

## EE3731C Programming Assignment Report

Name: Lim Yu Guang

Matriculation Number: A0172618B

### Q1. Mapping Between Character and Double Arrays

```
>> NumericArray = double('Mary is good at math.')
NumericArray =
    77    97   114   121    32   105   115    32   103   111   111   100    32    97   116    32   109    97   116   104    46

>> CharacterArray = char(NumericArray)
CharacterArray =
    'Mary is good at math.'

>> CharacterArray = char([80 114 111 98 108 101 109 32 105 115 32 115 111 108 118 101 100 46])
CharacterArray =
    'Problem is solved.'
```

(a) **Figure 1. MATLAB results for question 1(a)**

```
Command Window
>> char2double('Mary is good at math.')
ans =
    13     1    18    25    27     9    19    27     7    15    15     4    27     1    20    27    13     1    20     8    27

fx >> |
```

(b) **Figure 2. MATLAB results for question 1(b)**

```
Command Window
>> double2char([20 15 15 27 13 1 14 25 27 2 21 7 19])
ans =
    'too many bugs'

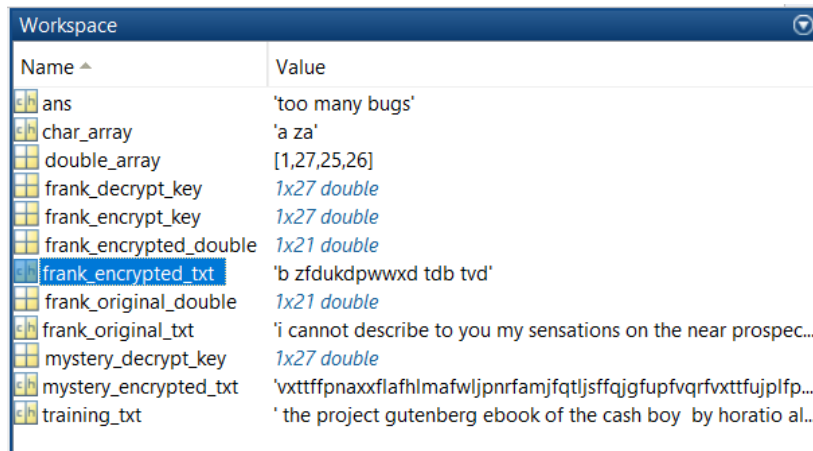
fx >> |
```

(c) **Figure 3. MATLAB results for question 1(c)**

### Q2. Encrypting/Decrypting A Message

```
Command Window
>> frank_original_double = char2double('Mary is good at math.');
```

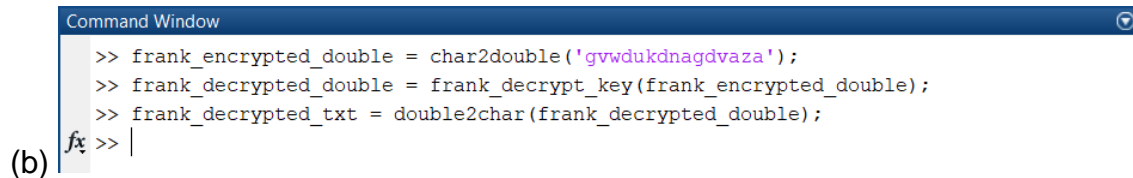
(a) **Figure 4. MATLAB commands to encrypt 'Mary is a good at math.'**



Name	Value
ans	'too many bugs'
char_array	'a za'
double_array	[1,27,25,26]
frank_decrypt_key	1x27 double
frank_encrypt_key	1x27 double
frank_encrypted_double	1x21 double
frank_encrypted_txt	'b zfdukdpwwxd tdb tvd'
frank_original_double	1x21 double
frank_original_txt	'i cannot describe to you my sensations on the near prospec...
mystery_decrypt_key	1x27 double
mystery_encrypted_txt	'vxttffpnaxxflafhlmafwljpnrfamjfqtljsffqjgufupfvqrfvxttfujplfp...
training_txt	'the project gutenber ebook of the cash boy by horatio al...

