****

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[Document Revision History 3](#_Toc33111302)

[Client 3](#_Toc33111303)

[Instructions 3](#_Toc33111304)

[Developer 4](#_Toc33111305)

[1. Algorithm Cipher 4](#_Toc33111306)

[2. Certificate Generation 4](#_Toc33111307)

[3. Deploy Cipher 4](#_Toc33111308)

[4. Secure Communications 4](#_Toc33111309)

[5. Secondary Testing 4](#_Toc33111310)

[6. Functional Testing 5](#_Toc33111311)

[7. Summary 5](#_Toc33111312)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **6/19/2022** | **Hillary Loyd** |  |

## Client



## Instructions

Deliver this completed Practices for Secure Software Report documenting your process for writing secure communications and refactoring code that complies with software security testing protocols.

Respond to the steps outlined below and replace the bracketed text with your findings in your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Hillary Loyd

## 1. Algorithm Cipher

* Provide a brief, high-level overview of the encryption algorithm cipher.
* Discuss the hash functions and bit levels of the cipher.
* Explain the use of random numbers, symmetric vs non-symmetric keys, and so on.
* Describe the history and current state of encryption algorithms.

When considering the best practices for security protection and in defending against various types of security attacks, I found that the best option for protection and coverage would be Advanced Encryption Standard (AES). This algorithm can also be known as the Rjindael algorithm created by Joan Daemen and Vincent Rijmen. It is also referred to as the gold standard for encrypting data and was adopted by the United Stated government in 2001. This algorithm is a 128-bit block cypher which supports the keys of 128. 192 and 256 bits. Trying to recover an AES key would take a very long time as long as it is implemented correctly and there isn’t any known security risk.

The best cipher is the AES. The reasoning for this is that it is resistant to most of the security attacks that we deal with. Also, that brute force is not an option because this would take a very long time to try and force. The reason that we may not choose the most secure cipher is if another system comes in to replace the AES. Another option could be if someone made something to break the AES and does not make it the most secure. Meaning that we must adapt to the changes that are happening as our technology grows.

The purpose of a hash function is to use an algorithm that takes an amount of data input and produces a fixed size output of encrypted text called a has value. This text can then be stored and can be accessed later to verify the user. The encryption process is designed to prevent theft during the transmission process. When using random numbers, symmetric vs non-symmetric keys they are used inside of the encryption algorithm. The differences are that symmetric keys are privately shared between the encrypt and decrypt key. While a non-symmetric are public keys from the receiver. This was in hopes of eliminating the sharing of keys. The risks of ciphers being solved is very limited and this more secure this would be the form of encryption to rely on. While this form of encryption does not meet every need it is well suited for what needs to be accomplished

## 2. Certificate Generation

Generate appropriate self-signed certificates using the Java Keytool, which is used through the command line.

* To demonstrate that the keys were effectively generated, export your certificates (CER file) and submit a screenshot of the CER file below.

A screenshot of a computer

Description automatically generated with medium confidence

## 3. Deploy Cipher

Refactor the code and use security libraries to deploy and implement the encryption algorithm cipher to the software application. Verify this additional functionality with a checksum.

* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.

Graphical user interface

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

I have successfully made the certificate. However the cert that the website is pulling off of is an old certificate that has been deleted from the PC. I have tried everything to even rebuilding the programing multiple times. However, the certificate on the left does not exist on this PC and I have tried everything to change it. I have tried to look for it and no where has it been found.

## 4. Secure Communications

Refactor the code to convert HTTP to the HTTPS protocol. Compile and run the refactored code to verify secure communication by typing **https://localhost:8443/hash** in a new browser window to demonstrate that the secure communication works successfully.

* Insert a screenshot below of the web browser that shows a secure webpage.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Like stated before I could not get the local host to be secure as this is pulling from a cert that is no longer on this PC. Attached is a picture of the MMC that shows that the certificate on the right is the only localhost certificate on this PC.

## 5. Secondary Testing

Complete a secondary static testing of the refactored code using the dependency check tool to ensure code complies with software security enhancements. You only need to focus on the code you have added as part of the refactoring. Complete the dependency check and review the output to ensure you did not introduce additional security vulnerabilities.

* Include the following below:
  + A screenshot of the refactored code executed without errors
  + Text

    Description automatically generated
  + A screenshot of the dependency check report

A screenshot of a computer

Description automatically generated

## 6. Functional Testing

Identify syntactical, logical, and security vulnerabilities for the software application by manually reviewing code.

* Complete this functional testing and include a screenshot below of the refactored code executed without errors.

Text

Description automatically generated

## 7. Summary

Discuss how the code has been refactored and how it complies with security testing protocols. Be sure to address the following:

* Refer to the Vulnerability Assessment Process Flow Diagram and highlight the areas of security that you addressed by refactoring the code.
* Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.
* Point out best practices for maintaining the current security of the software application to your customer.

This code has been refactored to enable a more secure application through https instead of http. In the application.properties file I added the information that was needed to be able to make this happen. This added an extra layer of protection to the application. While it was not shown in the steps above, I do believe that had the ghost certificate was not in the way that we would have been able to show this. While this does add security this does not keep all the hackers away. It is always important to keep up will all security updates as technology advances to become more proficient.