Lab #3 - September 20, 2018

Due Dec 31 at 11:59pm **Points** 12 **Questions** 12
Available Sep 20 at 9:20am - Dec 31 at 11:59pm 3 months **Time Limit** None
Allowed Attempts Unlimited

Instructions

[1] Good morning and welcome to the third lab session!

Lab sessions provide the opportunity for recitation and more in-depth understanding of the materials covered in class, as well as preparation for upcoming homework assignments.

Your attendance only of a given lab session (and, thus, your participation in the assignments and/or discussions) gives you full credit. You are expected to stay in the lab for the entire session, or until the TAs release the class possibly earlier than scheduled, and to actively participate in the discussions (e.g., asking questions, answering to questions, etc.).

Typically, lab assignments are offered in the form of an ungraded quiz, which should not be interpreted as a test or a mini exam.

[2] Please download the Java file below and run it as described in the questions.

SymmetricKeyTest.java

[3] Today's quiz is planned so that it reviews some materials covered in class in practice-oriented point of view. As always, this quiz will be available to you for practice or revision at a later time. Thus, if you are not able to complete the quiz during the lab, please take your time to do so at a later time (and if you need any clarification, seek help during TA or the instructor's office hours).

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	3 minutes	12 out of 12
LATEST	Attempt 3	1 minute	4 out of 12
	Attempt 2	3 minutes	12 out of 12
	Attempt 1	42 minutes	8 out of 12

Submitted Oct 4 at 1:56pm

The supplied program takes 5 arguments:

1. Algorithm
   - This a block-cipher algorithm that the program will use. It can take on the following values (in parenthesis the key size or key-size range):

        - DES (56 bit);
        - TripleDES (112 bit);
        - RC2 (8 to 1024 bits, but in increments of 8 bits; default value is 64 bits);
        - AES (128, 192, or 256 bit); or
        - Blowfish (any multiple of 8 bits between 32 and 448 bits).
    - In class we have discussed only DES, TripleDES and AES, but RC2 and Blowfish operate similarly - if needed, you can check their general description in Wikipedia.
2. Key size
    - The length of key that will be produced by the program and be used by the block cipher. Each algorithm accepts and must be used with certain key sizes, as shown above.
3. Mode of operation
    - As discussed in last week's lab and this week's lecture, a block cipher's mode of operation is the method by which the encryption algorithm is applied to messages that are longer than the block size, i.e., to consecutive blocks of the message. It can take on the following values: ECB, CBC, OFB, and **PCBC (https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Propagating_Cipher_Block_Chaining_(PCBC**
    .
    - Note that RC2 only accepts ECB.
4. Iterations
    - The number of times to run the encryption and decryption algorithms of the specified block cipher over the selected message in order to measure their efficiency.
    - We recommend a value above 100000 to get a more better estimate of the measured statistics.
5. Message
    - The message to encrypt.

Here is an example of a successful command-line call of the provided program - please note the required use of the quote marks with your selected message:

java SymmetricKeyTest AES 128 CBC 100000 "This is a test message"

Once you have read and understand this, please select Next.

In this part, you will run the program with different input and take into account the encryption times the program outputs.

When you are ready to move on, please select Next.

For each block-cipher algorithm and with its minimum allowable key size, run the program over the same message, each time using the same mode of operation and same number of iterations. For each algorithm, take note of the encryption and decryption times.

Once you have recorded these times, please select Next.

Using the AES algorithm, experiment with each different mode of operation. Take note of the encryption and decryption times.

Once you have recorded these times, please select Next.

Using the RC2 algorithm on ECB mode, experiment with different key sizes. Try at least 3 different key lengths. Take note of the encryption and decryption times.

Once you have recorded these times, please select Next.

Using the AES algorithm, experiment with the message length. Try at least 3 different message lengths. Take note of the encryption and decryption times.

Once you have recorded these times, please select Next.

Based on the results of your experiments, answer the following questions. Press Next to start.

Question 1

1 / 1 pts

What happens if you try to use a key size greater than 128 bits in any of the algorithms?

Do some research as to the reason for the apparent inconsistency between the algorithm definitions and the Java Standard Library implementations.

- ArrayIndexOutOfBoundsException
- The program runs as expected, using only 128 bits.

Correct!

- (•) InvalidKeyException

Yes, Java has a built-in cap on key sizes to allow for international use. Take a look at the documentation on the JCE (Java Cryptography Extension), which removes this cap.

