Homework 2

- Honor Code: You must work completely independently on this assignment. Do not discuss the questions or answers with each other before the assignment is due. Any breach of the honor code will be handled per the University's policy on academic honesty.
- Follow the instructions very careful. Answers that do not conform to the instructions will not be given credit.
- Understand all the code given to you in this lab. Search for documentation online if there is a primitive or API you have not encountered before. You are not responsible for understanding the underlying mathematics behind the cryptographic primitives. However, you are responsible for using these primitives in an application in a secure manner.
- Use Java 8.
- Only use the external Java libraries provided to you in the lab.
- Submit your solutions through Blackboard. Do not submit as a zip file. Submit a PDF of the text answers, and upload separately the two files GenerateScroogeKeyPair.java and GenerateDigitalSignature.java.

In this lab, you will implement preliminary parts of the ScroogeCoin cryptocurrency. In the next lab, you will act as Scrooge and verify transactions sent to you by users and add them to your blockchain.

1. Download and install the Java Unlimited Strength Jurisdiction files from http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html. These files remove any restrictions on cryptographic strengths. Read the README file to understand why this step is needed and how to carry out the installation. If you don't install these files, you will get an "Illegal key size" exception when trying to generate keys.

2. In this lab and future labs, you must use cryptographically secure random number generation where necessary. Read the following link on generating cryptographically secure random values in Java:

https://www.cigital.com/blog/proper-use-of-javas-securerandom/

In your submission, answer the following questions:

A. Can the Java SHA1PRNG be used securely for cryptographic operations such as generate private/public key pairs?

B. What pitfalls do programmers have be aware of when using pseudo-random number generators for cryptographic operations?

C. Why should a programmer be concerned about using `SecureRandom.getInstanceStrong()` in certain types of applications?

3. In ScroogeCoin, the central authority Scrooge receives transactions from users. Scrooge signs all hash pointers in the ScroogeCoin blockchain. To generate signatures, you will need to generate a private/public key pair on your computer that you can use to digitally sign transactions.

Bouncycastle (https://www.bouncycastle.org/) is a popular Java crypto library used in real world crypto systems. The lab includes a lib directory that contains the jars for this library.

Read and thoroughly understand the CryptoReference2.java file which uses crypto primitives like what Bitcoin uses. Try running the CryptoReference2 program on your computer and confirm that it completes successful without throwing exceptions. This program generates ECDSA keys. Read more about this type of cryptographic algorithm at https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm.

4. Fill in the GenerateScroogeKeyPair.java main method with code that does the following:

A. Generates a ECDSA key pair for Scrooge.

B. Stores the private key in an encrypted format on disk.

C. Store the public key in a separate, unencrypted file.

Run the class to generate the key pair for Scrooge. Name the key files as scrooge_sk.pem and scrooge_pk.pem so that it is clear who they belong to. You will use this key pair for the remaining parts of this lab.

In your submission, include your code and the contents of the file containing Scrooge's public key. Do not submit your secret key. Remember never to give out your secret key and to always encrypt the secret key file when storing it on disk.

5. Fill in the GenerateDigitalSignature main method with code that does the following:

A. Reads Scrooge's key pair from disk

B. Generate Scrooge's digital signature for the message "Pay 3 bitcoins to Alice". Do not include the quotations in the message. Capitalization matters.

In your submission, include your code and the digital signature in hexadecimal.