

RSA Sandbox User Manual

Created by Aron Schwartz and Professor John Acken
Portland State University | Winter 2020 | ECE Department

Table of Contents

1.0 Overview	3
1.1.0 Running the Tool	3
1.2.0 Data and Profile Folders	4
1.3.0 Default Profiles	4
2.0 System Data Management	5
2.1.0 Prime Number Management	5
2.2.0 Septuple Objects	7
2.2.1 Septuple Management Menu	8
2.2.2 The Active Septuple	8
2.2.3 Septuple Creation	8
2.2.4 Septuple Creation from Primes	9
2.3.0 Key Management	9
2.4.0 Displaying System Data	12
2.5.0 Saving and Loading System Data	12
3.0 Encrypting Plaintext and Strings	13
3.1.0 Encrypting Plaintext Files	13
3.2.0 Encrypting input strings	14
4.0 Fixed Point Analysis	15
4.1.0 Active Septuple Analysis and Septuple Comparison	15
4.2.0 Key Analysis and Transparency Profile Generation	17
5.0 Help Menu	19

1.0 RSA Sandbox Overview

The RSA Sandbox is a research tool that enables users to explore and experiment with the RSA encryption algorithm. The program allows a user to create RSA encryption “septuples”, generate prime numbers, create encryption keys, and combine the three to search for fixed point patterns. In addition, users can encrypt plaintext to file, output results in comma separated format, and save all system data to a user profile to pick up again later.

1.1 Running the Tool

The tool is written in Python 3, and is intended to be run from the windows command line. Prior to running the program, all source code can be downloaded from the following repository:

Link to Repository: https://github.com/aronjschwartz/RSA_Encryption.git

A screenshot of the main menu of the program can be seen below:

```
*****
*
*      Welcome to the RSA Encryption Sandbox
*
*****

***** MAIN MENU *****

1 - Encrypt/Decrypt without padding
2 - Encrypt/Decrypt with padding
3 - Manage Septuples
4 - Manage Keys
5 - Manage Prime Numbers
6 - Manage Plaintext
7 - Analyze Fixed Points
8 - Display System Data
9 - Save/Load System data
M - Reload main menu
H - Help Topics
Q - Exit program

Enter selection (M/m to reload menu):
```

Figure 1: The main menu is loaded upon initialization of RSA_Sandbox.py

To run the program, navigate to the directory housing the repository from the command line and run the RSA_Sandbox.py file. A screenshot illustrating initialization of the program can be seen below.

```

C:\Users\aronj\Grad_School_Classwork\Fall_2019\RSA_Encryption_Research\RSA_Encryption\Python_Code>python RSA_Sandbox.py
*****
*                                                                 *
*      Welcome to the RSA Encryption Sandbox                      *
*                                                                 *
*****

***** MAIN MENU *****
1 - Encrypt/Decrypt without padding
2 - Encrypt/Decrypt with padding
3 - Manage Septuples

```

Figure 2: Run the program by initializing RSA_Sandbox.py from the command line

1.2 Data and Profile Folders

When the program is initialized, it will check for the existence of four folders in its working directory and create them if they do not exist. They are as follows:

1. **Profiles** – Contains user profile sub-folders containing system data files in .csv format. Profile data can be created and loaded from the system data management menu.
2. **Results** – Contains output .csv files resulting from fixed point analysis
 - **/Results/Full_Septuple_Key_Comparisons/** : Contains output data for all septuple/all key analysis
 - **/Results/Septuple_Comparisons/** : Contains output data for septuple analysis
 - **/Results/Septuple_Transparency_Profiles/** : Contains output data for transparency profile generation
3. **Plaintext** – Contains plaintext files. Users can add text files to this folder to encrypt them
4. **Ciphertext** – Contains ciphertext output resulting from encrypting plaintext files

1.3 Provided Profiles

There are two profiles pre-generated and included in the Profiles directory. A description of both can be seen below.

Profile 1 “special septs”: This folder contains 5 historical septuples that are significant to the RSA community, such as the septuple used in the original RSA paper, the septuple used in the paper that described fixed points, the septuple from Behnaz’s thesis, etc.

Profile 2 “original_data”: This folder contains all possible septuples in the first 100 primes with the public keys (3, 5, 17, 257, 65537) appended to the key list for each septuple. This data is identical to the original data that demonstrated totient patterns in high transparency septuples. There are over 24000 unique septuples in this data set, and this profile can serve as a nice way to get started using the sandbox.

2.0 System Data Management

The RSA Sandbox allows creation and manipulation of septuples, encryption keys, and prime numbers. A description of each of these three elements, their respective relationships, and the mechanisms for properly utilizing them can be seen in the following sections.

