“Mini” Project: Cryptography

By Liam Purcell

For this project I chose to do something that I did briefly in high school and found extremely interesting, cryptography. Because a lot of cryptography is simple matrix operations; addition, multiplication, and inversion, I chose a difficult and long text to encrypt. To assist my encryption efforts, I learned how to write some simple code and ran it to help accelerate the calculations. (All calculations and intermediary steps, forward and backward, will be attached at the end of the paper)

Original Text

全くその通りだ。全く持って無意味だ。どんなに夢や希望を持っていても、胸腹な人生をおくつことができたとしても、岩で体を打ち砕かれても、同じだ。人はいずれ死ぬ。

ならば人生には意味がないのか？ そもそも生まれてきたことに意味わなかったのか？

死んだ仲間もそうなのか？ あの兵士たちも無意味だったのか？

いや、違う！

あの兵士たちいい身を充てるのは我々だ。あの勇敢な死者！

思うことができるのは…聖者である我々だ。

我々はここで死に！

次の聖者に意味を託す！それこそ唯一！

この残酷な世界にあらがうすべなのだ！

兵士よ行かれ！

兵士よ叫べ！

兵士よ戦え！

<u>Romanized Text</u>

This is the pronunciation of the Japanese text above, simply put into their closest phonetic spelling using English characters -

Mattaku sono toori da. Mattaku motte mu imi da. Donna ni yume ya kibou wo motte ite mo, koufuku na jinsei wo okuru koto ga dekita toshite mo, iwade karada wo uchikuda karete mo, onaji da. Hito wa izure shinu.

Naraba jinsei ni wa imi ga nai no ka? Somosomo umarete kita koto ni imi wa nakatta no ka?

Shinda nakama mo sou na no ka? Ano heishi-tachi mo mu imi datta no ka?

IYA, CHIGAU!

Ano heishi-tachi ni imi wo ataeru no wa wareware da! Ano yuukan na shisha! Aware na shisha!

Omou koto ga dekiru no wa... seija de aru wareware da!

Wareware wa koko de shini! Tsugi no seija ni imi wo takusu! Sore koso yui itsu!

Kono zankoku na sekai ni aragau sube na no da!

HEISHI YO IKARE! HEISHI YO SAKEBE! HEISHI YO TATAKAE

This monologue is from episode 53, "Perfect Game", from my favorite show, Attack on Titan. The commander, Erwin, and his troops are surrounded and in order to distract their attackers, Erwin essentially sentences them all to death and charges at them and the incoming waves of boulders being thrown at them. It is hard to visualize unless you have seen the show and know what is happening, but this is the speech he gives them to rally them behind him and do what must be done so others can complete their goal. Here is the English version that is not being encrypted, but provides context for what the final translation should resemble. It is important to note that the English version of the show adds a few lines that the Japanese script does not have, and that not all words translate perfectly, for example, above "聖者" (seija) translates to "saint", but more closely resembles "revered" in the context of the monologue.

> You‘re precisely right. It‘s all meaningless. No matter what dreams or hopes you had… No matter how blessed a life you‘ve lived… It‘s all the same if you‘re shattered by rocks. Everyone will day someday. Does that mean life is meaningless? Was there even any meaning in our being born? Would you say that to our fallen comrades? Their lifes… Were they meaningless? No they weren‘t! It‘s us who gives meaning to our comrade‘s lives! The brave fallen! The anguished fallen! The ones who will remember them… are us, the living! We die trusting the living who follow to find meaning in our lives! That is the

sole method in which we can rebel against this cruel world! My soldiers, rage! My soldiers, scream! My soldiers, fight!

This is the Japanese Romanized text translated to numbers, this translation simply follows a direct substitution, A=0, B=1, … , Z=25 excluding punctuation and spaces. The decision to exclude spaces was intentional as in Japanese there are not spaces as can be seen in the above text.

