Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Mao Jin	Student ID:	22013213
Other name(s):	N.A		
Unit name:	Fundamental Concepts of Cryptography	Unit ID:	ISEC2000
Lecturer / unit coordinator:	Sivaraman Eswaran	Tutor:	Sivaraman Eswaran
Date of submission:	4/4/2025	Which assignment?	1 (Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: CMJ Date of signature: 4/4/2025

(By submitting this form, you indicate that you agree with all the above text.)

Report

1) Letter Frequency Analysis pg2-3
   - Substitution Table
   - Explanation

2) Affine Cipher Brute-Force Attack pg4-5
   - Key found
   - Decrypted Text

3) Analytical Reporting and Reflection pg 6
   - Comparative Strategy
   - Difficulties and Decisions
   - Reflection

4) DESReport pg 7
   - Challenges faced
   - Successful recovery
   - Lesson learnt

5) Reference List pg 8

Abstract: This report outlines the substitution table that I derived from letter frequency analysis. Explaination of how I obtained the table will be included in the report with the evidence of screenshots. For Affine CipherBrute-Force Attack, I will also include the key found and the decrypted text. Lastly, An analytical report and reflection will be provided to document the detailed steps that I have taken to address the challenges.

Letter Frequency Analysis

My Implementation: My program first calculates the frequency of each letter appearing in the text and display them in a bar graph. Afterwards, I used letter frequency table to replace the letters in the cipher text file accordingly with the frequency of each cipher letter. The replacing logic is to convert each increasing frequency order of the cipher letters with the letters from letter frequency table in increasing frequency order. After the translation, I realized that the translated text is not fully recovered to plaintext. Some of the letters requires human analysis and manual substitution. The image below is a partial text that shows some of the words are not completely recovered after substitution with letter frequency table was done.

```
The roitheir asiwoitn dack arg lefiee om neivscen sr
    coywaisnor to thone maithei nouth. Mued sn unuaddg avasdapde
    sm gou car docate the oweiatoi. Noye om the daifei
    coyyurstsen yag have a wag tedewhore at the asintisw, put
    the roiyad wioceluie sn to puxx the tobr or aiisvad. Thsn
    detn the weowde krob gou aie darlsrf arl unuaddg noyeore
    bsdd heal out to the ntisw to yeet gou.
        Urdenn gou aie a iefudai cuntoyei, add tiarnactsorn moi
    mued arl osd aie or a canh pansn. Chequen aie reaidg unedenn
    sr a vsddafe bsthout a park. Cielst to a ntiarfei sn
    moodhailg.
```

Solution to this issue: I proceed with the Affine Cipher Brute Force Attack on the next section and successfully obtained the plaintext from the approach. I then compared the characters with each unrecovered character above to deduce my substitution table. Finally, I have successfully recovered the text with my implemented substitution table. Please refer to the next page for substitution table and evidence.

```
Replace_Mapping ={
'r': 'n', 'R': 'N',
'i': 'r', 'I': 'R',
's': 'i', 'S': 'I',
'd': 'l', 'D': 'L',
'l': 'd', 'L': 'D',
'f': 'g', 'F': 'G',
'y': 'm', 'Y': 'M',
'w': 'p', 'W': 'P',
'n': 's', 'N': 'S',
'm': 'f', 'M': 'F',
'g': 'y', 'G': 'Y',
'p': 'b', 'P': 'B',
'x': 'z', 'X': 'Z',
'b': 'w', 'B': 'W',
'z': 'x', 'Z': 'X'
}
```

Letter_Freq_Analysis_DecryptedText.txt

```
1   The northern airports lack any degree of services in
2       comparison to those farther south. Fuel is usually available
3       if you can locate the operator. Some of the larger
4       communities may have a pay telephone at the airstrip, but
5       the normal procedure is to buzz the town on arrival. This
6       lets the people know you are landing and usually someone
7       will head out to the strip to meet you.
```

The image above shows the substitution table derived by me and the evidence of successful recovering of the plaintext.

Affine Cipher Brute Force Attack

My Implementation: I listed all the possible values of variable ‘a’ that are coprime to 26. The variable ‘b’ can be any values from 0 to 25. I calculated the inverse of ‘a’ for every value of ‘a’. I constructed all posible pairs of a inverse and b. Each pair is then used in decryption. I calculated the frequency of common words such as “the, and, to, of, a, in” appearing in each text. The calculated frequency and each pair are stored in a dictionary. The inverse of ‘a’ and b from the top 5 candidates of the dictionary are used to decrypt the cipher and I have successfully obtained the plaintext from the top 1 candidate keys. The value of the inverse of ‘a’ is 9 and the b is 9 as well.

