Recently, I attended the Florida Cyber Conference 2019 held at the convention center downtown Tampa. The first day, they held a freeplay CTF that had an interesting little steganography / rotation cipher challenge that was two parts. The first part was simply one of the those ‘this .pdf is a xxx’ which, once solved took you to the real part of the challenge which I detail below.  
  
The setup to the challenge was to solve a Sudoku puzzle and use the top three rows as a substitution table for a substitution cipher.  
  
Essentially, the 26 English capital characters plus a “space” make an ideal 27 character set that can be placed in the first three rows of a Sudoku table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** |
| **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** |
| **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |  |

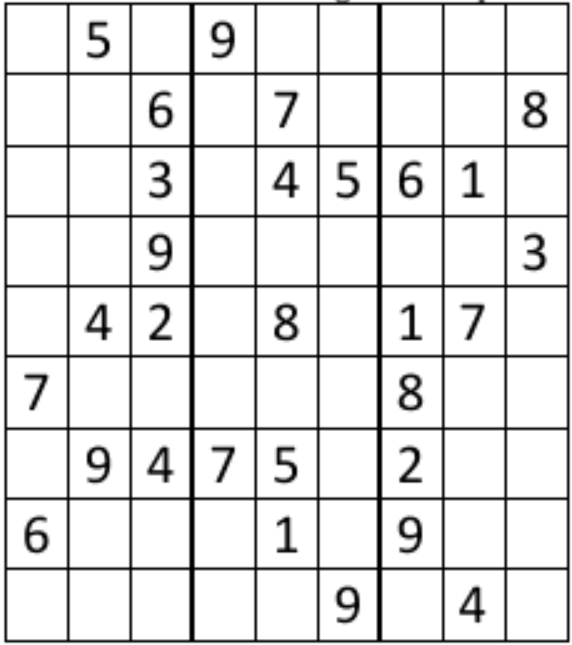
Then, the first three rows of a solved Sudoku can be used to define the substitution table for a substitution cipher. For example, if the first three rows of the Sudoku solution were;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **4** | **5** | **8** | **9** | **3** | **2** | **7** | **1** | **6** |
| **9** | **6** | **7** | **1** | **5** | **8** | **3** | **4** | **2** |
| **1** | **2** | **3** | **4** | **7** | **6** | **8** | **9** | **5** |

Then the substation table for the 27-character set would be;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **J** | **K** | **T** | **X** | **F** | **E** | **Y** | **G** | **R** |
| **U** | **L** | **S** | **D** | **N** | **W** | **I** | **P** | **H** |
| **A** | **B** | **C** | **M** | **V** | **O** | **Z** |  | **Q** |

That is, A rotated forward 4 spaces (within that block of 3x3, starting from the first space [upper leftmost square]) would become J, B would become K, and so on. Treating each 3x3 as self-contained from the others/  
In the example above  
A = J, B = K, C = T, D = X, E = F, F = E, G = Y, H = G, I = R, J = U, K = L, L = S,M = D, N = N,   
O = W, P = I, Q = P, R = H, S = A, T = B, U = C, V = M, W = V, X = O, Y = Z, Z = \_, \_ = Q

The unsolved Sudoku puzzle was this;  


The challenge is this.  
Suppose someone solved the above Sudoku and used it in the manner described above to get the following encrypted message;  
TDYUDLEOYAYUCXDBCYQCGGLOYRDCYVOOXYTDYOVTOZYTHOYVCFKOZUYDVOYTDYVPVOYONASTLRYDVSOYPVYOPVYOASHYZDMOASHYSDLCCFVYAVXYOASHYTHZOOYTPFOUYTHZOOYKDN

Decipher the message.  
  
I am interested in writing a script in Python to automate the deciphering on the message, given the solved Sudoku. The deciphered message is inconsequential and trifling to solve with other methods.