[12,0,19] [19,0,10] [20,18,14] [13,14,19] [14,14,17] [8,3,0] [12,0,19] [19,0,10]
[20,12,14] [19,19,4] [12,20,8] [12,8,3] [0,3,14] [13,13,0] [13,8,24] [20,12,4]
[24,0,10] [8,1,14] [20,22,14] [12,14,19] [19,4,8] [19,4,12] [14,10,14] [20,5,20]
[10,20,13] [0,9,8] [13,18,4] [8,22,14] [14,10,20] [17,20,10] [14,19,14] [6,0,3]
[4,10,8] [19,0,19] [14,18,7] [8,19,4] [12,14,8] [22,0,3] [4,10,0] [17,0,3] [0,22,14]
[20,2,7] [8,10,20] [3,0,10] [0,17,4] [19,4] [12,14,14] [13,0,9] [8,3,0] [7,8,19]
[14,22,0] [8,25,20] [17,4,18] [7,8,13] [20,13,0] [17,0,1] [0,9,8] [13,18,4] [8,13,8]
[22,0,8] [12,8,6] [0,13,0] [8,13,14] [10,0,18] [14,12,14] [18,14] [12,14,20] [12,0,17]
[4,19,4] [10,8,19] [0,10,14] [19,14,13] [8,8,12] [8,22,0] [13,0,10] [0,19,19] [0,13,14]
[10,0,18] [7,8,13] [3,0,13] [0,10,0] [12,0,12] [14,18,14] [20,13,0] [13,14,10] [0,0,13]
[14,7,4] [8,18,7] [8,19,0] [2,7,8] [12,14,12] [20,8,12] [8,3,0] [19,19,0] [13,14,10]
[0,8,24] [0,2,7] [8,6,0] [20,0,13] [14,7,4] [8,18,7] [8,19,0] [2,7,8] [13,8,8] [12,8,22]
[14,0,19] [0,4,17] [20,13,14] [22,0,22] [0,17,4] [22,0,17] [4,3,0] [0,13,14] [24,20,20]
[10,0,13] [13,0,18] [7,8,18] [7,0,0] [22,0,17] [4,13,0] [18,7,8] [18,7,0] [14,12,14]
[20,10,14] [19,14,6] [0,3,4] [10,8,17] [20,13,14] [22,0,18] [4,8,9] [0,3,4] [0,17,20]
[22,0,17] [4,22,0] [17,4,3] [0,22,0] [17,4,22] [0,17,4] [22,0,10] [14,10,14] [3,4,18]
[7,8,13] [8,19,18] [20,6,8] [13,14,18] [4,8,9] [0,13,8] [8,12,8] [22,14,19] [0,10,20]

[18,20,18] [14,17,4] [10,14,18] [14,24,20] [8,8,19] [18,20,10] [14,13,14] [25,0,13]
[10,14,10] [20,13,0] [18,4,10] [0,8,13] [8,0,17] [0,6,0] [20,18,20] [1,4,13] [0,13,14]
[3,0,7] [4,8,18] [7,8,24] [14,8,10] [0,17,4] [7,4,8] [18,7,8] [24,14,18] [0,10,4] [1,4,7]
[4,8,18] [7,8,24] [14,19] [0,19,0] [10,0,4]

While this is a lot to look at, it is incredibly simple to decode, you would just reverse the substitution previously applied. To further encrypt this message, you would operations to these sets of numbers. For my encryption, I chose to use the most complex method provided by our textbook, a toy block cipher. To encrypt this cipher, you go through the same 3 step process multiple times. You first add the first n numbers of the key to the n-tuplet, in this case 3 and triplets. Then, you diffuse by the encryption matrix A, in this case a 3x3. Then you do another substitution, this typically flips the alphabet so A becomes Z, B becomes Y, etc.. It is important to note that during the diffusion step you can get numbers greater than the highest value of your original substitution, for example 33 does not correspond to anything. To account for that you are constantly using modular division to make sure that everything will translate in the end, so that 33 becomes a 7 in this case because for my cipher there are 26 possible characters.

The key for my cipher is 200101307, because it is 9 numbers long and I use triplets, the encryption loop described above will repeat 3 times. This key was chosen somewhat at random, it is my birthday 2001/01/30 and then just the number 7 to make it 9 digits long.

