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 Φ(n). It is defined as the number of integers that are coprime 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 $c=5p+9 \bmod 26$. 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 can do it manually or programmatically. Once the decryption function is determined, apply it to decrypt the ciphertext. Provide both the decrypted message and an explanation of your methodology.

Problem 2 Frequency Analysis

Alice has crafted a message for Bob using a simple substitution cipher. The encrypted message, segmented for readability, is "TNFOS FOZSW PZLOC GQAOZ WAGQR PJZPN ABCZP QDOGR AMTHA RAXTB AGZJO GMTHA RAVAP ZW". In this ciphertext, 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. It is great if you can do it programmatically but not required.

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.

```
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 at least 5 different key configurations, altering various aspects like wiring, ring settings, stepping mechanisms, reflector types, plugboard presence, and initial display positions.

3d) The codebreakers at Bletchley Park have intercepted a ciphertext "WVUVJCSQBFLWSGTHDREWOSXYIAYEUBHHXY" which they know corresponds to the plaintext "ATTACK AT 5PM AT ATLANTIC Z ISLAND". 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

For Students Using a Local Jupyter Notebook:

1. **PDF File:** Compile a PDF document containing all your essay and computational answers. Ensure that this document is well-organized, with clear explanations and discussions for each problem. Use appropriate headings and mark the problem numbers clearly for easy reference.
2. **Jupyter Notebook:** Alongside the PDF, submit your Jupyter Notebook file (.ipynb) containing all the coding solutions. Your code 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 Students Using a Cloud-Based Jupyter Notebook:

1. **Single PDF File with Link:** If you're using a cloud-based Jupyter Notebook platform that is accessible to the course assistants (CAs) and the instructor, you have the option to submit a single PDF file. This file should include a link to your Jupyter Notebook in the cloud. Ensure that the permissions for the notebook are set correctly so that the CAs and instructor can access it. Include all essay and computational answers in the PDF, clearly marking the problem numbers.
2. **Alternative Two-File Submission:** Alternatively, you may choose to submit two files as in the case for local Jupyter Notebook users: one PDF file for all essay and computational answers and one Jupyter Notebook file for coding answers.

General Guidelines:

1. Ensure that your submission is clear, well-organized, and comprehensively covers all aspects of the assignment.
2. For coding problems, include comments and explanations in your notebook to clarify your approach and methodology.
3. Review the formatting and content of your PDF and Jupyter Notebook before submission to ensure readability and completeness.

Grading rubrics

	Max Point	Expectations
Problem 1	**10**	
1a	2	Find the formula correctly
1b	3	**Code** to find the cipher correctly
1c	5	Solve the decryption manually, show your work
Problem 2	**10**	
2a)	2	Get the formula correctly
2b)	8	Decrypt the message correctly, show at lease some process to the result
Problem 3	**20**	
3a	4	Write the formula accurately
3b	6	Workflow diagram clearly marked steps
3c	4	**Code** Demonstrate 7 test cases corectly
3d	7	**Code** shows the solution