2.1.0 Prime Number Management

The program maintains a list of prime numbers that can be used to generate keys and septuples. The prime number management menu can be accessed by pressing '5' from the main menu, as shown below.

```
Enter selection (M/m to reload menu): 5

***** PRIMES MANAGEMENT MENU *****

1 - Generate prime list
2 - Generate prime list (display only)
3 - Add custom prime to list
4 - Clear prime list
5 - Display prime list
Q - Return to main menu

Enter selection (q to quit):
```

Figure 3: The prime number management menu allows creation of internal primes list

Both options '1' and '2' will create prime numbers, however option '2' will only display the list and will not append it to the internal data. To generate primes, the user simply inputs a starting and ending value. The program will generate a list of every prime number between the two provided values. The following screenshot shows an example process of generating all prime numbers between 2 and 5000.

```
Enter selection (q to quit): 1
Enter lower limit for prime generation: 2
Enter upper limit for prime generation: 5000
[3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
7, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 3
, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 57
757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853
1039, 1049, 1051, 1061, 1063, 1069, 1087, 1091, 1093, 1097, 1103, 11
1291, 1297, 1301, 1303, 1307, 1319, 1321, 1327, 1361, 1367, 1373, 138
553, 1559, 1567, 1571, 1579, 1583, 1597, 1601, 1607, 1609, 1613, 1619
11, 1823, 1831, 1847, 1861, 1867, 1871, 1873, 1877, 1879, 1889, 1901,
7, 2089, 2099, 2111, 2113, 2129, 2131, 2137, 2141, 2143, 2153, 2161,
, 2371, 2377, 2381, 2383, 2389, 2393, 2399, 2411, 2417, 2423, 2437, 2
2663, 2671, 2677, 2683, 2687, 2689, 2693, 2699, 2707, 2711, 2713, 27
2909, 2917, 2927, 2939, 2953, 2957, 2963, 2969, 2971, 2999, 3001, 301
221, 3229, 3251, 3253, 3257, 3259, 3271, 3299, 3301, 3307, 3313, 3319
17, 3527, 3529, 3533, 3539, 3541, 3547, 3557, 3559, 3571, 3581, 3583,
9, 3793, 3797, 3803, 3821, 3823, 3833, 3847, 3851, 3853, 3863, 3877,
, 4079, 4091, 4093, 4099, 4111, 4127, 4129, 4133, 4139, 4153, 4157, 4
4373, 4391, 4397, 4409, 4421, 4423, 4441, 4447, 4451, 4457, 4463, 44
4673, 4679, 4691, 4703, 4721, 4723, 4729, 4733, 4751, 4759, 4783, 478
987, 4993, 4999]

Generated 668 primes
```

Figure 4: 668 primes are generated for the input range 2 to 5000

As shown, the system will generate the primes within the given input range. A final message showing the total number of primes that were generated is also shown, in this case 668 values. As with the septuple menu, other options exist to simply view the prime list or to clear it away entirely.

The user can also choose option '3' to add custom primes to the prime list. This is useful for creating lists of primes that are not sequential in nature, such as the public keys 3, 5, 17, 257, 65537. An example of creating custom primes using this option can be seen below.

```
Enter selection (q to quit): 3
Enter prime to add (empty return when done): 3
3 appended to prime list
Enter prime to add (empty return when done): 5
5 appended to prime list
Enter prime to add (empty return when done): 17
17 appended to prime list
Enter prime to add (empty return when done): 257
257 appended to prime list
Enter prime to add (empty return when done): 65537
65537 appended to prime list
Enter prime to add (empty return when done):

***** PRIMES MANAGEMENT MENU *****

1 - Generate prime list
2 - Generate prime list (display only)
3 - Add custom prime to list
4 - Clear prime list
5 - Display prime list
Q - Return to main menu

Enter selection (q to quit): 5
[3, 5, 17, 257, 65537]
```

Figure 5: Custom prime values can be appended to the list for increased flexibility

2.2.0 Septuple Objects

Performing encryption in the RSA Sandbox is accomplished by the creation of septuple objects, a python object that houses the 7 parameters and internal functionality needed to perform RSA encryption and decryption. The 7 parameters of a septuple and a brief description can be seen below.