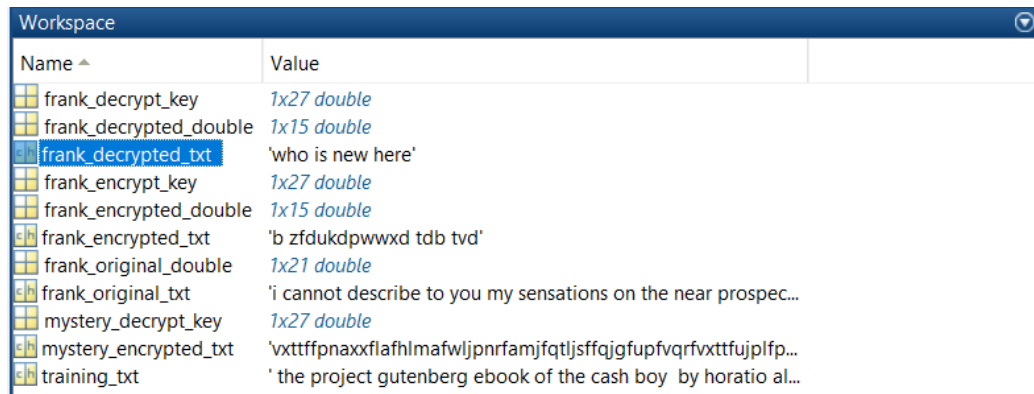
**Figure 5. MATLAB results for question 2(a)**

With reference to figure 5, the result for question 2(a) is 'b zfdukdpwwxd tdb tvd'.



```
>> frank_encrypted_double = char2double('gvwdukdnagdvaza');
>> frank_decrypted_double = frank_decrypt_key(frank_encrypted_double);
>> frank_decrypted_txt = double2char(frank_decrypted_double);
fx >> |
```

(b) **Figure 6. MATLAB commands to decrypt 'gvwdukdnagdvaza'**



Name	Value
frank_decrypt_key	1x27 double
frank_decrypted_double	1x15 double
frank_decrypted_txt	'who is new here'
frank_encrypt_key	1x27 double
frank_encrypted_double	1x15 double
frank_encrypted_txt	'b zfdukdpwwxd tdb tvd'
frank_original_double	1x21 double
frank_original_txt	'i cannot describe to you my sensations on the near prospec...
mystery_decrypt_key	1x27 double
mystery_encrypted_txt	'vxttffpnaxxflafhlmafwljpnrfamjfqtljsffqjgufupfvqrfvxttfujplfp...
training_txt	'the project gutenber ebook of the cash boy by horatio al...

**Figure 7. MATLAB results for question 2(b)**

With reference to figure 7, the result for question 2(b) is 'who is new here'.

### Q3. Probability of Consecutive Characters

(a)

pr\_trans =

Columns 1 through 7

0.0001	0.0209	0.0429	0.0575	0.0003	0.0080	0.0178
0.0396	0.0010	0.0005	0.0010	0.3178	0.0005	0.0005
0.1323	0.0003	0.0252	0.0003	0.1929	0.0003	0.0003
0.0247	0.0002	0.0004	0.0082	0.1202	0.0017	0.0028
0.0438	0.0026	0.0265	0.0811	0.0366	0.0090	0.0033
0.0650	0.0004	0.0004	0.0004	0.0632	0.0284	0.0004
0.0382	0.0008	0.0004	0.0004	0.1400	0.0004	0.0024
0.1961	0.0011	0.0001	0.0009	0.4085	0.0007	0.0001
0.0079	0.0098	0.0385	0.0645	0.0253	0.0204	0.0194
0.0927	0.0028	0.0028	0.0028	0.3034	0.0028	0.0028
0.0012	0.0006	0.0006	0.0006	0.2969	0.0031	0.0006
0.0816	0.0063	0.0022	0.0712	0.1644	0.0204	0.0012
0.1365	0.0106	0.0005	0.0003	0.2542	0.0035	0.0003
0.0204	0.0111	0.0375	0.1372	0.0778	0.0053	0.1074
0.0049	0.0070	0.0085	0.0121	0.0032	0.0584	0.0017
0.1029	0.0004	0.0004	0.0013	0.1918	0.0004	0.0044
0.0080	0.0080	0.0080	0.0080	0.0080	0.0080	0.0080
0.1077	0.0006	0.0060	0.0183	0.2012	0.0018	0.0213
0.0729	0.0013	0.0128	0.0003	0.1206	0.0032	0.0007
0.0357	0.0007	0.0017	0.0002	0.0920	0.0017	0.0004
0.0133	0.0164	0.0297	0.0135	0.0192	0.0017	0.0389
0.0328	0.0008	0.0008	0.0008	0.7689	0.0008	0.0008
0.2087	0.0009	0.0006	0.0016	0.1202	0.0031	0.0003
0.0226	0.0045	0.1448	0.0045	0.0362	0.0045	0.0045
0.0028	0.0014	0.0003	0.0003	0.0417	0.0006	0.0003
0.0351	0.0175	0.0175	0.0175	0.4035	0.0175	0.0175
0.0667	0.0282	0.0231	0.0218	0.0117	0.0300	0.0170

