of known plaintexts thus the encryption scheme fails the *indistinguishability* notion. The proposed scheme does not address test2.