- **P and Q:** Two initially chosen prime numbers which govern other parameter choices
- **N:** Product of P and Q
- **Totient:** Product of (P-1)*(Q-1)
- **E:** Encryption exponent, part of the public encryption key (**E, N**)
- **D:** Decryption exponent, part of the private decryption key (**D, N**)
- **K:** Internal parameter related to choosing D

For consistency, septuples are represented in this manual, the RSA Sandbox program, and in output files in the following format

Septuple format: [P, Q, N, Totient, E, K, D]

2.2.1 The Septuple Management Menu

The Septuple Management menu allows the creation, viewing, and manipulation of septuple objects. The septuple management menu can be accessed from the main menu by pressing ‘3’ as seen below.

```
Enter selection (M/m to reload menu): 3

*** Displaying septuple list (2 loaded) ***
0 - [17, 23, 391, 352, 65537, 12102, 65] *** active ***
1 - [17, 257, 4369, 4096, 101, 9, 365]

***** SEPTUPLE MANAGEMENT MENU *****

1 - Change active septuple
2 - Add Septuple
3 - Create septuples from primes
4 - Clear septuples
5 - View septuples
Q - Return to main menu

Enter selection (q to quit):
```

Figure 6: Septuple management menu accessed by pressing ‘3’ from main menu

As shown in the screenshot, initializing the septuple menu will initially display all septuples loaded in the program, with a descriptor string accompanying the “active” septuple. This list can also be seen by pressing ‘5’ from the septuple management menu as shown, and the list can be cleared by pressing ‘4’.

2.2.2 The Active Septuple

The “active” septuple is the septuple that will be used by default when performing plaintext and/or input string encryption. The user can choose a new active septuple from among the loaded septuples at any time by pressing ‘1’ in the septuple management menu. There is also an opportunity to specify a septuple other than the active one when performing plaintext encryption.

2.2.3 Septuple Creation

Option ‘2’ allows a new septuple to be generated from user input and added to system whilst ensuring proper values are provided for P, Q, and E. A screenshot of a typical septuple creation flow can be seen below.

```

1 - Change active septuple
2 - Add Septuple
3 - Create septuples from primes
4 - Clear septuples
5 - View septuples
Q - Return to main menu

Enter selection (q to quit): 2
Creating septuple..
No primes list loaded
Enter p: 17
Enter q: 257
Specify e? Press enter to use default (65537) [Y/N]: n

Encryption object created!
P: 17
Q: 257
N: 4369
T: 4096
E: 65537
D: 1
K: 16

```

Figure 6: Septuples can be created from the command line by inputting P, Q, and E

As shown, the program will prompt the user to input P and Q. The user can also input E, or use the default value of 65537. Error messages will be generated and the object creation will fail if bad and/or incompatible values are provided.

```

Enter selection (q to quit): 2
Creating septuple..
Enter p: 16
16 is not a prime number!
Object creation failed!

***** SEPTUPLE MANAGEMENT MENU *****

```

Figure 7: System will prevent incorrect P, Q, or E input values

2.2.4 Septuple Generation from Primes

In addition to manually creating septuples one at a time, septuples can be generated from the internal list of primes. Choosing option '3' will take all possible unique combinations of numbers in the internal prime and use those values for P and Q to generate septuples. E values

are randomly chosen from among the “public key” list (3, 5, 17, 257, 65537). An example of this mechanism creating all septuple combinations for a prime list between 2 and 200 can be seen below.

```

1 - Change active septuple
2 - Add Septuple
3 - Create septuples from primes
4 - Clear septuples
5 - View septuples
Q - Return to main menu

Enter selection (q to quit): 3

Generating septuples....

Generated 990 septuples

```

Figure 8: Septuples can be generated automatically using the internal prime number list

2.3.0 Key Management

The program maintains an internal dictionary associating a list of encryption keys with each septuple. The initial key used in septuple creation is first added to the dictionary, and the user can add more keys to any septuple via custom input in addition to creating them directly from the prime number list. The key management menu can be accessed by pressing ‘4’ from the command line as seen below.

```

Enter selection (M/m to reload menu): 4
Key generation selected

***** KEY MANAGEMENT MENU *****

1 - Add keys
2 - Swap keys
3 - Clear keys
4 - View keys
5 - Add keys from prime list
6 - Add keys to all septuples from prime list
Q - Return to main menu

Enter selection (q to quit):

```

Figure 9: The key management menu allows creation and manipulation of public keys

An example of option ‘4’, i.e. “View Keys” following creation of a single septuple is shown below.

```

Enter selection (q to quit): 4

*** Displaying septuple list (1 loaded) ***
0 - [17, 59, 1003, 928, 257, 18, 65] *** active ***

Select septuple to view keys: 0
( 1003 , 257 )

```

Figure 10: Single septuple with a single key

As one can see, the key list is shown in the form (N, E), and in this case the N and E values are identical to the values occupying the septuple object itself since no other keys exist yet.