Columns 8 through 14

0.0007	0.0574	0.0002	0.0163	0.0563	0.0279	0.2012
0.0005	0.0139	0.0074	0.0005	0.0931	0.0010	0.0010
0.1348	0.0435	0.0003	0.0506	0.0632	0.0003	0.0003
0.0002	0.0725	0.0006	0.0002	0.0204	0.0017	0.0082
0.0005	0.0069	0.0003	0.0070	0.0343	0.0193	0.0763
0.0004	0.0711	0.0004	0.0004	0.0086	0.0004	0.0004
0.1125	0.0489	0.0004	0.0004	0.0179	0.0012	0.0103
0.0001	0.1571	0.0001	0.0001	0.0004	0.0007	0.0160
0.0001	0.0018	0.0001	0.0059	0.0570	0.0466	0.2250
0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028
0.0006	0.0453	0.0006	0.0006	0.0081	0.0012	0.0820
0.0004	0.0875	0.0002	0.0059	0.1796	0.0020	0.0010
0.0003	0.0568	0.0003	0.0003	0.0019	0.0090	0.0019
0.0009	0.0303	0.0015	0.0579	0.0096	0.0002	0.0083
0.0126	0.0062	0.0080	0.0113	0.0229	0.0615	0.1290
0.0258	0.0460	0.0004	0.0004	0.1020	0.0018	0.0004
0.0080	0.0080	0.0080	0.0080	0.0080	0.0080	0.0080
0.0065	0.0463	0.0001	0.0209	0.0063	0.0152	0.0185
0.0539	0.0634	0.0001	0.0138	0.0078	0.0061	0.0044
0.2744	0.0570	0.0001	0.0001	0.0115	0.0061	0.0011
0.0002	0.0185	0.0002	0.0002	0.0733	0.0114	0.1027
0.0008	0.1369	0.0008	0.0008	0.0008	0.0008	0.0008
0.1934	0.1760	0.0003	0.0009	0.0171	0.0003	0.0227
0.0136	0.1176	0.0045	0.0045	0.0045	0.0045	0.0045
0.0003	0.0092	0.0003	0.0003	0.0011	0.0062	0.0025
0.0175	0.0702	0.0175	0.0175	0.0526	0.0175	0.0175
0.0527	0.0542	0.0045	0.0042	0.0139	0.0345	0.0157

Columns 15 through 21

0.00001	0.0172	0.00001	0.0984	0.0984	0.1200	0.0064
0.1703	0.0005	0.0005	0.0891	0.0099	0.0178	0.1594
0.1858	0.0003	0.0026	0.0194	0.0006	0.0997	0.0245
0.0811	0.0002	0.0002	0.0159	0.0165	0.0002	0.0114
0.0011	0.0129	0.0016	0.1361	0.0479	0.0285	0.0006
0.2024	0.0004	0.0004	0.1748	0.0007	0.0370	0.0373
0.1010	0.0004	0.0004	0.1042	0.0135	0.0048	0.0481
0.0904	0.0001	0.0001	0.0064	0.0015	0.0280	0.0049
0.0448	0.0028	0.0001	0.0335	0.1351	0.1164	0.0008
0.3876	0.0028	0.0028	0.0169	0.0028	0.0028	0.1376
0.0006	0.0006	0.0006	0.0006	0.0460	0.0006	0.0006
0.0640	0.0082	0.0002	0.0020	0.0092	0.0115	0.0168
0.0737	0.0385	0.0003	0.0781	0.0218	0.0003	0.0377
0.0768	0.0007	0.0015	0.0004	0.0299	0.0826	0.0057
0.0404	0.0152	0.0001	0.1030	0.0188	0.0453	0.1975
0.1467	0.0670	0.0004	0.1462	0.0158	0.0350	0.0250
0.0080	0.0080	0.0080	0.0080	0.0080	0.0080	0.7920
0.0910	0.0046	0.0001	0.0138	0.0477	0.0522	0.0081
0.0534	0.0187	0.0008	0.0001	0.0331	0.1040	0.0335
0.1314	0.0008	0.0001	0.0251	0.0098	0.0160	0.0106
0.0007	0.0465	0.0002	0.1310	0.1186	0.1535	0.0002
0.0393	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
0.1062	0.0003	0.0003	0.0090	0.0093	0.0003	0.0003
0.0045	0.2443	0.0045	0.0045	0.0045	0.1946	0.0181
0.3174	0.0011	0.0003	0.0042	0.0252	0.0076	0.0003
0.0175	0.0175	0.0175	0.0175	0.0175	0.0175	0.0175
0.0316	0.0198	0.0011	0.0130	0.0512	0.0912	0.0070