The first step of the encryption process is adding elements of the key to the triplets. So, the first triplet becomes [14,0,19], the second [21,0,10], and so on. As this can get repetitive, I wrote some code to help speed this up. The code is as follows:

```
import numpy as np
```

```
outputfile = open("output.txt", "a")

inputfile = open("input.txt", "r+")

A = [2,0,0]

for line in inputfile:

    newline = line.strip("\n")

    templist = newline.split( "," )

    B = [int(x) for x in templist]

    r = (np.add(A,B))

    outputfile.write(str(r) + "\n")

inputfile.close()

outputfile.close()
```

What this code does is fairly simple, the first line just allows you to not have to type “numpy” every time you use a NumPy operation. The next two lines simply open two files, the input being the file that the numbers to be encrypted are pulled from, and the export being where the results will be pasted into. The next line “A = [x,y,z]” is the portion where the key is established x, y, and z are placeholders for the key as it is changing with every iteration of the encryption process. Next is the for statement, this establishes a loop in the indented portion of the code and makes the code loop as long as there is a line in the code that follows the conditions set. Up to this point in the code, something very similar will be present in the following steps in some form or another. The newline statement ends the line at the end of the line of text in the input file and the templist statement denotes that everything being read in the file is separate as long as there is a comma in between them. Those two lines set the parameters for what goes into B, B is the text that is actually being encrypted. The line “B = [int(x) for x in templist]” simply turns the characters read from the input file into integers as they are not read as such despite being numbers. Finally, “r” adds A and B, it is written to the output file, and the files are closed.

The next piece of code written is the modular division step, this takes place in MOD 26 as I excluded commas, spaces, and periods from my key. This gets ran after every transformation even if it does not change anything, it is just in order to be safe. The code is as follows:

```
import numpy as np

outputfile = open("output.txt", "a")

inputfile = open("input.txt", "r")

for line in inputfile:

   newline = line.strip("\n")

   templist = newline.split( "," )

   B = [int(x) for x in templist]

   r = (np.mod(B,26))

   outputfile.write(str(r) + "\n")

inputfile.close()

outputfile.close()
```

This is incredibly similar to the code that does the addition, except there is no A, as nothing is being added or multiplied to the input, and "np.add" as become "np.mod" as modular division is happening instead of addition. Within the "np.mod" statement the "B" states that B is what is being divided, and the "26" is what it is being divided by. This is by far the simplest string of code, however it is arguably the most important.

Now, the second step in encryption is diffusion. In this step the n-tuplets are multiplied by an n x n matrix, in my case triplets and a 3 x 3 matrix. The matrix used is

$$\begin{matrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{matrix}.$$

The code used to multiply this matrix to all of the triplets is also fairly simple, however more complex than the other two examples presented before. The code is as follows:

```
import numpy as np

outputfile = open("output.txt", "a")

inputfile = open("input.txt", "r")

A = [[1,2,3],[0,1,4],[5,6,0]]

for line in inputfile:

    newline = line.strip("\n")

    templist = newline.split( "," )

    B = [int(x) for x in templist]

    C = np.transpose(B)
```

```
        r = np.transpose(np.dot(A,C))
        outputfile.write(str(r) + "\n")
    inputfile.close()
    outputfile.close()
```

What this code is a little bit different than the other examples as the orientation of the items being multiplied is more important. So, the first 3 lines are the same as in the other examples, the line beginning with "A" is the only difference up to that point, what that line establishes is the matrix that will be used to diffuse the triplets. From there the next four lines are the same, but then C is established. What C does is change B from a 1 x 3 matrix to a 3 x 1 matrix. This is because you cannot multiply a 3 x 3 matrix by a 1 x 3 matrix. The next line does two operations in one, first it takes the dot product of matrices A and C, the matrix multiplication itself, and then for the sake of formatting it transposes it back to a 1 x 3 matrix. The last three lines writing the results and closing the opened files are the same.

The final step of the encryption is the application of the S-Box, this is the aforementioned second substitution that flips the alphabet, A becomes Z, B becomes Y etc.. With my basic knowledge of coding, I could not find a way to make this step automate itself, however I did have an idea which could work. My problem with this step was when you replace a 0 with a 25, A with Z, you still have to work through the other values, so eventually you will end up with the original 0 that became a 25 being flipped back to a 0. An idea that I had but could not successfully implement at the time of finishing this project was rather than changing a 0 to a 25, I could change a 0 to 51, a number that would not be flipped back to 0, but could then be modularly divided back to 25. However, for the work done in this project, this step will be omitted.