Option '1' allows a user to append a new key to a septuple via command line input. For more expansive key generation, option '5' can be used to append the entire prime list to the key list of a chosen septuple, and further still, option '6' can be used to append the prime list to every single septuple loaded in the program. The following screenshots illustrate an example of this mechanism. This assumes that prior to entering the key management menu, the user has already generate an internal prime list of all primes between 2 and 100.

```

Enter selection (q to quit): 5

*** Displaying septuple list (1 loaded) ***
0 - [17, 59, 1003, 928, 257, 18, 65] *** active ***

Select septuple to add keys: 0
Key 3 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 5 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 7 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 11 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 13 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 17 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 19 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 23 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 29 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 31 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 37 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 41 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 43 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 47 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 53 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 59 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 61 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 67 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 71 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 73 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 79 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 83 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 89 appended to septuple [17, 59, 1003, 928, 257, 18, 65]
Key 97 appended to septuple [17, 59, 1003, 928, 257, 18, 65]

```

Figure 11: Prime list can be appended to key list of a septuple

As shown, the primes between 2 and 100 are added to the chosen septuple. In contrast with **Figure 9**, the following screenshot shows the new output upon viewing the keys for the same septuple.

```

*** Displaying septuple list (1 loaded) ***
0 - [17, 59, 1003, 928, 257, 18, 65] *** active ***

Select septuple to view keys: 0
( 1003 , 257 )
( 1003 , 3 )
( 1003 , 5 )
( 1003 , 7 )
( 1003 , 11 )
( 1003 , 13 )
( 1003 , 17 )
( 1003 , 19 )
( 1003 , 23 )
( 1003 , 29 )
( 1003 , 31 )
( 1003 , 37 )
( 1003 , 41 )
( 1003 , 43 )
( 1003 , 47 )
( 1003 , 53 )
( 1003 , 59 )
( 1003 , 61 )
( 1003 , 67 )
( 1003 , 71 )
( 1003 , 73 )
( 1003 , 79 )
( 1003 , 83 )
( 1003 , 89 )
( 1003 , 97 )

```

Figure 12: Twenty four additional keys now exist in the key list

It is worth noting that the loaded key in the septuple is 257. Option ‘2’ in the key management menu allows choosing a new key from a septuples key list. The septuple object will regenerate the D and K values accordingly upon swapping of the key value. For convenience, a user can enter a key value that does not exist when swapping keys. The program will detect this, ensure co-primality is met, and perform the swap whilst adding the new key to the dictionary.

2.4.0 Displaying System Data

All primes, septuple, key data, and active plaintext file can be viewed by pressing option ‘8’ from the main menu. The program will allow the option of verbosely viewing all key data in addition to septuples, primes, and plaintext data.

```

Enter selection (M/m to reload menu): 8
View keys verbosely? [Y/N]:n

0 - [17, 23, 391, 352, 65537, 12102, 65] ( 46 keys loaded)
1 - [17, 59, 1003, 928, 101, 85, 781] ( 45 keys loaded)
***** Primes *****
All primes between: [3, 199]

Plaintext loaded from file: None

```

Figure 13: All system data can be displayed via the display system data menu option

2.5.0 Saving and Loading System Data

Users can save all septuples, keys, prime numbers, and active septuple information to a personal profile for later restoration. The data saving/loading menu can be accessed by pressing ‘9’ from the main menu as seen below.

```

Enter selection (M/m to reload menu): 9

***** SYSTEM DATA MANAGEMENT MENU *****

1 - Save data to profile
2 - Load data from profile
Q - Return to main menu

Enter selection (q to quit):

```

Figure 14: Data can be saved and re-loaded from the system data management menu

As mentioned in section 1.0, a “Profiles” directory is created upon program bootup if it does not exist. Option ‘1’ will prompt a user to enter a name to associate with their data. Once this is entered, all system data is saved to .csv files inside a folder with the provided name. To re-load the data, a user can choose option ‘2’ and enter the name user previously to save the data. The system will find the profile with the same name and load the contents of the .csv files back into the program. A screenshot of a profile and its four internal data files can be seen below