Columns 22 through 27

0.0371	0.0074	0.0009	0.0339	0.0005	0.0719
0.0010	0.0005	0.0005	0.0584	0.0005	0.0134
0.0003	0.0003	0.0003	0.0032	0.0003	0.0177
0.0043	0.0021	0.0002	0.0066	0.0002	0.5993
0.0127	0.0090	0.0087	0.0126	0.0001	0.3807
0.0004	0.0004	0.0004	0.0029	0.0004	0.3033
0.0004	0.0004	0.0004	0.0008	0.0004	0.3511
0.0001	0.0003	0.0001	0.0053	0.0001	0.0804
0.0251	0.0001	0.0030	0.0001	0.0023	0.1136
0.0028	0.0028	0.0028	0.0028	0.0028	0.0028
0.0006	0.0019	0.0006	0.0050	0.0006	0.4994
0.0039	0.0018	0.0002	0.0885	0.0002	0.1693
0.0003	0.0003	0.0003	0.0724	0.0003	0.1996
0.0037	0.0013	0.0013	0.0166	0.0001	0.2739
0.0092	0.0410	0.0001	0.0220	0.0002	0.1598
0.0004	0.0004	0.0004	0.0184	0.0004	0.0652
0.0080	0.0080	0.0080	0.0080	0.0080	0.0080
0.0058	0.0011	0.0001	0.0324	0.0001	0.2719
0.0001	0.0052	0.0001	0.0020	0.0001	0.3875
0.0001	0.0064	0.0002	0.0148	0.0001	0.3021
0.0002	0.0002	0.0005	0.0012	0.0007	0.2071
0.0008	0.0008	0.0008	0.0008	0.0008	0.0041
0.0003	0.0040	0.0003	0.0025	0.0003	0.1205
0.0226	0.0045	0.0181	0.0045	0.0045	0.0950
0.0003	0.0014	0.0003	0.0003	0.0006	0.5740
0.0175	0.0175	0.0175	0.0175	0.0526	0.0175
0.0039	0.0505	0.0003	0.0265	0.0000	0.3257

**Figure 8. MATLAB results for question 3(a)**

$\text{pr\_trans}(1,1) = 9.8020\text{e-}05$  and  $\text{pr\_trans}(2,3) = 4.9505\text{e-}04$

```
>> max(pr_trans(:))

ans =

    0.7920

fx >> |
```

**Figure 9. Highest probability of pr\_trans**

With reference to figure 9, the highest probability is 0.7920 located at pr\_trans(17,21). The alphabetical transition of the highest probability is when present, i = q and future, j = u.

```
>> logn_pr = logn_pr_txt(frank_encrypted_txt, pr_trans)

logn_pr =

   -8.6855e+03

(b) fx >> |
```

**Figure 10. logn\_pr of frank\_encrypted\_txt**

```
>> logn_pr = logn_pr_txt(frank_original_txt, pr_trans)

logn_pr =

   -3.7872e+03

fx >> |
```

**Figure 11. logn\_pr of frank\_original\_txt**

```
>> frank_encrypted_double = char2double(frank_encrypted_txt);
>> frank_decrypted_double = frank_decrypt_key(frank_encrypted_double);
>> frank_decrypted_txt = double2char(frank_decrypted_double);
>> logn_pr = logn_pr_txt(frank_decrypted_txt, pr_trans);
>> logn_pr = logn_pr_txt(frank_decrypted_txt, pr_trans)

logn_pr =

   -3.7872e+03

(c) fx >> |
```