Calculations

Initial Addition Step

[14 0 19] [21 0 10] [2 18 14] [16 14 17] [10 3 0] [14 0 19]21 0 10] [21 19 4]14 20 8] [2 3 14] [15 13 0] [15 8 24] [22 12 4] [26 0 10] [10 1 14] [22 22 14] [14 14 19] [21 4 8] [21 4 12] [16 10 14] [22 5 20] [12 20 13] [2 9 8] [15 18 4] [10 22 14] [16 10 20] [19 20 10] [16 19 14] [8 0 3] [6 10 8] [21 0 19] [16 18 7] [10 19 4] [14 14 8] [24 0 3] [6 10 0] [19 0 3] [2 22 14] [22 2 7] [10 10 20] [5 0 10] [2 17 4] [21 4 12] [16 14 13] [2 9 8] [5 0 7] [10 19 14] [24 0 8] [27 20 17] [6 18 7] [10 13 20] [15 0 17] [2 1 0] [11 8 13] [20 4 8] [15 8 22] [2 8 12] [10 6 0] [15 0 8] [15 14 10] [2 18 14] [14 14 18] [16 12 14] [22 12 0] [19 4 19] [6 10 8] [21 0 10] [16 19 14] [15 8 8] [14 8 22] [2 13 0] [12 0 19] [21 0 13] [16 10 0] [20 7 8] [15 3 0] [15 0 10] [2 12 0] [14 14 18] [16 20 13] [2 13 14] [12 0 0] [15 14 7] [6 8 18] [9 8 19] [2 2 7] [10 12 14] [14 20 8] [14 8 3 [2 19 19] [2 13 14] [12 0 8] [26 0 2] [9 8 6] [2 20 0] [15 14 7] [6 8 18] [9 8 19] [2 2 7] [10 13 8] [10 12 8] [24 14 0] [21 0 4] [19 20 13] [16 22 0] [24 0 17] [6 22 0] [19 4 3] [2 0 13] [16 24 20] [22 10 0] [15 13 0] [20 7 8] [20 7 0] [2 22 0] [19 4 13] [2 18 7] [10 18 7] [2 14 12] [16 20 10] [16 19 14] [8 0 3] [6 10 8] [19 20 13] [16 22 0] [20 4 8] [11 0 3] [6 0 17] [22 22 0] [19 4 22] [2 17 4] [5 0 22] [2 17 4] [24 0 17] [6 22 0] [12 14 10] [16 3 4] [20 7 8] [15 8 19] [20 20 6] [10 13 14] [20 4 8] [11 0 13] [10 8 12] [10 22 14] [21 0 10] [22 18 20] [20 14 17] [6 10 14] [20 14 24] [22 8 8] [21 18 20] [12 14 13] [16 25 0] [15 10 14] [12 20 13] [2 18 4] [12 0 8] [15 8 0] [19 0 6] [2

20 18] [22 1 4] [15 0 13] [16 3 0] [9 4 8] [20 7 8] [26 14 8] [12 0 17] [6 7 4] [10 18 7] [10 24 14] [20 0 10] [6 1 4] [9 4 8] [20 7 8] [26 14 19] [2 19 0]

[12 0 4]