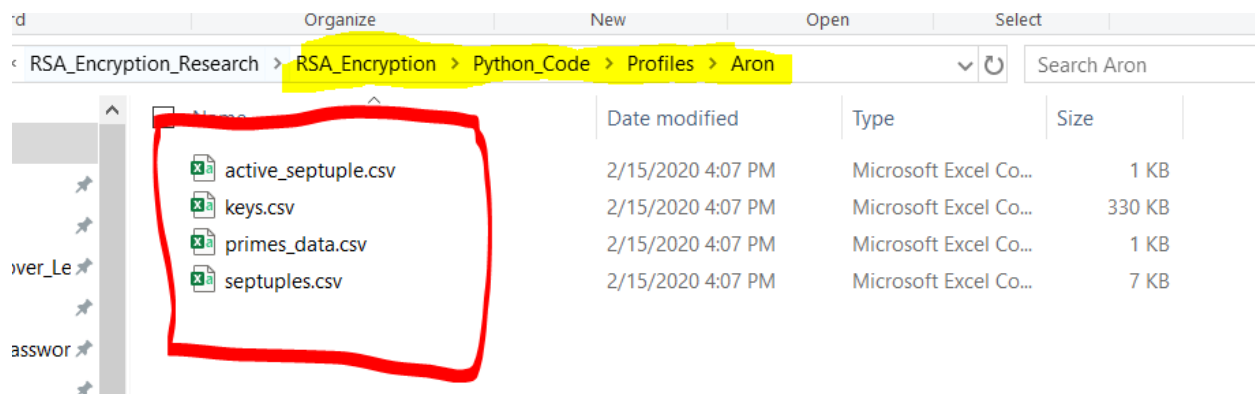


Figure 15: CSV files save the four system data components and can be edited directly

Users can take advantage of this mechanism by editing the data files directly. Upon re-loading the profile, the new data will be re-loaded. This can be used to enter desirable septuples directly into the .csv file for instance and remove the need to create the objects within the program.

3.0.0 Encrypting Plaintext files and Strings

The Sandbox can be used to encrypt plaintext files and input strings. An explanation of both mechanisms can be seen in the following sections.

3.1.0 Encrypted Plaintext Files

Users can use the active septuple to perform encryption/decryption plaintext files. A screenshot of the non-padding encryption menu can be seen below.

```
Enter selection (M/m to reload menu): 1
Encrypt/Decrypt without padding selected
Active septuple: [17, 23, 391, 352, 65537, 12102, 65]

***** ENCRYPTION MENU *****

1 - Encrypt plain text file
2 - Encrypt an input string
Q - Return to main menu

Enter selection (q to quit):
```

Figure 16: The active septuple can be used to encrypt plaintext and input strings

If no active encryption object yet exists, the program will detour and prompt the user to create a valid septuple.

Option one will read all text files in the Plaintext folder and prompt the user to select the plaintext file they wish to encrypt. An example can be seen below.

```
Enter selection (q to quit): 1
0 - plaintext_0.txt
1 - plaintext_1.txt
Enter selection (q to quit): 0

Plaintext set to file: plaintext_0.txt
Use active object? [17, 23, 391, 352, 65537, 12102, 65]? [Y/N]: y

The plain text is: RSA encryption is asymmetric in nature, not symmetric.

The ciphertext is: ĆİÉÊđŋĞrā=tĤ÷ŋÊĤsÊċsāŮŮđtrĤĞÊĤŋÊŋċtŮrdōÊŋ÷tÊsāŮŮđtrĤĞ.

Decrypted: RSA encryption is asymmetric in nature, not symmetric.
```

Figure 17: User selects plaintext to encrypt, results are shown and output to Ciphertext folder

The resulting ciphertext from the encryption will be appended in a text file format to the Ciphertext folder with the plaintext file and public key used incorporated into the file name, as seen below.





RSA_Encryption_Research > RSA_Encryption > Python_Code > Ciphertext					Search Ciphertext
<input type="checkbox"/> Name	Date modified	Type	Size		
 cipher_plaintext_0_(323,80).txt	2/14/2020 12:02 PM	Text Document	1 KB		
 cipher_plaintext_0_(391,65537).txt	2/21/2020 12:29 PM	Text Document	1 KB		
 cipher_plaintext_0_(1357,101).txt	2/15/2020 2:15 PM	Text Document	1 KB		
 cipher_plaintext_1_(391,65537).txt	2/21/2020 12:26 PM	Text Document	1 KB		

Figure 18: Ciphertext results are added to the ciphertext folder with the key used and plaintext file that was encrypted

3.2.0 Encrypting input strings

Users can also encrypt/decrypt input strings for quick visualization of the algorithm in action. This process can be done by pressing '2' in the encryption menu as seen below.

```
Enter selection (q to quit): 2
The active: [17, 23, 391, 352, 65537, 12102, 65]
Enter string to encrypt: But soft, what light from yonder window breaks?

Plain text: But soft, what light from yonder window breaks?
Cipher text: 00tÊs:UJtôÊY0ctÊkHÍ0tÊUJr÷ÛÊã÷η1drÊYHη1÷YÊêrdç0sI'
Decrypted cipher text: But soft, what light from yonder window breaks?
```

Figure 19: Input strings can also be encrypted/decrypted for quick visualization and algorithm testing

4.0.0 Fixed Point Analysis

The RSA Sandbox features several mechanisms for exploring the relative strength and occurrence of fixed points for any septuple/key combination. An overview of the fixed point analysis mechanisms and their functionality can be seen in the following sections.