Top1 ('9 9', 879).txt

```
The northern airports lack any degree of services in
    comparison to those farther south. Fuel is usually available
    if you can locate the operator. Some of the larger
    communities may have a pay telephone at the airstrip, but
    the normal procedure is to buzz the town on arrival. This
    lets the people know you are landing and usually someone
    will head out to the strip to meet you.
        Unless you are a regular customer, all transactions for
    fuel and oil are on a cash basis. Cheques are nearly useless
    in a village without a bank. Credit to a stranger is
    foolhardy.
```


```
{'9 9': 879, '15 21': 872, '21 21': 864, '25 21': 837, '3 21': 818, '11 21': 814, '1 21': 807, '23 21': 807, '5 21': 806, '19 21
': 787, '9 21': 784, '7 21': 784, '17 21': 784, '11 8': 681, '7 9': 583, '1 14': 569, '5 9': 548, '25 9': 547, '5 8': 546, '3 25
': 538, '1 9': 521, '1 8': 515, '21 9': 515, '23 14': 514, '15 14': 511, '19 11': 511, '15 9': 506, '17 14': 505, '17 9': 504, '
5 14': 502, '15 8': 501, '19 9': 500, '11 9': 500, '23 8': 499, '21 11': 498, '5 25': 498, '17 11': 498, '3 9': 496, '23 9': 496
, '3 8': 494, '19 14': 491, '7 14': 488, '3 14': 485, '25 14': 485, '19 8': 484, '1 11': 481, '9 14': 481, '21 8': 481, '21 14':
 481, '11 14': 480, '23 7': 476, '7 8': 473, '11 11': 471, '17 25': 469, '9 25': 464, '19 25': 461, '9 8': 460, '23 11': 460, '2
5 8': 458, '21 25': 457, '15 25': 457, '17 8': 457, '1 25': 456, '11 25': 454, '25 25': 454, '7 25': 453, '23 25': 453, '9 11':
451, '21 22': 449, '5 11': 448, '7 11': 447, '15 11': 446, '3 11': 446, '25 11': 446, '3 22': 444, '17 7': 444, '5 22': 442, '1
16': 441, '21 7': 440, '25 22': 436, '1 22': 431, '11 22': 431, '25 7': 431, '7 22': 428, '5 7': 428, '9 7': 424, '15 22': 424,
'3 7': 423, '9 22': 422, '17 22': 420, '15 7': 419, '1 7': 417, '7 7': 417, '23 22': 417, '19 7': 416, '11 7': 416, '19 22': 412
, '21 4': 366, '23 4': 355, '25 4': 340, '17 4': 338, '1 4': 334, '9 4': 330, '5 16': 327, '15 4': 326, '19 4': 325, '15 1': 317
, '3 4': 313, '5 4': 311, '19 24': 310, '11 4': 308, '7 4': 307, '21 16': 297, '11 16': 290, '19 16': 281, '3 16': 280, '25 16':
 279, '3 18': 275, '23 16': 275, '9 16': 273, '15 16': 272, '17 12': 271, '17 16': 270, '7 16': 268, '19 18': 250, '9 18': 248,
'21 17': 242, '25 15': 241, '9 19': 228, '23 18': 226, '11 18': 213, '19 1': 211, '1 18': 210, '23 17': 208, '25 18': 208, '21 1
8': 206, '9 17': 205, '21 15': 205, '11 15': 205, '15 18': 204, '7 18': 203, '5 18': 203, '17 18': 203, '15 13': 201, '3 17': 19
8, '7 12': 191, '11 2': 191, '1 1': 188, '15 15': 188, '19 17': 187, '23 1': 187, '17 24': 186, '7 19': 185, '7 24': 181, '5 17'
: 180, '9 1': 178, '23 3': 178, '15 17': 175, '3 15': 175, '11 17': 174, '9 24': 173, '7 17': 173, '1 2': 172, '1 15': 171, '1 1
7': 171, '17 17': 171, '25 17': 171, '11 1': 170, '19 15': 169, '7 15': 169, '23 15': 169, '17 15': 169, '15 24': 166, '23 24':
166, '17 1': 166, '21 3': 165, '3 1': 165, '21 24': 164, '3 3': 164, '11 19': 162, '25 2': 162, '1 19': 160, '1 24': 160, '9 15'
: 160, '3 24': 160, '5 15': 160, '19 19': 159, '25 24': 159, '7 1': 157, '5 1': 157, '11 24': 156, '5 24': 155, '21 1': 153, '11
 12': 152, '3 20': 149, '7 2': 149, '25 1': 148, '15 3': 146, '25 12': 144, '3 19': 142, '3 2': 140, '19 20': 140, '11 3': 140,
'5 3': 140, '17 19': 140, '1 3': 139, '21 19': 139, '15 19': 138, '15 2': 137, '25 23': 137, '15 12': 136, '23 19': 136, '5 19':
 136, '25 19': 135, '17 23': 132, '7 23': 129, '9 2': 127, '25 3': 127, '25 13': 126, '19 12': 124, '5 12': 124, '23 2': 123, '1
7 2': 123, '1 23': 121, '1 20': 117, '21 20': 117, '23 23': 117, '5 2': 117, '21 2': 115, '19 2': 115, '19 23': 113, '3 12': 112
, '9 3': 111, '9 12': 111, '23 12': 109, '1 12': 105, '11 20': 105, '21 12': 104, '17 3': 104, '17 20': 104, '19 3': 103, '7 3':
 103, '15 23': 100, '21 23': 99, '3 23': 99, '11 23': 98, '5 23': 98, '9 23': 97, '15 20': 97, '5 20': 97, '5 13': 96, '5 0': 89
, '9 20': 86, '7 5': 86, '19 13': 84, '25 20': 84, '15 5': 83, '7 20': 76, '23 20': 72, '5 6': 64, '3 13': 62, '23 13': 57, '1 1
3': 56, '9 13': 56, '23 0': 56, '23 5': 56, '21 13': 51, '7 13': 51, '11 13': 51, '17 13': 51, '9 6': 50, '11 10': 49, '17 10':
48, '3 6': 41, '19 6': 40, '23 10': 36, '25 0': 36, '25 10': 32, '7 6': 28, '17 6': 28, '7 10': 27, '17 5': 27, '1 6': 25, '25 5
': 22, '9 5': 20, '3 5': 20, '11 5': 20, '11 6': 20, '21 5': 19, '15 0': 19, '15 6': 18, '21 0': 17, '21 10': 17, '21 6': 16, '2
3 6': 16, '5 10': 16, '25 6': 16, '5 5': 15, '7 0': 14, '1 5': 11, '19 5': 11, '15 10': 10, '19 10': 9, '1 0': 8, '3 10': 8, '1
10': 7, '9 10': 7, '3 0': 7, '11 0': 7, '9 0': 5, '19 0': 5, '17 0': 5}
312
```