**Figure 12. Natural logarithm of p(frank\_encrypted\_txt | frank\_decrypt\_key)**

```

>> frank_encrypted_double = char2double(frank_encrypted_txt);
>> frank_decrypted_double = mystery_decrypt_key(frank_encrypted_double);
>> frank_decrypted_txt = double2char(frank_decrypted_double);
>> logn_pr = logn_pr_txt(frank_decrypted_txt, pr_trans)

logn_pr =

    -8.2353e+03
fx >> |

```

**Figure 13. Natural logarithm of  $p(\text{frank\_encrypted\_txt} \mid \text{mystery\_decrypt\_key})$**

#### **Q4. Metropolis Algorithm**

(a) (i)

```

>> [accept_new_key, prob_accept] = metropolis(frank_decrypt_key, mystery_decrypt_key, pr_trans, frank_encrypted_txt)

accept_new_key =

     0

prob_accept =

     0
fx >> |

```

**Figure 14. MATLAB results for question 4(a)(i) using mystery\_decrypt\_key as the new key**

(ii)

```

>> [accept_new_key, prob_accept] = metropolis(frank_decrypt_key, frank_decrypt_key_2, pr_trans, frank_encrypted_txt)

accept_new_key =

     0

prob_accept =

    1.5248e-36
fx >> |

```

**Figure 15. MATLAB results for question 4(a)(ii) using frank\_decrypt\_key with 12<sup>th</sup> and 13<sup>th</sup> element swapped as the new key**

(b) Run 15000: log probability = -3790.7192

##### **Final decrypted text**

i cannot describe to you my sensations on the near prospect of my undertaking  
it is impossible to communicate to you a conception of the trembling sensation  
half pleasurable and half fearful with which i am preparing to depart i am going  
to unexplored regions to the land of mist and snow but i shall kill no albatross  
therefore do not be alarmed for my safety or if i should come back to you as

worn and woeful as the ancient mariner you will smile at my allusion but i will disclose a secret i have of ten attributed my attachment to my passionate enthusiasm for the dangerous mysteries of ocean to that production of the most imaginative of modern poets there is something at work in my soul which i do not understand i am practically industrious painstaking a workman to execute with perseverance and labour but besides this there is a love for the marvellous a belief in the marvellous intertwined in all my prozects which hurries me out of the common pathways of men even to the wild sea and unvisited regions i am about to explore but to return to dearer considerations shall i meet you again after having traversed immense seas and returned by the most southern cape of africa or america i dare not expect such success yet i cannot bear to look on the reverse of the picture continue for the present to write to me by every opportunity i may receive your letters on some occasions when i need them most to support my spirits i love you very tenderly remember me with affection should you never hear from me again

```
decrypt_key =

Columns 1 through 19

    5    13    10    27    22    25    23    12     2    21    19    11     6    14    17     7    26    16    24

Columns 20 through 27

    20     9     8    15     4     3    18     1

>> frank_decrypt_key

frank_decrypt_key =

Columns 1 through 24

    5    13    26    27    22    25    23    12     2    21    19    11     6    14    17     7    10    16    24    20     9     8    15     4

Columns 25 through 27

    3    18     1
```

**Figure 16. Comparison of the generated decrypt\_key and frank\_decrypt\_key**

As seen in figure 16, frank\_decrypt\_key differs from the generated decrypt\_key at column 3 and 17 of the two matrices. In the frank\_decrypt\_key, letter c is mapped to letter z and letter q is mapped to letter j, whereas, in the generated decrypt\_key, letter c is mapped to letter j and letter q is mapped to letter z.

The keys difference result in a different final text because letter c and letter q are not mapped correctly to letter z and letter j respectively. For instance, in the final text, using the generated decrypt\_key, there is a wrong word such as 'prozects'. If frank\_decrypt\_key is used instead, 'projects' will be produced after decryption.

The algorithm does not show up exactly the correct answer because the acceptance of a new key depends on the log\_new and log\_curr values generated from log\_pr\_txt.m, which is reliant on the pr\_trans generated from the training\_txt. As such, the acceptance of a new key indirectly depends on the training\_txt being used. In this assignment, the statistical regularity of

normal written English is being exploited to obtain the pr\_trans from training\_txt. However, the training\_txt might not be the best representation of the probabilities of all alphabetical sequences. In addition, using a different set of training\_txt with different amount characters will yield a different pr\_trans values. As such, the training\_txt might not have provided sufficient information to generate the pr\_trans which affected the outcome.