4.1.0 Active Septuple Analysis and Septuple Comparison

A screenshot of the fixed point analysis menu can be seen below.

```

Enter selection (M/m to reload menu): 7

***** FIXED POINT ANALYSIS MENU *****

1 - Analyze active septuple
2 - Compare all septuples
3 - Generate transparency profile
4 - Compare all septuples and keys
Q - Return to main menu

Enter selection (q to quit):

```

Figure 20: Fixed point analysis allows septuple and key comparison transparency analysis

The user can choose to analyze the strength of the active septuple by choosing option '1'. They can also compare the strength of every septuple loaded in the program with option '2'. Screenshots of these mechanisms in action can be seen below.

```

***** FIXED POINT ANALYSIS MENU *****

1 - Analyze active septuple
2 - Compare all septuples
3 - Generate transparency profile
4 - Compare all septuples and keys
Q - Return to main menu

Enter selection (q to quit): 1
Analyzing active septuple: [17, 23, 391, 352, 65537, 12102, 65]
***** Results *****:
Septuple: [17, 23, 391, 352, 65537, 12102, 65], Holes found: 48, Transparency: 12.31%

```

Figure 21: Option '1' will analyze the active septuple

```

***** RANKED RESULTS *****
Septuple: [5, 17, 85, 64, 17, 13, 49], Holes found: 80, Transparency: 95.24 %
Septuple: [3, 5, 15, 8, 5, 3, 5], Holes found: 10, Transparency: 71.43 %
Septuple: [13, 17, 221, 192, 257, 87, 65], Holes found: 82, Transparency: 37.27 %
Septuple: [5, 7, 35, 24, 17, 12, 17], Holes found: 12, Transparency: 35.29 %
Septuple: [5, 13, 65, 48, 257, 91, 17], Holes found: 22, Transparency: 34.38 %
Septuple: [3, 7, 21, 12, 3, 1, 24], Holes found: 6, Transparency: 30.0 %
Septuple: [11, 17, 187, 160, 17, 12, 113], Holes found: 48, Transparency: 25.81 %
Septuple: [3, 17, 51, 32, 5, 2, 13], Holes found: 12, Transparency: 24.0 %
Septuple: [5, 11, 55, 40, 5, 1, 56], Holes found: 12, Transparency: 22.22 %
Septuple: [17, 41, 697, 640, 65537, 52532, 513], Holes found: 150, Transparency: 21.55 %
Septuple: [3, 11, 33, 20, 5, 1, 36], Holes found: 6, Transparency: 18.75 %
Septuple: [5, 97, 485, 384, 17, 5, 113], Holes found: 82, Transparency: 16.94 %
Septuple: [17, 29, 493, 448, 65537, 37596, 257], Holes found: 82, Transparency: 16.67 %
Septuple: [3, 13, 39, 24, 3, 1, 40], Holes found: 6, Transparency: 15.79 %
Septuple: [5, 29, 145, 112, 17, 5, 33], Holes found: 22, Transparency: 15.28 %
Septuple: [17, 19, 323, 288, 17, 1, 17], Holes found: 48, Transparency: 14.91 %
Septuple: [3, 29, 87, 56, 5, 4, 45], Holes found: 12, Transparency: 13.95 %
Septuple: [7, 13, 91, 72, 17, 4, 17], Holes found: 12, Transparency: 13.33 %
Septuple: [17, 73, 1241, 1152, 65537, 58312, 1025], Holes found: 150, Transparency: 12.1 %
Septuple: [5, 73, 365, 288, 257, 58, 65], Holes found: 42, Transparency: 11.54 %
Septuple: [3, 73, 219, 144, 257, 116, 65], Holes found: 24, Transparency: 11.01 %
Septuple: [3, 19, 57, 36, 257, 207, 29], Holes found: 6, Transparency: 10.71 %
Septuple: [5, 23, 115, 88, 17, 11, 57], Holes found: 12, Transparency: 10.53 %
Septuple: [17, 89, 1513, 1408, 257, 117, 641], Holes found: 150, Transparency: 9.92 %
Septuple: [5, 89, 445, 352, 17, 7, 145], Holes found: 42, Transparency: 9.46 %
Septuple: [3, 89, 267, 176, 65537, 24204, 65], Holes found: 24, Transparency: 9.02 %
Septuple: [11, 97, 1067, 960, 257, 223, 833], Holes found: 96, Transparency: 9.01 %
Septuple: [3, 23, 69, 44, 5, 1, 9], Holes found: 6, Transparency: 8.82 %
Septuple: [11, 13, 143, 120, 257, 242, 113], Holes found: 12, Transparency: 8.45 %
Septuple: [5, 53, 265, 208, 65537, 15439, 49], Holes found: 22, Transparency: 8.33 %