Question 2

0 / 1 pts

Why does the RC2 implementation require a 40-bit key minimum, despite the standard specifying only an 8-bit minimum? Intuitively, what is the problem with an 8-bit key (hint: how many bits make up an ASCII character)?

You Answered

- (•) Using 8-bit keys would be too slow in practice.

The reason is security related, not performance related.

Correct Answer

- ( )

An 8-bit key is insecure. In ECB mode, if using an 8-bit key and 8-bit characters, it can be subject to a frequency analysis attack.

Question 3

0 / 1 pts

Which algorithm was the fastest?

You Answered

- (•) DES

Correct Answer

- ( ) AES
- ( ) Blowfish

Question 4

0 / 1 pts

Which Algorithm was the slowest?

- ( ) RC2

You Answered

- (•) TripleDES

Correct Answer

- ( ) Blowfish

Question 5

0 / 1 pts

What is a scenario in which a slower algorithm may be more preferable to use?

Correct Answer

- ( ) The block cipher is used to conceal passwords.
- ( ) A slower algorithm is never more preferable to use.

You Answered

- (•) A slower algorithm is better when you are working with larger messages as it can be more accurate.

This is incorrect. The accuracy should be perfect with all algorithms (e.g., encrypting a message and decrypting its ciphertext should always result in the same message).

Question 6

0 / 1 pts

Which mode of operation was the fastest?

- OFB
- You Answered: CBC
- Correct Answer: ECB

Question 7

0 / 1 pts

What is the benefit to using one of the chaining modes as opposed to ECB?

- Chaining modes use less bandwidth.
- Correct Answer: Chaining modes increase confusion and diffusion. These modes have more randomness in ciphertext.
- You Answered: Chaining modes are faster.

Chaining modes are actually slower.

Question 8

1 / 1 pts

How does PCBC differ from CBC?

- PCBC is optimized for parallel programming while CBC is not.
- PCBC can only be used with AES while CBC can be used with other algorithms.
- Correct!: PCBC causes both ciphertext and plaintext changes to propagate over the output, whereas CBC only propagates plaintext changes.

CBC uses the ciphertext of the previous block and xors it with the next plaintext block to feed the block cipher. PCBC actually utilizes the previous message block and the previous ciphertext block when preparing the next message block for the block cipher.

In PCBC mode, each block of plaintext is XORed with both the previous plaintext block and the previous ciphertext block before being encrypted. It was designed to cause small changes in the ciphertext to propagate indefinitely when decrypting, as well as when encrypting.

Question 9

1 / 1 pts

One of these algorithms is the accepted standard in the US. Which one? Why?

- DES because it has been around the longest and is, therefore, the most trusted.
- TripleDES, because it was made to replace DES.

Correct!

- AES, because it takes a wider range of key sizes and handles modern brute-force attacks better.

AES. It was designed as a replacement to TripleDES, which was itself designed to replace DES. These replacements happen as hardware improves, reducing the amount of time necessary to perform brute-force attacks. AES also has a wider range of potential key sizes, which future-proofs it for longer than its predecessors, which could only take 56 (DES), 112, or 256 (TripleDES) bits.

Question 10

1 / 1 pts

How did key length impact processing time assuming it is proportional to the message length? What about security?

Having a larger key substantially increases processing time, but also increases security.

Having a larger key decreases both processing time and security.

Correct!

Larger key size minimally raises processing time on average, but improves security significantly.

Question 11

0 / 1 pts

Did changing the message length impact the processing time?

For instance, assume instead that the application was multi-threaded and the ECB mode of operation is used. What impact would this optimization have on longer messages?

Larger messages take longer to encrypt, but using ECB with multi-threading will not speed up the process.

You Answered

Message size makes no difference in the processing time, but ECB is faster when used with multi-threading.

Message size makes encryption take longer. (But, indeed, ECB can be used with multi-threading to speed up processing.)

Correct Answer

Larger messages take longer to encrypt and decrypt. When multi-threading is used over ECB, the message would be encrypted faster.

Question 12

0 / 1 pts

We have hard-coded the PKCS5Padding algorithm into the program. What does this algorithm accomplish?

You Answered

- (•) This algorithm removes invalid characters from the message and replaces them with valid ASCII characters.
- ( ) This algorithm performs applies additional one-time pad masking of the message using bit-wise XOR, prior to the application of the block cipher.

Correct Answer

- ( ) The algorithm pads the input message to an acceptable block size for the associated cipher.