First Diffusion

[19 24 18] [25 14 1] [2 22 14] [22 12 3] [17 4 8] [16 3 16] [19 24 18] [25 14 1] [10 16 0] [19 9 11] [0 0 8] [13 20 14] [24 7 2] [15 13 23] [25 0 19] [6 2 0] [4 14 0] [2 5 4] [4 0 8] [21 12 24] [1 10 25] [13 0 25] [0 14 10] [14 7 10] [13 20 24] [18 15 12] [11 8 1] [18 0 0] [18 12 10] [11 8 7] [18 23 12] [17 12 14] [24 16 12] [0 24 1] [21 20 6] [8 9 8] [14 20 24] [7 12 16] [0 10 12] [2 12 17] [10 0 12] [21 4 18] [12 12 6] [9 14 25] [22 7 8] [13 0 25] [5 14 8] [18 15 12] [0 2 25] [12 23 8] [22 6 16] [14 10 21] [11 20 8] [18 15 24] [14 16 23] [4 1 16] [14 8 25] [0 10 20] [19 18 19] [2 4 6] [22 6 8] [13 6 23] [21 2 3] [2 22 14] [18 8 24] [4 16 22] [20 12 0] [6 2 15] [24 16 12] [25 14 1] [18 23 12] [3 14 19] [18 18 14] [2 13 10] [17 24 8] [8 0 1] [10 10 10] [6 13 12] [21 3 15] [19 14 23] [0 12 4] [18 8 24] [17 20 18] [18 17 10] [12 0 8] [12 16 3] [24 2 0] [4 6 15] [1 4 22] [24 16 18] [0 0 8] [13 20 14] [19 17 20] [18 17 10] [10 6 8] [6 8 0] [17 6 15] [16 20 0] [12 16 3] [24 2 0] [4 6 15] [1 4 22] [8 19 24] [6 18 18] [0 14 22] [7 16 1] [20 20 7] [8 22 4] [23 16 16] [24 22 6] [10 16 15] [15 0 10] [20 0 16] [16 10 14] [15 13 23] [6 13 12] [8 7 12] [20 22 12] [14 4 15] [7 20 14] [15 20 2] [14 10 16] [8 8 18] [18 23 12] [17 12 14] [24 16 12] [20 20 7] [8 22 4] [0 10 20] [20 12 3] [5 16 4] [14 22 8] [15 14 15] [22 7 8] [19 10 25] [22 7 8] [23 16 16] [24 22 6] [18 2 14] [8 19 20] [6 13 12] [10 6 19] [0 18 12] [0 17 24] [0 10 20] [24 0 3] [10 4 20] [18 0 0] [25 14 1] [14 20 10] [21 4 2] [16 14 12] [16 6 2] [10 14 2] [13 20 5] [1 14 14] [14 25 22] [25 14 5] [13 20 24] [24 8 14] [10 6 8] [5 8 19] [11 24 17] [18 14 0]

[10 17 12] [2 0 23] [22 3 20] [15 10 17] [6 13 12] [0 20 6] [11 16 8] [6 23 20] [15 20 2] [22 2 12] [24 14 22] [20 17 10] [15 10 17] [6 13 12] [7 12 6] [14 19 20] [24 16 8]

Second Addition Step

[20 24 19] [0 14 2] [3 22 15] [23 12 4] [18 4 9] [17 3 17] [20 24 19] [0 14 2] [11 16 1] [20 9 12] [1 0 9] [14 20 15] [25 7 3] [16 13 24] [0 0 20] [7 2 1] [5 14 1] [3 5 5] [5 0 9] [22 12 25] [2 10 0] [14 0 0] [1 14 11] [15 7 11] [14 20 25] [19 15 13] [12 8 2] [19 0 1] [19 12 11] [12 8 8] [19 23 13] [18 12 15] [25 16 13] [1 24 2] [22 20 7] [9 9 9] [15 20 25] [8 12 17] [1 10 13] [3 12 18] [11 0 13] [22 4 19] [13 12 7] [10 14 0] [23 7 9] [14 0 0] [6 14 9] [19 15 13] [1 2 0] [13 23 9] [23 6 17] [15 10 22] [12 20 9] [19 15 25] [15 16 24] [5 1 17] [15 8 0] [1 10 21] [20 18 20] [3 4 7] [23 6 9] [14 6 24] [22 2 4] [3 22 15] [19 8 25] [5 16 23] [21 12 1] [7 2 16] [25 16 13] [0 14 2] [19 23 13] [4 14 20] [19 18 15] [3 13 11] [18 24 9] [9 0 2] [11 10 11] [7 13 13] [22 3 16] [20 14 24] [1 12 5] [19 8 25] [18 20 19] [19 17 11] [13 0 9] [13 16 4] [25 2 1] [5 6 16] [2 4 23] [25 16 19] [1 0 9] [14 20 15] [20 17 21] [19 17 11] [11 6 9] [7 8 1] [18 6 16] [17 20 1] [13 16 4] [25 2 1] [5 6 16] [2 4 23] [9 19 25] [7 18 19] [1 14 23] [8 16 2] [21 20 8] [9 22 5] [24 16 17] [25 22 7] [11 16 16] [16 0 11] [21 0 17] [17 10 15] [16 13 24] [7 13 13] [9 7 13] [21 22 13] [15 4 16] [8 20 15] [16 20 3] [15 10 17] [9 8 19] [19 23 13] [18 12 15] [25 16 13] [21 20 8] [9 22 5] [1 10 21] [21 12 4] [6 16 5] [15 22 9] [16 14 16] [23 7 9] [20 10 0] [23 7 9] [24 16 17] [25 22 7] [19 2 15] [9 19 21] [7 13 13] [11 6 20] [1 18 13] [1 17 25] [1 10 21] [25 0 4] [11 4 21] [19 0 1] [0 14 2] [15 20 11] [22 4 3] [17 14 13] [17 6 3] [11 14 3] [14 20 6] [2 14 15] [15 25 23] [0 14 6] [14 20 25] [25 8 15] [11 6 9] [6 8 20] [12 24 18] [19 14 1] [11 17 13] [3 0 24]