(c) Run 15000: log probability = -4265.771

## Final decrypted text

well three or four months run along and it was well into the winter now i had been to school most all the time and could spell and read and write qust a little and could say the multiplication table up to six times seven is thirty five and i don t reckon i could ever get any further than that if i was to live forever i don t take no stock in mathematics anyway at first i hated the school but by and by i got so i could stand it whenever i got uncommon tired i played hookey and the hiding i got next day done me good and cheered me up so the longer i went to school the easier it got to be i was getting sort of used to the widow s ways too and they warn t so raspy on me living in a house and sleeping in a bed pulled on me pretty tight mostly but before the cold weather i used to slide out and sleep in the woods sometimes and so that was a rest to me i liked the old ways best but i was getting so i liked the new ones too a little bit the widow said i was coming along slow but sure and doing very satisfactory she said she warn t ashamed of me one morning i happened to turn over the salt cellar at breakfast i reached for some of it as juick as i could to throw over my left shoulder and keep off the bad luck but miss watson was in ahead of me and crossed me off she says take your hands away huckleberry what a mess you are always making the widow put in a good word for me but that warn t going to keep off the bad luck i knowed that well enough i started out after breakfast feeling worried and shaky and wondering where it was going to fall on me and what it was going to be there is ways to keep off some kinds of bad luck but this wasn t one of them kind so i never tried to do anything but qust poked along low spirited and on the watch out

```
decrypt_key =
Columns 1 through 21
18 24 17 16 26 27 4 6 3 14 2 15 21 8 22 20 1 19 7 12 9
Columns 22 through 27
23 13 5 10 25 11
>> mystery_decrypt_key
mystery_decrypt_key =
Columns 1 through 21
18 24 10 16 26 27 4 6 3 14 2 15 21 8 22 20 1 19 7 12 9
Columns 22 through 27
23 13 5 17 25 11
fx >> |
```

**Figure 17. Comparison of the generated decrypt\_key and mystery\_decrypt\_key**



As observed in figure 17, `mystery_decrypt_key` differs from `decrypt_key` at column 3 and 25 of the two matrices. In the `mystery_decrypt_key`, letter c is mapped to letter j and letter y is mapped to letter q, whereas, in the generated `decrypt_key`, letter c is mapped to letter q and letter y is mapped to letter j.

The keys difference result in a different final text because letter c and letter y are not map correctly to letter q and letter j respectively. For instance, in the final text, using the generated `decrypt_key`, there are wrong words such as 'qust' and 'juick' after decryption. If `mystery_decrypt_key` is used instead, 'just' and 'quick' will be produced after decryption.

Similar to the question 4(b), the algorithm does not show up exactly the correct answer because the acceptance of a new key depends on the `log_new` and `log_curr` values generated from `log_pr_txt.m`, which is reliant on the `pr_trans` generated from the `training_txt`. As such, the acceptance of a new key indirectly depends on the `training_txt` being used. In this assignment, the statistical regularity of normal written English is being exploited to obtain the `pr_trans` from `training_txt`. However, the `training_txt` might not be the best representation of the probabilities of all alphabetical sequences. In addition, using a different set of `training_txt` with different amount characters will yield a different `pr_trans` values. As such, the `training_txt` might not have provided sufficient information to generate the `pr_trans` which affected the outcome.

## **Q5. Extra Credits**

Suggestions:

1. In this assignment, only 1 sample is being generated. More samples could have been generated to get the correct `decrypt_key`. In addition, the `rng(5, 'twister')` in `mcmc_decrypt_text.m` has to be removed to generate multiple varying samples. When more samples are generated, we could find the MAP estimate of multiple samples to generate the correct `decrypt_key`.
2. The number of characters used in the `training_txt` could be changed or increased to provide a more accurate representation of the `pr_trans`.
3. The `rand(1)` model used to accept a new `decrypt_key` could have been improved. `rand(1)` consists of numbers within the range of 0 to 1 only. As such, there is chance of accepting a key that is considered not the correct `decrypt_key` when  $\text{log\_new} < \text{log\_curr}$ .