```

Figure 22: Option ‘2’ will compare all septuples and sort by transparency

Performing comparison of all septuples will display the results and also output them to the “Results” folder as mentioned in section 1.0. Septuple comparison results are stored in the subfolder ‘Septuple Comparison’ within the results folder as shown below and have a timestamp in their file title to differentiate them.

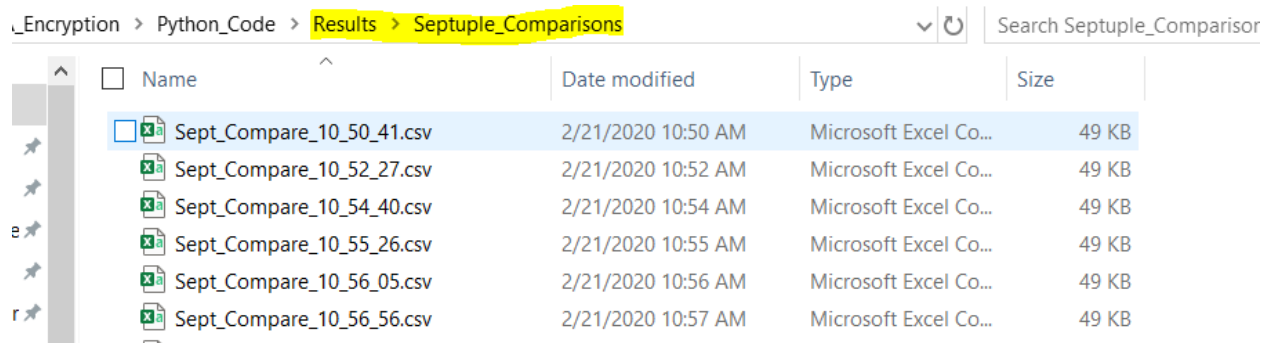


Figure 23: Septuple comparison results are stored in the ./Results/Septuple_Comparisons directory

4.2.0 Key Analysis and Transparency Profile Generation

Option ‘3’ can be used to analyze all keys associated with a given septuple and rank them by strength. This is referred to as a “transparency profile” for a given septuple and key set. An example of the output of this mechanism can be seen below.


```

***** RANKED RESULTS *****
Septuple: [53, 71, 3763, 3640, 71, 56, 2871], Key: 71, Holes found: 210, Transparency: 5.58 %
Septuple: [53, 71, 3763, 3640, 53, 25, 1717], Key: 53, Holes found: 156, Transparency: 4.15 %
Septuple: [53, 71, 3763, 3640, 79, 13, 599], Key: 79, Holes found: 78, Transparency: 2.07 %
Septuple: [53, 71, 3763, 3640, 29, 27, 3389], Key: 29, Holes found: 72, Transparency: 1.91 %
Septuple: [53, 71, 3763, 3640, 41, 32, 2841], Key: 41, Holes found: 52, Transparency: 1.38 %
Septuple: [53, 71, 3763, 3640, 61, 58, 3461], Key: 61, Holes found: 52, Transparency: 1.38 %
Septuple: [53, 71, 3763, 3640, 43, 23, 1947], Key: 43, Holes found: 42, Transparency: 1.12 %
Septuple: [53, 71, 3763, 3640, 11, 1, 331], Key: 11, Holes found: 30, Transparency: 0.8 %
Septuple: [53, 71, 3763, 3640, 31, 19, 2231], Key: 31, Holes found: 30, Transparency: 0.8 %
Septuple: [53, 71, 3763, 3640, 5, 1, 4368], Key: 5, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 13, 1, 3920], Key: 13, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 17, 8, 1713], Key: 17, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 37, 29, 2853], Key: 37, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 73, 22, 1097], Key: 73, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 89, 10, 409], Key: 89, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 97, 19, 713], Key: 97, Holes found: 12, Transparency: 0.32 %
Septuple: [53, 71, 3763, 3640, 3, 2, 2427], Key: 3, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 7, 1, 4160], Key: 7, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 19, 12, 2299], Key: 19, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 23, 19, 3007], Key: 23, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 47, 38, 2943], Key: 47, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 59, 23, 1419], Key: 59, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 67, 3, 163], Key: 67, Holes found: 6, Transparency: 0.16 %
Septuple: [53, 71, 3763, 3640, 83, 7, 307], Key: 83, Holes found: 6, Transparency: 0.16 %

