

CS 5158/6058 Data Security and Privacy,

Fall 2022

Project 2: Symmetric-Key Encryption

Margi Amin M15219371

Software: Python 3.6.3 is required to run this if you're using Windows 10.

Program Location: ..\prp\_m15219371\prp

Files and structure:

Files included in program:

-ciphertext.txt

-permutation.txt

- permutation.py

-prp.py

-prp\_enc\_dec .py

-PseudorandomPermutation.txt

**Description:**

a) **Permutation\_\_Family function:**

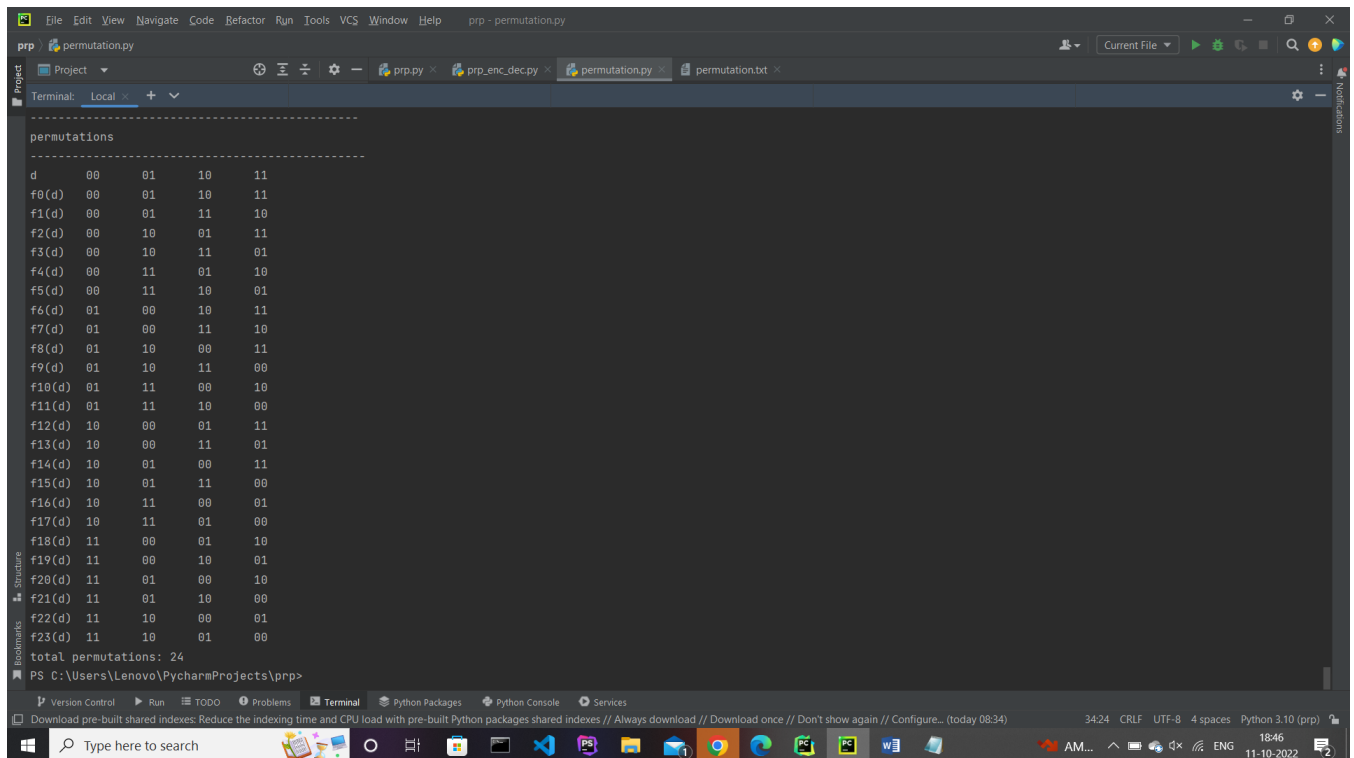
This program generates all the permutation of permutation family  $Perm(n,n)$  and saves those in the file permutation.txt .