[23 3 21] [16 10 18] [7 13 13] [1 20 7] [12 16 9] [7 23 21] [16 20 3] [23 2 13] [25 14 23] [21 17 11] [16 10 18] [7 13 13] [8 12 7] [15 19 21] [25 16 9]

Second Diffusion

[21 22 10] [8 22 6] [14 4 17] [7 2 5] [1 14 10] [22 19 25] [21 22 10] [8 22 6] [20 20 21] [22 5 24] [2 10 5] [21 2 8] [22 19 11] [10 5 2] [8 2 0] [14 6 21] [10 18 5] [2 25 19] [6 10 25] [17 8 0] [22 10 18] [14 0 18] [10 6 11] [10 25 13] [25 16 8] [10 15 3] [8 16 4] [22 4 17] [24 4 11] [0 14 4] [0 23 25] [9 20 6] [18 16 13] [3 6 19] [5 22 22] [2 19 21] [0 16 13] [5 2 8] [8 10 13] [3 6 9] [24 0 3] [9 2 4] [6 14 7] [12 14 4] [12 17 1] [14 0 18] [9 24 10] [10 15 3] [5 2 17] [8 7 21] [8 22 21] [23 20 5] [1 4 24] [20 11 3] [15 8 15] [6 17 5] [5 8 19] [6 16 13] [12 20 0] [6 6 13] [10 16 21] [20 24 2] [12 18 18] [14 4 17] [6 4 13] [2 4 17] [22 16 21] [7 14 21] [18 16 13] [8 22 6] [0 23 25] [14 16 0] [22 0 21] [10 5 15] [15 8 0] [15 8 19] [12 2 11] [20 13 9] [24 15 24] [16 6 2] [14 6 25] [6 4 13] [11 18 2] [8 9 15] [14 10 13] [5 6 5] [6 6 7] [13 18 9] [1 18 8] [10 14 13] [2 10 5] [21 2 8] [13 23 20] [8 9 15] [24 16 13] [0 12 5] [0 18 22] [8 24 23] [5 6 5] [6 6 7] [13 18 9] [1 18 8] [18 15 3] [22 16 13] [20 2 11] [20 24 6] [7 0 17] [16 16 21] [3 6 8] [12 24 23] [13 2 21] [23 18 2] [20 16 1] [4 18 15] [10 5 2] [20 13 9] [10 7 9] [0 22 3] [19 16 21] [15 2 4] [13 6 18] [8 0 5] [4 6 15] [0 23 25] [9 20 6] [18 16 13] [7 0 17] [16 16 21] [6 16 13] [5 2 21] [1 10 22] [8 6 25] [14 0 8] [12 17 1] [14 10 4] [12 17 1] [3 6 8] [12 24 23] [16 10 3] [6 25 3] [20 13 9] [5 8 13] [24 18 9] [6 13 3] [6 16 13] [11 16 21] [4 10 1] [22 4 17] [8 22 6] [10 12 13] [13 16 4] [6 14 13] [12 18 17] [22 0 9] [20 18 8] [23 22 16] [4 13 17] [20 12 6] [25 16 8] [8 16 17] [24 16 13] [4 10 0] [10 18 22] [24 18 23] [6 17 1] [23 18 15] [14 9 3] [12 4 10] [20 13 9] [10 22 21] [19 0 0] [12 3 17] [13 6 18] [14 2 23] [18 2 1] [10 9 25] [12 4 10]

[20 13 9] [1 14 8] [12 25 7] [6 0 13]