```

Figure 24: Option ‘3’ will compare all keys loaded for a given septuple

Similar to septuple comparisons, key analysis results will be appended to the “Transparency Profiles” directory within the results folder.

ryption > Python_Code > Results > Transparency_Profiles

Name	Date modified	Type	Size
Trans_Profile_11_00_58.csv	2/21/2020 11:00 AM	Microsoft Excel Co...	
Trans_Profile_11_01_22.csv	2/21/2020 11:01 AM	Microsoft Excel Co...	
Trans_Profile_11_32_54.csv	2/15/2020 11:32 AM	Microsoft Excel Co...	
Trans_Profile_11_35_51.csv	2/15/2020 11:35 AM	Microsoft Excel Co...	
Trans_Profile_11_36_20.csv	2/15/2020 11:36 AM	Microsoft Excel Co...	
Trans_Profile_12_43_08.csv	2/18/2020 12:49 PM	Microsoft Excel Co...	
Trans_Profile_12_43_40.csv	2/21/2020 12:43 PM	Microsoft Excel Co...	

Figure 25: Transparency profile results are stored in ./Results/Transparency_Profiles directory

4.3.0 Full septuple and key analysis

Option '4' will perform a fixed point analysis for all septuples and all keys loaded in the program using combinations of the previous mechanisms.

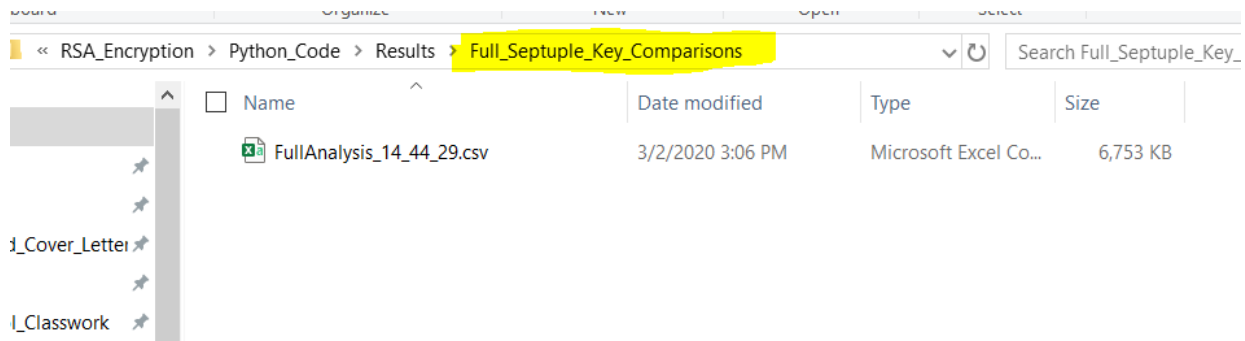


Figure 26: Full septuple and key analysis has unique output folder

5.0.0 The Help Menu

The help menu can be accessed from the main menu by pressing 'h'. Users can select system options for a brief functionality overview.

```
***** HELP MENU *****

1 - Encrypt/Decrypt without padding
2 - Encrypt with padding
3 - Manage Keys
4 - Manage Prime Numbers
5 - Analyze Holes
6 - Output Results
7 - Manage Septuples
8 - Plaintext Message Selection
9 - Specify Ciphertext
10 - Display System Data
11 - Save/Load System data
12 - RSA Sandbox Overview Help
Q - Return to main menu

Enter selection (q to quit): 1
***** Encryption/Decryption without padding Help Overview *****

The encryption/decryption (no padding) option allows a user to encrypt and/or decrypt plaintext
from an input file, as well as encrypt/decrypt an input string without using a padding scheme.
The active septuple is used by default, and the plain text, cipher text, and decrypted cipher text is displayed.
Encrypting plaintext files will also generate a cipher text file and output to the cipher text folder.

For a more detailed overview, please see the RSA Sandbox user manual.
```

Figure 26: The help menu can be used for functionality descriptions while running the Sandbox