****

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[\_Toc33111301](#_Toc33111301)

[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** | **6/26/2022** | **Dale Ayers** |  |

## 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

[insert name here]

## 1. Algorithm Cipher

Artemis Financial needs checksums for their files, and there is no better hashing function for checksums than SHA-256. SHA-256 is the current industry standard because, unlike its counterparts MD5 and SHA-1 etc., SHA-256 has never had any reports of any collision. Collisions are improbable when using SHA-256; the numerical value of a collision happening by chance is 2^289.

In SHA-256, messages up to 2⁶⁴ bit (2.3 exabytes, or 2.3 billion gigabytes) are transformed into digests of size 256 bits (32 bytes). This means that whatever we put into SHA-256, the hash function will produce a chunk of data the size a 32-bit string.

Symmetric algorithms have a simple process that is significantly faster and requires less computational power than asymmetric. Symmetric Algorithms are great for communication in 1 to 1 and small group scenarios (Thakkar, 2021). If we have a large chunk of data that needs to be encrypted, Symmetric algorithms like AES 128, 192, or 256 are a great choice. Asymmetric encryption encompasses two distinct encryption keys that are mathematically related. One is called a private key, and the other a public key. The public key is available to anyone who asks for it (it's public) and is used to encrypt the data. The private key is held securely by the individual who is to receive the information and decrypts the data. This prevents man-in-the-middle attacks because of the need for the private key (Thakkar, 2021). Asymmetric encryption is an excellent strategy for servers that connect with a high volume of clients because they would only need to protect and manage one key. Another advantage is creating an encrypted connection without having to meet offline to exchange keys first. Asymmetric encryption also provides authentication because the data is encrypted by a public key that can only be decrypted by a private key that is related to it mathematically; we know the person receiving the information is the person we want to receive the information (Thakkar, 2021). For Asymmetric algorithms, RSA is the most widely used and secure algorithm.

Asymmetric encryption encompasses two distinct encryption keys that are mathematically related. One is called a private key, and the other a public key. The public key is available to anyone who asks for it (it's public) and is used to encrypt the data. The private key is held securely by the individual who is to receive the information and decrypts the data. This prevents man-in-the-middle attacks because of the need for the private key (Thakkar, 2021). Asymmetric encryption is an excellent strategy for servers that connect with a high volume of clients because they would only need to protect and manage one key. Another advantage is creating an encrypted connection without having to meet offline to exchange keys first. Asymmetric encryption also provides authentication because the data is encrypted by a public key that can only be decrypted by a private key that is related to it mathematically; we know the person receiving the information is the person we want to receive the information (Thakkar, 2021). For Asymmetric algorithms, RSA is the most widely used and secure algorithm.

The history of cryptography goes all the way back to 1900 BCE in ancient Egypt; more recently, though, SHA-256 was developed by the United States National Security agency in 2001 and has become the industry standard ever since. SHA-256 is used widely from blockchain transaction verification to generating checksums for file downloads SHA-256 has many different applications.

## 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.

Text

Description automatically generated

## 3. Deploy Cipher

Graphical user interface, text, application, Word

Description automatically generated

## 4. Secure Communications

## Graphical user interface, text, application, Word Description automatically generated

## 5. Secondary Testing

## I cant get a picture of the code executed without error because I deleted the console and the port is taken up. I hope this is proof its working though.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, email

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.

Graphical user interface, text, application

Description automatically generated

## 7. Summary

To secure our code we used three principles, Input Validation by creating a checksum, cryptography by using the SHA-256 algorithm to create a hash function and encapsulation by making information inside the serverController class private. Each layer of protection adds just a little bit more to your code to make it more secure. If we were to focus on just input validation for example, we would could not say that our code was secure because there are so many other levels of security that need to be addressed. Because the cyber security threats are constantly evolving the best practice for maintaining the current security is to work with a security expert and make sure that the security standards are up to date.

References

Apograf Team. (2018). Symmetric cryptography and key management: Considerations on key exhaustion, rotation and security models. Retrieved from <https://www.cryptomathic.com/news-events/blog/symmetric-cryptography-and-key-management-considerations-on-key-exhaustion-rotation-and-security-models>

Manico, J., & Detlefsen, A. (2015). Iron-clad Java. Retrieved from <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch01.html#ch01lev1sec23>

Thakkar, J. (2021, March 11). Types of encryption: 5 encryption algorithms & how to choose the right one. Retrieved from

<https://www.thesslstore.com/blog/types-of-encryption-encryption-algorithms-how-to-choose-the-right-one/>

Tobias, E. (2021, February 15). 128 or 256 bit Encryption: Which Should I Use?. Retrieved from <https://agilemodeling.com/essays/changeManagement.htm>