Applied Crypto

CS-GY 6903 CF01/CF02

Insights from Classic Ciphers and Enigma Machine

Problem1: Affine Cipher

The Affine cipher is a type of monoalphabetic substitution cipher that employs modular arithmetic to encrypt the letters of a message. The encryption formula is c = (a * p + b) mod n, where 'p' is the plaintext letter, 'c' is the ciphertext letter, 'n' is the modulus (usually the size of the alphabet, which is 26 for English letters), 'a' and 'b' are keys, and 'a' must be chosen such that 'a' and 'n' are coprime (relatively prime). In this cipher, each letter of the plaintext is first converted into an integer: A=0, B=1, C=2, and so on up to Z=25.

1a) What is the size of key space for a fixed modular n? Students can use the notation of the Euler's totient $\Phi(n)$ which is defined as the number of integers less than n and relative prime to n.

1b) Imagine you're a cryptographer tasked with sending a secure message using the Affine Cipher. Your message consists only of capital letters, and you've decided to use the encryption formula $c = 5p+9$ mod 26. Your challenge is to encrypt a given plaintext, ensuring that spaces and other non-letter characters are omitted, as the domain of your cipher is limited to 26 capital letters. Write a python program to encrypt the phrase “CRYPTOISFUN”. Develop a general solution that can be applied to any plaintext using the specified Affine Cipher encryption formula. Explain your process clearly.

1c) Eve has intercepted a ciphertext “QJKESREOGHGXXREOXEO” and, through her intelligence sources, discovered that it’s encrypted using an Affine Cipher. She also has limited information about the encryption process: the letter 'T' is encrypted to 'H' and 'O' to 'E'. With this knowledge, she aims to decrypt the message. Your challenge is to help Eve by developing a method to find the decryption function of the Affine Cipher using the given information. You do it manually. Once the decryption function is determined, apply it to decrypt the ciphertext. Provide both the decrypted message and an explanation of your methodology. (Note: if you are not familiar with modular operations, you may hold on to this calculation until we are done with the first Math module.)

Problem 2 Frequency Analysis

Alice has crafted a message for Bob using a simple substitution cipher. The encrypted message, segmented is "TNFOS FOZSW PZLOC GQAOZ WAGQR PJZPN ABCZP QDOGR AMTHA RAXTB AGZJO GMTHA RAVAP ZW", where spaces are not part of the original encryption and are added only for convenience. Eve, who has intercepted the message, knows that the word "liberty" appears somewhere in the plaintext.

2a) Calculate the Size of the Key Space. Explain how the key space is calculated and its implications for the cipher's security.

2b) Given Eve's knowledge that the word "liberty" is in the plaintext, devise a strategy to decrypt the message. This task requires analyzing the ciphertext, making educated guesses, and testing hypotheses about the cipher's key. Your goal is to uncover the original message sent by Alice to Bob. You solve it manually.

Problem 3: Understanding and Analyzing the Enigma Machine.

The Enigma machine, used extensively during World War II, is a fascinating example of early mechanical encryption technology. With its complex system of rotors, reflectors, and plugboards, it offered a then-unprecedented level of security. Your task involves understanding the Enigma's encryption process, estimating the size of its key space, and performing cryptanalysis on a given ciphertext.

3a) Assess and calculate the size of the key space of the Enigma machine. Consider all elements that contribute to the key space: rotor wiring, ring settings, rotor stepping, reflector choices, plugboard configurations, and the initial position of rotors.

3b) Refer to the manual at https://py-enigma.readthedocs.io/_/downloads/en/latest/pdf/ Here is one code sample for enigma machine. Provide an explanation of the Enigma machine's code flow based on the given code using box (workflow) diagram. Just in case, you know you need to install the package using pip install py-enigma in term or !pip install py-enigma in jupyter notebook.

```
from enigma.rotors.rotor import Rotor
from enigma.plugboard import Plugboard
from enigma.machine import EnigmaMachine

rL = Rotor('my rotor1', 'EKMFLGDQVZNTOWYHXUSPAIBRCJ', ring_setting=0, stepping='Q')
rM = Rotor('my rotor2', 'BDFHJLCPRTXVZNYEIWGAKMUSQO', ring_setting=5, stepping='V')
rR = Rotor('my rotor3', 'ESOVPZJAYQUIRHXLNFTGKDCMWB', ring_setting=10, stepping='J')

reflector = Rotor('my reflector', 'YRUHQSLDPXNGOKMIEBFZCWVJAT')

pb = Plugboard.from_key_sheet('AK BZ CG DL FU HJ MX NR OY PW')

machine = EnigmaMachine([rL, rM, rR], reflector, pb)

machine.set_display('UPS')      # set rotor positions or use its default
position = machine.get_display()    # read rotor position
print(position)

# Encrypt A letter
#print(machine.key_press('C'))
# Encrypt a text
print(machine.process_text('Enigma machine is powerful for Q'))
```

3c) Test the code with different key configurations, altering various aspects like wiring, ring settings, stepping mechanisms, reflector types, plugboard presence, and initial display positions. No Submission.

3d) Test and observe the outcome when the plaintext has numbers or special characters. Write your observations. Suggest ways to improve the mechanism encryption.

3e) The codebreakers at Bletchley Park have intercepted a ciphertext “WVUVJCSQBFLEJGFNIZNIGYGOCWSUVNCIIIA” which they know corresponds to the plaintext “ATTACKXATXXXXXXATXATLANTICXZXISLAND”. It sounds like “ATTACKxATxxxxxx ATLANTICxZxISLAND”. Your challenge is to determine the initial rotor display position used to encrypt this message programmatically. Use your code to simulate the Enigma machine and discover the initial settings.

Submission Instruction

1. **Project report using PDF format (**proj1_your_team_number.pdf**)**
    - Ensure that your submission is clear, well-organized, and comprehensively covers all aspects of the assignment.
    - Use appropriate headings and mark the problem numbers clearly for easy reference.
    - Use the over page to list the team members and their contributions.
    - May include codes in the report. Students still need to submit the Jupyter Notebook.
2. **Jupyter Notebook** (proj1-Your_team_number.ipynb)
    - Students shall use Jupyter Notebook for the coding.
    - The Jupyter Notebook file shall contain all the coding solutions. It should be well-commented, indicating what each segment does, and should correspond to the problems as numbered in the assignment.
    - Ensure that the notebook is executable without errors and that the outputs of your code are visible. For example, add all the pip install statements in a code box at the top of your ipynb notebook if they are not installed already in the CA's machine.
    -

* If students use Jupyter Notebook to do all the work includes those discussion/essay problems. They still need to submit the exported PDF file that meets the requirement and the notebook.

* It is OK students use colab or any other cloud-based development environments for their team collaboration. When they submit the work, they need to submit the actual notebook code (..ipynb), not the link to their project.

Grading rubrics

	Max Point	Expectations
Problem 1	**3**	
1a	1	Find the formula correctly
1b	1	Find the cipher correctly
1c	1	Solve the decryption, show your work
Problem 2	**3**	
2a)	1	Get the formula correctly
2b)	2	Decrypt the message correctly
Problem 3	**6**	
3a	1	Write the formula accurately
3b	1	Draw the workflow diagram clearly with steps
3c	0	
3d	1	Answers are varied.
3e	3	Write the code and show the result. Follow code writing practice.