Third and Final Addition Step

[24 22 17] [11 22 13] [17 4 24] [10 2 12] [4 14 17] [25 19 6] [24 22 17] [11 22 13] [23 20 2] [25 5 5] [5 10 12] [24 2 15] [25 19 18] [13 5 9] [11 2 7] [17 6 2] [13 18 12] [5 25 0] [9 10 6] [20 8 7] [25 10 25] [17 0 25] [13 6 18] [13 25 20] [2 16 15] [13 15 10] [11 16 11] [25 4 24] [1 4 18] [3 14 11] [3 23 6] [12 20 13] [21 16 20] [6 6 0] [8 22 3] [5 19 2] [3 16 20] [8 2 15] [11 10 20] [6 6 16] [1 0 10] [12 2 11] [9 14 14] [15 14 11] [15 17 8] [17 0 25] [12 24 17] [13 15 10] [8 2 24] [11 7 2] [11 22 2] [0 20 12] [4 4 5] [23 11 10] [18 8 22] [9 17 12] [8 8 0] [9 16 20] [15 20 7] [9 6 20] [13 16 2] [23 24 9] [15 18 25] [17 4 24] [9 4 20] [5 4 24] [25 16 2] [10 14 2] [21 16 20] [11 22 13] [3 23 6] [17 16 7] [25 0 2] [13 5 22] [18 8 7] [18 8 0] [15 2 18] [23 13 16] [1 15 5] [19 6 9] [17 6 6] [9 4 20] [14 18 9] [11 9 22] [17 10 20] [8 6 12] [9 6 14] [16 18 16] [4 18 15] [13 14 20] [5 10 12] [24 2 15] [16 23 1] [11 9 22] [1 16 20] [3 12 12] [3 18 3] [11 24 4] [8 6 12] [9 6 14] [16 18 16] [4 18 15] [21 15 10] [25 16 20] [23 2 18] [23 24 13] [10 0 24] [19 16 2] [6 6 15] [15 24 4] [16 2 2] [0 18 9] [23 16 8] [7 18 22] [13 5 9] [23 13 16] [13 7 16] [3 22 10] [22 16 2] [18 2 11] [16 6 25] [11 0 12] [7 6 22] [3 23 6] [12 20 13] [21 16 20] [10 0 24] [19 16 2] [9 16 20] [8 2 2] [4 10 3] [11 6 6] [17 0 15] [15 17 8] [17 10 11] [15 17 8] [6 6 15] [15 24 4] [19 10 10] [9 25 10] [23 13 16] [8 8 20] [1 18 16] [9 13 10] [9 16 20] [14 16 2] [7 10 8] [25 4 24] [11 22 13] [13 12 20] [16 16 11] [9 14 20] [15 18 24] [25 0 16] [23 18 15] [0 22 23] [7 13 24] [23 12 13] [2 16 15] [11 16 24] [1 16 20] [7 10 7] [13 18 3] [1 18 4] [9 17 8] [0 18 22] [17 9 10] [15 4 17] [23 13 16] [13 22 2] [22 0 7] [15 3 24] [16 6 25] [17 2 4] [21 2 8] [13 9 6] [15 4 17] [23 13 16] [4 14 15] [15 25 14] [9 0 20]