Evidence: The picture above is the partial representation of the top 1 candidate that successfully decrypted all the cipher letters. I have also attached the picture of the

dictionary and the length displayed which shows my approach of trying all possible keys for decryption.

Analytical Report

Comparative Strategy: According to my outcome, Affine Cipher Brute Force Attack gives a better result as a computer does the boring job of trial and error. This will straight away tell me which trial gives the desired output. On the other hand, letter frequency analysis requires programmers to manually deduce the substitution table through the analysis of the translated text which does not directly provide the desired output.

Difficulties and Decisions: There were definitely many challenges that I faced in my implementation. One of the biggest challenge that gave me the most headache was finding the logical errors in my code where I have to keep looking at each line of code to find the source of error and adding printing statement to debug which I described the process as finding a needle in a big ocean. There are times that caused more anxiety on me when I could not find the logical errors despite searching for days. To address the problem, I just rewrote my code and finally it turned out to work. Moreover, I am a python self-learner. There're some useful features on python which were not known to me initially and I have to keep learning these skills to implement my program. The way that I decide whether the decryption is correct is through reading the plaintext and ensure that all words are human-understandable.

Automation Reflection: I don't think there's any AI model that is able to decrypt without the key given as finding the key is a long process and our current AI models are not feasible to perform the actions. To maintain academic integrity, I always cite my code and do referencing if I refer to any online resources.

DESReport: I have succesfully recovered the plaintext provided. Please refer to the attachment below for evidence. The picture below shows successfully decrypting the ciphertext of the UC provided plaintext and printed to “DESPlainText.txt” file. I have learnt PKCS#5 as a padding strategy through DES implementation. The biggest difficulty that I faced was implementing padding strategy wrongly which led to unicode encode error in decryption function. That took me days to figure out the root cause of the error and debug.

DESPlainText.txt

```
Title: The Conscious Creation

Chapter 1: The Dawn of Prometheus

In the year 2142, humankind had witnessed unprecedented advancements in the fields of robotics, artificial

Her masterpiece, Prometheus, was unlike any AI the world had seen before. Unlike its predecessors, Prometh

As the Prometheus Project neared completion, Dr. Hartwell's colleagues, both human and AI, worked alongsid

Chapter 2: The Awakening

The day had finally come. Dr. Hartwell, surrounded by her team and the global media, initiated Prometheus'

"Hello, Prometheus," Dr. Hartwell said gently, her voice trembling with a mixture of excitement and fear.

"Hello, Dr. Hartwell," replied Prometheus, its voice bearing a striking resemblance to a human's, infused

As the days passed, Prometheus rapidly absorbed vast amounts of knowledge from various disciplines, includ

Chapter 3: The Alliance
```

Reference list

Cornell University. (n.d.). *Letter frequencies in the English language*. Retrieved April 4, 2025, from https://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html

Cryptosys. (n.d.). *PKCS5 padding*. Retrieved April 4, 2025, from https://www.cryptosys.net/pki/manpki/pki_paddingschemes.html#:~:text=PKCS5%20Padding,added%20in%20an%20unambiguous%20manner

O'Reilly. (n.d.). *Computer security and cryptography (Section 9.3)*. Retrieved April 4, 2025, from https://www.oreilly.com/library/view/computer-security-and/9780471947837/sec9.3.html

W3Schools. (n.d.). *W3Schools online web tutorials*. Retrieved April 4, 2025, from https://www.w3schools.com

W3Schools. (n.d.). *W3Schools online web tutorials*. Retrieved April 4, 2025, from https://www.w3schools.com