Author : Orion Raysher Lee

Admin No. : 190455P

Module Group : IT2554-03

Content:

1. Code structure

2. Navigating through the web application

3. Input accepted for each cryptography techniques and algorithms

Code Structure

1. CryptoPackage contains the python codes to perform all the cryptographic techniques and algorithms and each are accompanied with their own individual test modules and a test module that tests all of them.

Run ‘\_test\_all.py’ to test all modules

Shift cipher: Run ‘shift\_cipher\_test.py’ to test ‘shift\_cipher.py’ code

Mono-alphabet cipher: Run ‘monoalpha\_cipher\_test.py’ to test ‘monoalpha\_cipher.py’ code

Rail fence technique: Run ‘railfence\_cipher\_test.py’ to test ‘railfence\_cipher.py’ code

Simple columnar transposition technique: Run ‘column\_cipher\_test.py’ to test ‘column\_cipher.py’ code

Vernam cipher: Run ‘vernam\_cipher\_test.py’ to test ‘vernam\_cipher.py’ code

Diffie-Hellman key exchange: Run ‘dhkey\_exchange\_test.py’ to test ‘dhkey\_exchange.py’ code

AES: Run ‘aes\_test.py’ to test ‘aes.py’ code

2. static contains all the images and CSS used

3. templates contains all the HTML codes

4. cryptoapp.py contains the python codes which involves the routing of different HTML pages and backend processing of given inputs using the appropriate requested cryptography techniques and algorithms.

Navigating around the web application

1. Home page (When starting the app) tells you the contents of the web application

2. Learn tab will teach you the following things :

* Need for security
* Trusted Systems and reference monitor
* Security models
* Security management practices
* Types of attacks

*Note: There is also a quiz to test your knowledge after learning*

3. Cryptography Techniques tab will let you use different techniques and learn how they work :

* Shift cipher
* Mono-alphabet cipher
* Rail fence technique
* Simple columnar transposition technique
* Vernam cipher
* Diffie-Hellman key exchange

4. Symmetric Algorithms tab will let you use different algorithms and learn how they work :

* Electronic Code Book (ECB)
* Cipher Block Chaining (CBC)
* Cipher Feedback (CFB)
* Output Feedback (OFB)

Input accepted for each cryptography techniques and algorithms

Shift cipher

Characters field represent the set of characters that it will use for encryption and decryption.

Alphabets contains “ABCDEFGHIJKLMNOPQRSTUVWXYZ”

Base64 characters contains “ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=”

Input: Key must be an integer

Input: Plaintext/Ciphertext cannot be empty

*Note: Characters in plaintext/ciphertext that are not found in the set of characters used for encryption and decryption will be ignored (meaning that character will stay the same after encryption/decryption)*

Mono-alphabet cipher

Input: Key must contain one of every character in the alphabets (Can be left empty to generate a random key, generated key will be shown after submitting)

Input: Plaintext/Ciphertext cannot be empty

Rail fence technique

Input: Key must be more than or equal to 2, must be an integer (Because 1 row would not affect the input value)

Input: Plaintext/Ciphertext cannot be empty

*Note: Output for encryption might contain spaces at the end which DOES MATTER if you are trying to decrypt it, ensure that you copy the entire thing including space in the back and in the front*

Simple columnar transposition technique

Input: Columns must be more than or equal to 2, must be an integer (Because 1 column would not affect the input value)

Input: Key, number of characters in key must be the same as number of columns as each character represents 1 column (E.g. If column = 5, key needs to have 5 characters like “Hello” or “Great”)

Input: Plaintext/Ciphertext cannot be empty

*Note: Output for encryption might contain spaces at the end which DOES MATTER if you are trying to decrypt it, ensure that you copy the entire thing including space in the back and in the front*

Vernam cipher

Input: Key, must have the same number of characters as plaintext/ciphertext (E.g. if plaintext/ciphertext is “Hello”, key must have 5 characters like “World”, “Power”)

Input: Plaintext/ciphertext must not be empty, must have the same number of characters in key

*Note: Non-alphabetic characters will be ignored, meaning they will stay the same after encryption/decryption*

Diffie-Hellman key exchange

Input: n, must be a prime number

Input: g, must be a prime number

Input x, must be 0 or more, must be an integer

Input y, must be 0 or more, must be an integer

AES

For encryption:

Input: Plaintext/Ciphertext must not be empty

*Note: Key is randomly generated and IV (only for CBC, CFB and OFB) is randomly generated and will be shown upon submitting. Key size is specified by user.*

For decryption:

Input: Key must not be empty

Input: IV must not be empty (field shows up only for CBC, CFB, OFB)

Input: Plaintext/Ciphertext must not be empty

*Note: In case if either the key or IV contains* ***both*** *single quote and double quote characters, it might and probably not work (not tested). What I have managed to fix is if it contains one of those two characters.*

Refer to CryptoPackage 🡪 AES.py 🡪 function called convert\_to\_bin(string)