Command to run:

python permutation.py -permutegen n=2 permutation.txt

Output:

Here is screenshot of output of the permutationFamily function with input of n=2 . and it gives 24 permutation as req.



The screenshot shows a PyCharm IDE window with a terminal output. The terminal displays a table of permutations for a 4x4 matrix. The table lists 24 permutations, each represented as a function of the input vector d. The permutations are listed in a specific order, and the total number of permutations is 24.

```
permutations
-----
d      00      01      10      11
f0(d) 00      01      10      11
f1(d) 00      01      11      10
f2(d) 00      10      01      11
f3(d) 00      10      11      01
f4(d) 00      11      01      10
f5(d) 00      11      10      01
f6(d) 01      00      10      11
f7(d) 01      00      11      10
f8(d) 01      10      00      11
f9(d) 01      10      11      00
f10(d) 01      11      00      10
f11(d) 01      11      10      00
f12(d) 10      00      01      11
f13(d) 10      00      11      01
f14(d) 10      01      00      11
f15(d) 10      01      11      00
f16(d) 10      11      00      01
f17(d) 10      11      01      00
f18(d) 11      00      01      10
f19(d) 11      00      10      01
f20(d) 11      01      00      10
f21(d) 11      01      10      00
f22(d) 11      10      00      01
f23(d) 11      10      01      00

total permutations: 24
PS C:\Users\Lenovo\PycharmProjects\prp>
```

**b) Pseudorandom permutation function:**

This program generate all the permutation included in psedorandom permutation  $F: K \times D \rightarrow R$  . with parameter n and l . and saves the output in pseudo\_permutation.txt file.

Command to run:

Python prp\_enc\_dec.py -prpgen n=4 l=4 pseduo\_permutation.txt

Output:

Here is the screenshot of output where n=4 and l=4.

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help prp - permutation.py
Project
Terminal Local +
f19(d) 11 00 10 01
f20(d) 11 01 00 10
f21(d) 11 01 10 00
f22(d) 11 10 00 01
f23(d) 11 10 01 00
total permutations: 24
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -prpgen n=4 l=4 pseduo_permutation.txt

-----
Pseudorandom Permutations
-----
d      0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
k=0000,f(d) 1101 0011 1100 0001 1111 1110 1001 0010 0000 0111 1011 0101 0110 1000 0100 1010
k=0001,f(d) 1011 0000 0110 1010 0111 0101 0010 1000 0011 1100 1001 0001 1111 0100 1110 1101
k=0010,f(d) 1111 1000 0110 0011 1100 1001 1110 1010 0000 0101 0001 0010 0100 0111 1101 1011
k=0011,f(d) 1001 1110 1010 0000 0010 0100 0101 0001 1011 0011 0111 1100 1111 1000 0110 1101
k=0100,f(d) 1110 0001 1000 1010 0101 0000 1001 1111 0010 0100 1101 1100 0111 0110 0011 0111
k=0101,f(d) 1000 1011 0001 0000 0111 1010 1111 1101 0101 0011 1100 0110 0100 1001 1110 0010
k=0110,f(d) 1000 0000 1111 1110 0110 0100 1001 1010 1101 1011 0001 1100 0101 0010 0111 0011
k=0111,f(d) 0001 1011 0000 1111 1110 1010 0100 0011 1001 1101 0010 1000 0101 1100 0110 0111
k=1000,f(d) 1101 0010 1100 1111 1001 1110 0110 1010 1011 0000 0011 0100 0101 0001 1000 0111
k=1001,f(d) 1011 0001 0110 1111 0000 1100 1010 1000 0011 0101 0111 0010 0100 1101 1001 1110
k=1010,f(d) 1010 1001 1110 1011 0111 0000 0010 1101 0100 1000 0110 0011 0101 1100 0001 1111
k=1011,f(d) 1101 0001 1010 0000 0011 1110 1100 0111 0010 1001 0100 1000 1111 0110 1011 0101
k=1100,f(d) 1011 1000 0101 0111 1101 0011 0000 1100 0100 1010 0110 0010 0001 1111 1110 1001
k=1101,f(d) 1101 0000 1000 1100 0011 0001 1110 1011 1001 0010 1010 0111 0110 0101 0100 1111
k=1110,f(d) 1110 1111 1010 0111 0101 0100 1101 0110 1100 1001 0011 1011 0000 0001 1000 0010
k=1111,f(d) 1100 0000 0010 0001 1101 1000 0011 0100 1111 0110 1010 1110 0101 1001 1011 0111
PS C:\Users\Lenovo\PycharmProjects\prp>

```

### c) Encryption with block cipher CBC mode:

This function encrypt the message in CBC mode with 4-bit key , generate a 4-bit random IV and use prp and output ciphertext and saves in the ciphertext.txt file.

Command to run:

Python prp\_enc\_dec.py -enc\_cbc m=100111000011 l=4 k=1100 pseduo\_permutation.txt  
cipher.txt n=4 l=4 pseduo\_permutation.txt

Output:

Here is screenshot of output with random Iv and different ciphertext every time.

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help prp - permutation.txt
Project
Terminal Local +
IV= 0011
0011, 0110 0110 0011
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1000
1000, 1000 1101 1110
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1110
1110, 1100 1011 0100
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1100
1100, 0011 1001 0110
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1010
1010, 0111 0010 1000
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1000
1000, 1000 1101 1110
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1100
1100, 0011 1001 0110
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 0101
0101, 0001 1111 0001
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1000
1000, 1000 1101 1110
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt
IV= 1100
1100, 0011 1001 0110
PS C:\Users\Lenovo\PycharmProjects\prp>

