CIS 444 – Cyber Defense and Operations
Project Specifications
Spring 2021

Scenario[1]

Year 1985. You work at NSA as a computer scientist, at a unit designed to intercept and decrypt communications between two given targets. You have access to a supercomputer called CRAY-2 (which roughly has a similar computing power of a 2011 Apple iPad).

Your team intercepts communication from an enemy state's defense headquarters to different field units, and communication from a field unit to another. CIA provides you with the following technology information:

	Encryption Capability	Decryption Capability
Defense Headquarters	CS – DES – 3DES (Does not use CS)	CS – DES – 3DES (Does not use CS)
Field Unit - 1	CS only	CS – DES – 3DES
Field Unit - 2	CS only	CS – DES – 3DES

According to CIA, the field units are given the decryption devices with a hardcoded key, which means the key cannot be recovered without significant reverse engineering efforts that can take over a year. CIA also reports that field units do not have funding enough to acquire equipment to encrypt new data using DES and 3DES, which means they can only send information using Caesar Cipher with their existing equipment. The Caesar Cipher dependent devices can be configured, but the key must be known and decided upon prior to communication by both sending and receiving parties.

CIA thinks that it is unnecessary to risk its resources to find or decipher keys for Caesar Cipher-based equipment. Therefore, it is your unit's duty to compute the key for this cypher.

On the other hand, CIA is aware that DES and 3DES encrypted information will not be decrypted fast enough to be useful with any brute-force approach[2]. Therefore, their agents were able to use their assets to approach the head of covert communications, Mr. A. Gromov, at the enemy state's defense headquarters. CIA report suggests that Mr. A. Gromov, while sharing drinks with the asset, boasted that the key is the repetition of his birth year[3] and an extra character. The asset could not acquire more information and there is no way you can verify if this information is correct or just bragging. As for Mr. A. Gromov's age, CIA does not have the birth date or birth year of him. He can be anywhere from 35 to 50 years old.

[1] All thematic information given in this project document is fictional.
[2] With the technology of the year 1985
[3] If the birth year was 1985, 1985 would have been converted to hex 7C1. A 5-time repetition of this would give us a 15-character key, where we need 16 characters for a 64-bit key.

Your team is expected to report they keys used for all communications intercepted, the computation time for decryption process and verification of decryption success (output intelligibility) time.

Data Description

You will receive a comma separated text file called `input.csv` in which every line represents a communication intercepted from point A to point B. The first column represents the sender, the second column represents the receiver, the forth column represents the encryption algorithm, and the last column represents the intercepted ciphertext. An example file content is given below.

```
HQ, FU1, DES, XYZ
FU1, FU2, CS, UGHAWE
FU2, FU1, CS, OLKHU
HQ, FU2, 3DES, IUSJT
```

Note that there are no headers in this file.

Verification of Decryption Success

Verification of decryption success used to be performed with real people checking if the decrypted content is intelligible. Namely, every output of the algorithm is checked for every different key.

For your convenience, it is guaranteed that the communication is in English. Your verification process can be automatized with a 3rd party English dictionary[4]. I do not mind which dictionary you use, but make sure that you include it in your submission, and make sure that its uncompressed version is less than 10 MBs. You can also use a dictionary file that includes the 1000 most frequently used English words, and it would still work perfectly (so please no overkill here). Using the dictionary, you will check if the output includes any English words. Higher match means higher intelligibility. Do note that there can be numbers and names (location or person) in the text that will not appear in the dictionary, therefore, do not expect 100% of the words in the output will appear in the dictionary.

For Caesar Cipher, your keyspace is 26 characters, therefore, the verification process will be repeated only 26 times, and you will select the key and output that will give you the highest intelligibility.

For DES and 3DES, your keyspace is larger even with the intelligence provided to you. You will need to brute-force all alternative keys that conforms with the given intelligence, and select the key and output that will give you the highest intelligibility.

Notes

- There is no specific programming language to implement the project. You are free to choose. I recommend Python or Java, but this is just a recommendation.
- You can use built-in logical functions such as XOR but you cannot use existing implementations for Caesar Cipher or DES.

[4] Such as https://github.com/dwyl/english-words/blob/master/words_alpha.txt

- You can use internet resources such as stackoverflow.com for solving problems, BUT be careful about using existing code. Some other team may have used the same code block, and it would reflect as the two teams copied code from each other.
- 3DES algorithm uses the same key everytime for the purposes of this project. Note that, in practice, it uses 2 or 3 unique keys at each phase.

What to turn in

- An output.csv file that is in a similar format of the input.csv. The first column of each row represents the sender, second column represents the receiver, third column represents the encryption algorithm, forth column represents the key, and fifth column represents the plaintext (decrypted text)
- All the code in a zip file
- The dictionary that you used
- A short demo video explaining how to run your code with demonstration that your code works
- A one-page report that includes the following information:
    - Description of your approach to solving the problem
    - The table that shows how long it takes to try one key on average for each algorithm with your implementation and how long it takes to verify the intelligibility of the output on average in seconds
    - How you determined the keyspace for DES and 3DES
    - Mathematical calculation of how long it would take to find the DES key if you did not have the intelligence provided