CSC650 SECURE NETWORKED SYSTEMS

Project 1

Due: 7:00PM, Tuesday, 03/08/2016

1. Goal

The goal of this project is to write a simple encryption algorithm library to implement DES, ECB, and CBC. The programming language is Java.

2. Instruction

You should write an encryption library with the following API:

```
public class Crypto
```

```
int[] DES(int[] plaintext, int[] key)
int[] ECB(String plaintext, String key)
int[] CBC(String plaintext, String key, String IV)
```

(1) Implement the DES encryption algorithm. The signature of the DES encryption function should be similar to

```
int[] DES(int[] plaintext, int[] key)
```

Here, the input variables `plaintext` and `key` are both integer arrays of size 64. Each element in these two arrays is either 0 or 1. Throw away an error message if the size of the arrays is not 64. The output of the function is the ciphertext, which is also an integer array of size 64.

(2) Implement the ECB encryption algorithm. The signature of the ECB encryption function should be similar to

```
int[] ECB(String plaintext, String key)
```

Here, the input variables `plaintext` and `key` are both strings. You need to first covert them into two integer arrays, where each element is either 0 or 1, by transforming each letter in the strings into its 8-bit ASCII code.
If the size of the integer array obtained from the string variable `key` is less than 64, throw away an error message; otherwise, we extract the subarray consisting of the first 64 elements as the key to DES encryption algorithm.
When you breaks the integer array obtained from the string variable `plaintext` into 64-bit blocks, pad 0 to the last block if its size is less than 64.
After you encrypt each block using DES, concatenate the outputs into one integer array. Then, group every 8 elements in the array as one ASCII code and convert it into its decimal format. Hence, the final output of the ECB encryption function is an array of decimal numbers.

(3) Implement the CBC encryption algorithm. The signature of the CBC encryption function should be similar to

```
int[] CBC(String plaintext, String key, String IV)
```

Here, there are three input variables: `plaintext`, `key`, and `IV`, all of which are strings. You need to first covert them into two integer arrays, where each element is either 0 or 1, by transforming each letter in the strings into its 8-bit ASCII code.
If the size of the integer array obtained from the string variable `key` is less than 64, throw away an error message; otherwise, we extract the subarray consisting of the first 64 elements as the key to DES encryption algorithm.
If the size of the integer array obtained from the string variable `IV` is less than 64, throw away an error message; otherwise, we extract the subarray consisting of the first 64 elements as the Initialization Vector to CBC encryption algorithm.
When you break the integer array obtained from the string variable `plaintext` into 64-bit blocks, pad 0 to the last block if its size is less than 64.
After you encrypt each block using DES, concatenate the outputs into one integer array. Then, group every 8 elements in the array as one ASCII code and convert it into its decimal format. Hence, the final output of the CBC encryption function is an array of decimal numbers.

3. **Test Data**

(1) DES

Input:

```
int[] plaintext = { 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1,
0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1};
int[] key = { 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0,
0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0,
1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1,
1, 1, 1, 1, 1, 0, 0, 0, 1 };
```

Output:

```
Int[] ciphertext = { 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1,
0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0,
0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1}
```

(2) ECB

Input:

```
String plaintext = “I LOVE SECURITY”;
String key = “ABCDEFGH”;
Output:
Int[] ciphertext = {198, 252, 213, 112, 106, 165, 23, 145,
29, 52, 125, 61, 85, 217, 102, 155};
```

Input:

```
String plaintext = “GO GATORS!”
String key = “ABCDEFGH”
Output:
Int[] ciphertext = {86, 100, 180, 248, 126, 142, 38, 5, 255,
224, 149, 93, 149, 189, 237, 2};
```

(3) CBC

Input:

```
String plaintext = “I LOVE SECURITY”;
String key = “ABCDEFGH”;
String IV = “ABCDEFGH”;
Output:
Int[] ciphertext = {63, 69, 76, 252, 154, 205, 193, 162, 46,
88, 102, 161, 151, 14, 56, 97};
```

Input:

```
String plaintext = “SECURITYSECURITY”;
```

```
String key = “ABCDEFGH”;
String IV = “ABCDEFGH”;
Output:
Int[] ciphertext = {232, 111, 39, 242, 85, 25, 41, 106, 39,
52, 175, 62, 196, 141, 176, 70};
```

Note: In this case, we can observe that identical blocks in the plaintext are encrypted into different ciphertext blocks with CBC.

4. Submission

You should implement these three encryption algorithms in three methods. You can write your own main function to invoke them and test your code, but you do not need to submit your main function. We will write a main function and test your code by using our test cases.

Submit one source code file consisting all the methods to the regular submission link on iLearn.