```

#### d) Decryption with block cipher in CBC mode:

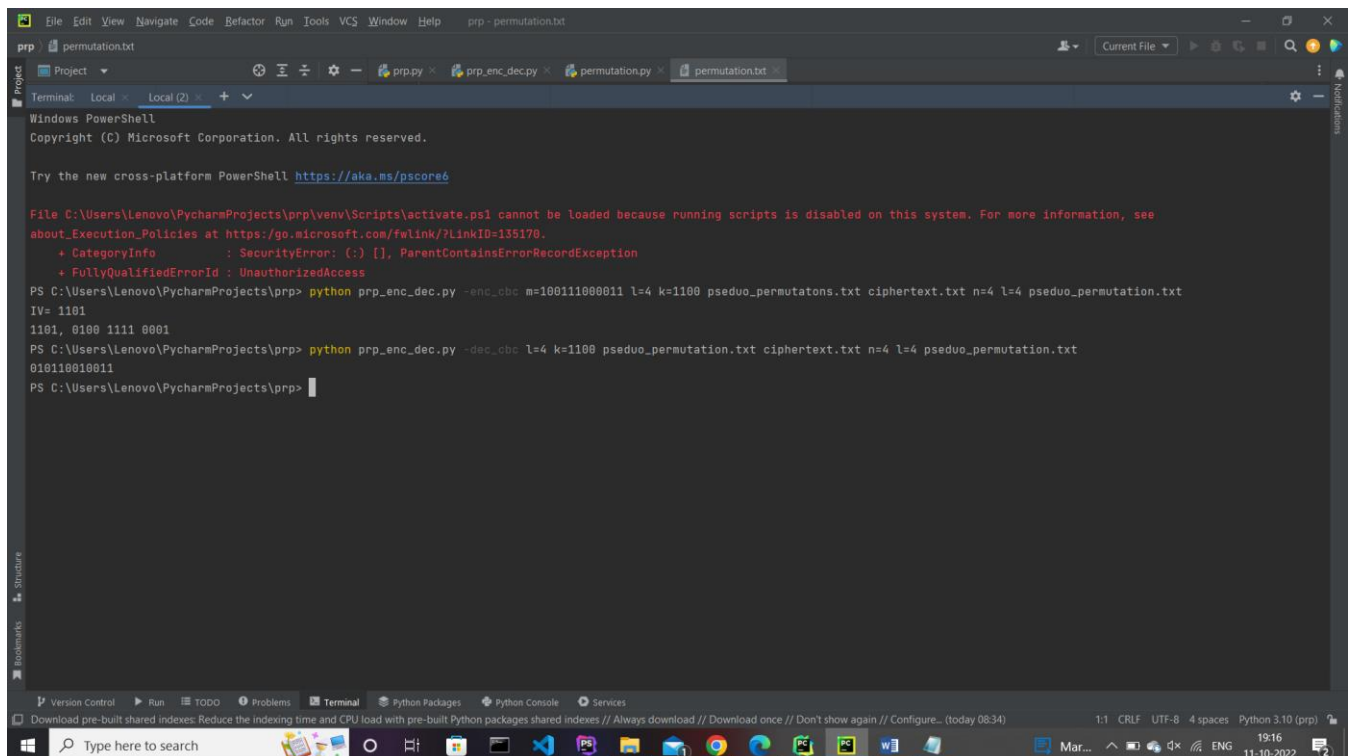
In this function cipher text in the cipher.txt file is decrypted with the given key and recovers the message and prints it in terminal.