Final Diffusion

[15 12 18] [16 22 5] [19 22 5] [24 24 10] [5 4 0] [3 17 5] [15 12 18] [16 22 5] [17 2 1] [24 25 25] [9 6 7] [21 10 2] [13 13 5] [24 15 17] [10 4 15] [9 14 17] [7 14 17] [3 25 19] [21 8 1] [5 10 18] [16 6 3] [14 22 7] [1 0 23] [19 1 7] [1 24 2] [21 3 25] [24 8 21] [1 22 19] [11 24 3] [12 6 21] [15 21 23] [13 20 24] [9 18 19] [18 6 14] [9 8 16] [23 1 9] [17 18 7] [5 10 0] [13 12 11] [14 18 14] [5 14 5] [23 20 20] [1 18 25] [24 6 3] [21 23 21] [14 22 7] [7 14 22] [21 3 25] [6 20 0] [5 15 19] [9 4 5] [24 16 16] [1 24 18] [23 25 25] [22 18 8] [1 13 17] [24 8 10] [23 18 11] [24 22 13] [3 8 3] [25 24 5] [20 8 25] [22 14 1] [19 22 5] [25 6 17] [7 22 23] [11 24 13] [18 22 4] [9 18 19] [16 22 5] [15 21 23] [18 18 25] [5 8 21] [11 15 17] [3 10 8] [8 8 8] [21 22 9] [19 25 11] [20 9 17] [6 16 1] [21 4 17] [25 6 17] [25 2 22] [17 19 5] [19 12 15] [4 2 24] [11 10 3] [22 4 6] [7 0 24] [23 16 19] [9 6 7] [21 10 2] [13 1 10] [17 19 5] [15 18 23] [11 8 9] [22 4 19] [19 14 17] [4 2 24] [11 10 3] [22 4 6] [7 0 24] [3 3 13] [13 18 13] [3 22 23] [6 24 25] [4 18 24] [5 24 9] [11 14 14] [23 14 11] [0 10 14] [11 2 4] [1 22 3] [5 2 13] [24 15 17] [19 25 11] [23 19 3] [25 10 17] [8 24 24] [3 20 24] [25 2 12] [21 22 3] [7 16 19] [15 21 23] [13 20 24] [9 18 19] [4 18 24] [5 24 9] [23 18 11] [18 10 0] [7 22 2] [15 4 13] [10 8 7] [21 23 21] [18 2 15] [21 23 21] [11 14 14] [23 14 11] [17 24 25] [11 13 13] [19 25 11] [6 10 10] [7 4 9] [13 1 19] [23 18 11] [0 24 10] [25 16 17] [1 22 19] [16 22 5] [19 14 7] [3 8 20] [19 16 25] [19 10 1] [21 12 21] [0 0 15] [9 10 2] [1 5 9] [8 12 5] [1 24 2] [11 8 21] [15 18 23] [22 12 17] [6 4 17] [23 8 9] [15 23 17] [24 2 4] [13 23 9] [22 20 21] [19 25 11] [11 4 15] [17 2 6] [15 21 15] [25 2 12] [7 18 19] [23 8 13] [23 7 15] [22 20 21] [19 25 11] [25 22 0] [3 3 17] [17 2 19]

Translated from Numbers to Letters

Pmsqwftwfyykfeadrfpmsqwfrcbyzzjghvkcnnfyprkepjorhordztvibfksqgdowhbaxtbhb

ycvdzyivbwtlydmgvpvxnuyjstsgojiqxbjrshfkanmlosofofxuubszygdvxvowhhowvdzg

uafptjefyqqbysxzzwsibnryikxslywndidzyfuizwobtwfzgrhwxlynswejstqwfpvxsszfivlp

rdkiiiivwjtzlujrgqbverzgrzcwrtftmpecylkdweghayxqtjghvkcnbkrtfpsxlijwettorecylkd

weghayddnnsndwxgyzesyfyjlooxolakolcebwdfcnyprtzlxtdzkriyyduyzcmvwdhqtpvxn

uyjstesyfyjxslskahwcpenkihvxvscpvxvlooxolryzlnntzlgkkhejnbtxslaykzqrbwtqwftoh

diutqztkbvmvaapjkcbfjimfbyclivpsxwmrgerxijpxrycenxjwuvtzlleprcgpvpzcmhstxinx

hpwuvtzlzwaddrrct

This is completely incoherent and the exact goal of this encryption, however by knowing the key, 200101307, and the encryption matrix, $\begin{matrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{matrix}$, you can work backwards to decrypt this message. To do this you first invert the encryption matrix to get A^{-1}, $\begin{matrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{matrix}$, and multiply all the diffused triplets by that, then you modularly divide, and subtract the last three entries in the key, this process repeats until you end up with the original "encoded" triplets. From there you can translate them into the Japanese Romanized text and back into Japanese. To do this in the code, all you would do is change the matrix to A^{-1}, and make A negative in the addition code, then run them in reverse order. The modular division code does not need to be touched.

References

Matrix Theory and Linear Algebra, By Peter Selinger

NumPy fundamentals — NumPy v1.23 Manual. (n.d.).
https://numpy.org/doc/stable/user/basics.html

numpy.add — NumPy v1.23 Manual. (n.d.).
https://numpy.org/doc/stable/reference/generated/numpy.add.html

numpy.dot — NumPy v1.23 Manual. (n.d.).
https://numpy.org/doc/stable/reference/generated/numpy.dot.html

numpy.mod — NumPy v1.23 Manual. (n.d.).
https://numpy.org/doc/stable/reference/generated/numpy.mod.html

Special thanks to my mom who actually knows how to code, for checking everything I wrote when it broke repeatedly for 2-3 weeks straight.