|  |
| --- |
|  |
| Fall 2023/2024 | |  |
| **Cryptography Midterm**  **(Take-home Programming Test)** | | |

## Exam Availability:

This is an open-book take-home exam. It is available for 24 hours starting from 8:00 pm Thursday, December 14th.

## Open Book Exam:

This exam is take-home and open-book. Being open book means you can use your notes and you can search online for information.

The following is not allowed:

1. Collaboration in any form with other students
2. Collaborating with a third party inside or outside PSUT, virtual or in person 3- Posting exam-related questions online or in case someone else posted the

question to borrow the answers

1. Borrowing someone else’s code. All code presented in your exam files

should be your original work.

1. Using short code segments from online is acceptable as long as:
   1. It does not require using an external library
   2. It does not include advanced Python operations (not taught at PSUT like regular expressions and lambda calculus)
   3. Make a reference to the source. This should be done as a comment above the code segment.

Failing to abide by the above will be treated as an act of academic integrity violation.

## Frequent Submissions:

To avoid any last-minute surprises, make frequent submissions of your code to the E-Learning exam folder. The directory will be configured to only store your last submission, and all previous submissions will be discarded.

There are two hard-line rules with no exceptions:

1. You can not submit your solution through email, for any reason
2. No late submissions will be accepted. You have 50 hours, so plan to submit ahead of time.

## Exam Structure:

This second exam is composed of three programming questions. All questions are mandatory. The exam questions are:

1. CT Cipher
2. Mathx Cipher
3. Atberti Cipher

Each student is provided with three unique ciphertexts, referring to the first three questions, respectively.

## Exam Setup:

***Step 1: Preparing your Files***

Create a project/directory called midterm. You will use this project to host all your exam2 files. You can use any Python IDE of your choice.

Unzip your midterm.zip file. If you do not unzip the file, your work will not be saved.

Open the Task1\_set.zip, Task2\_set.zip and Task3\_set.zip files and extract your personal files. Your files will be named as:

ciphertext1\_first\_last.txt, ciphertext2\_first\_last.txt, ciphertext3\_first\_last.txt

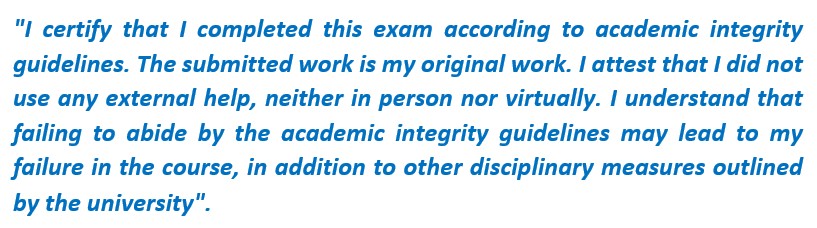
Copy the following files into your midterm Project (Total 8 files):

1. midterm\_template.py
2. midterm\_test.py
3. midterm\_output.txt
4. utilities.py
5. engmix.txt
6. ciphertext1\_first\_last.txt
7. ciphertext2\_first\_last.txt
8. ciphertext3\_first\_last.txt

Create the following three empty text files:

1. plaintext1\_first\_last.txt
2. plaintext2\_first\_last.txt
3. plaintext3\_first\_last.txt

In total, you should have 11 files in your midterm project.



***Step 2: Providing your Information:***

Rename the file midterm\_template.py to midterm.py

Open the midterm.py file. Perform the following:

Edit NAME: This should match how your name as spelled in the files given to you.

Under the Midterm.get\_ID() method, edit the return value to reflect your student ID.

Under the Midterm.get\_certification() method, edit the return value to reflect the following academic integrity pledge. No submission will be accepted without this pledge. Make sure that your string exactly matches the following:

***Step 3: Testing***

Open the file: midterm\_test.py file and run the code. Your output should not produce any exceptions (if you have applied Step 1 and 2).

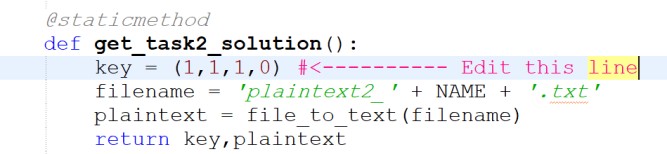
When you test your code, compare your output with midterm\_output.txt. There will be sections where the output is student-dependant. These sections are identified between < >.

***Step 4: Utilities and Imports***

Before proceeding, open the utilities.py file and carefully inspect the given functions. You can save a lot of time if you use the functions given to you in the utilities file.

You are not allowed to import any library other than utilities, math or copy. Also, you can not create new classes or methods inside your utilities.py or midterm.py files.