Command to run:

```
Python prp_enc_dec.py -dec_cbc l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4  
pseduo_permutation.txt
```

Output:

Here is the screenshot of output.



```
prp - permutation.txt  
prp.py x prp_enc_dec.py x permutation.py x permutation.txt x  
Terminal: Local x Local (2) x  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
File C:\Users\Lenovo\PycharmProjects\prp\venv\Scripts\activate.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see  
about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.  
+ CategoryInfo          : SecurityError: (:) [], ParentContainsErrorRecordException  
+ FullyQualifiedErrorId : UnauthorizedAccess  
  
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -enc_cbc m=100111000011 l=4 k=1100 pseduo_permutations.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt  
IV= 1101  
1101, 0100 1111 0001  
  
PS C:\Users\Lenovo\PycharmProjects\prp> python prp_enc_dec.py -dec_cbc l=4 k=1100 pseduo_permutation.txt ciphertext.txt n=4 l=4 pseduo_permutation.txt  
010110010011  
  
PS C:\Users\Lenovo\PycharmProjects\prp>
```

#### e) Encryption with block cipher in ECB mode:

This function encrypt the message in ECB mode with 4-bit key and use prp and output ciphertext and saves in the ciphertext.txt file.

Command to run:

```
Python prp_enc_dec.py -enc_ecb m=100111000011 l=4 k=1100 pseduo_permutation.txt  
cipher.txt n=4 l=4 pseduo_permutation.txt
```

Output:

Here is screenshot of output with same ciphertext every time.

The image shows a Windows PC screen. The main part of the screen is occupied by the Visual Studio Code (VS Code) editor. The top of the VS Code window shows a menu bar with options: File, Edit View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. Below the menu bar is a toolbar with icons for saving, opening, and other file operations. The editor's title bar shows the project name 'prp' and the file 'permutation.txt'. The editor area displays the content of 'permutation.txt', which includes a copyright notice for Microsoft Corporation, a link to a PowerShell tutorial, and a security error message: 'SecurityError: (:) [], ParentContainsErrorRecordException + FullyQualifiedErrorId : UnauthorizedAccess'. Below the error message, there is a series of commands and their outputs, all involving the 'python prp\_enc\_dec.py' command with various parameters. The bottom of the VS Code window shows a sidebar with tabs for 'Version Control', 'Run', 'TODO', 'Problems', 'Terminal', 'Python Packages', 'Python Console', and 'Services'. The 'Terminal' tab is active, showing the same commands and outputs as the editor area. The Windows taskbar is visible at the bottom of the screen, showing the Start button, a search bar, and several pinned applications including File Explorer, VS Code, and a web browser. The system tray in the bottom right corner shows the date and time as '11-10-2022 19:25' and the language as 'ENG'.