### Step 5: How to record your findings?



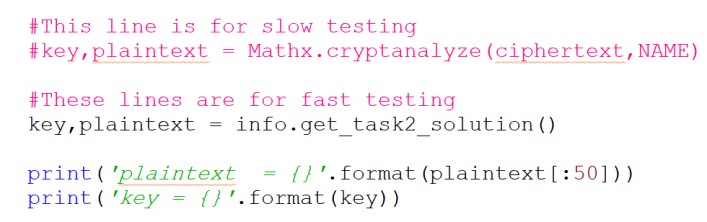
Your main objective is to break the three ciphertexts given to you. To document your solution, there are three things that you need to do.

***Action 1:*** Put the code that you used to break each ciphertext in the relevant class within midterm.py file. It is insufficient to break the ciphertexts using external tools and provide the recovered plaintexts. Instead, you need to provide the code and the recovered plaintexts/keys.

***Action 2:*** When you break each ciphertext, you need to put the key used to break the cipher under Midterm.get\_taskx\_solution().

For instance, if the key to break your ciphertext2.txt is (1,1,1,0), then you need to edit the key under get\_task2\_solution to look like the following:

***Action 3:*** If you successfully recover the plaintext, then you need to put the entire plaintext within the relevant plaintextx\_first\_last.txt file. Note that the testing file only displays the plaintext partially. To get the entire plaintext, you can either use utilities.text\_to\_file or edit the relevant



testing line to display the entire plaintext, and then copy it to the file, like the following:

Edit this line to:

(plaintext)

The first method is safer, as you might forget to copy some whitespaces, when you use manual copy.

***Step 6: Submission:***

Create a zip file called: "first\_last\_midterm.zip". The zip file should contain ONLY the following seven files:

1. midterm.py
2. plaintext1\_first\_last.txt
3. plaintext2\_first\_last.txt
4. plaintext3\_first\_last.txt
5. ciphertext1\_first\_last.txt
6. ciphertext2\_first\_last.txt
7. ciphertext3\_first\_last.txt

### Grading Notes:

The midterm is graded out of 15 points. Each question is worth 5 points. The cryptanalysis is worth 3/5 and the rest of the code is worth 2/5. Therefore, the cryptanalysis is 60% of your midterm grade.

You will not be graded on efficiency. However, if your cryptanalysis takes too much time, then the instructor will interrupt the program, and a deduction of 10% will be applied (assuming that you have correct output in the plaintext files).

For any task, if running the relevant testing function leads to errors, i.e., program crashes, a deduction of 25% will be applied to that task's grade. If the instructor was not able to fix the error, a grade 0 is assigned to that task.

Each of the following, will lead to 1 pt deduction, with maximum total of 4 points deduction:

1. Making incorrect file naming, for any file.
2. Failing to submit a single zip file with the proper name

3- Failing to include the correct number of files in your zip submission 4- Forgetting to write your NAME in midterm.py

1. Forgetting to write your ID in midterm.py
2. Importing a module, other than those permitted. All lines that use that import will be commented.
3. Failing to provide plaintext files. Even if you were unable to break the ciphertext, you need to provide empty plaintext files.

Any submission that does not have the academic integrity certification will be rejected. This is similar to forgetting to sign an official document.

# Task 1: CT Cipher (5 pts)

This task is relevant to the ciphertext1\_first\_last.txt file.

The ciphertext is produce using columnar transposition cipher. You know the following:

1. The key is a keyword from the dictionary 'engmix.txt'
2. The key contains the first character of your first name
3. The columnar transposition orders the keyword characters in the reverse order. For example, keyword mark becomes rmka not akmr.
4. The length of the keyword is even.

# Task 2: Mathx Cipher (5 pts)

This task is relevant to the ciphertext1\_first\_last.txt file.

The ciphertext is produce using some mathematical cipher. You know the following:

[1] The equation is 𝑦 = (𝑎𝑏2 − 𝑐)𝑥 + 𝑏𝑐 𝑚𝑜𝑑 𝑚

1. The key is a tuple of (a,b,c,m), where all of them are non-negative integers. If m = 48, then the base is BASE[:48].
2. The key demonstrates Affine cipher behavior.
3. The value of m is larger or equal to 46.

# Task 3: Atberti Cipher (5 pts)

This task is relevant to the ciphertext3\_first\_last.txt file.

This is not a typo. The cipher is called: Atberti not Alberti, because it is a combination of Atbash + Alberti ciphers.

You know the following:

1. The key is (base, pointer). The argument base is a string representing one of the strings defined in BASES. The pointer is a character in the defined

base. For instance, if base = 'lower', then pointer should be one of the lower-case characters.

1. The outer wheel contains the characters defined in the base. The inner wheel is its atbash, i.e., the reverse of the outer wheel. The first character in the outer wheel is aligned with the pointer in the inner wheel.

Good luck!