****

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[CS 305 Project Two 1](#_Toc111229007)

[Document Revision History 3](#_Toc111229008)

[Client 3](#_Toc111229009)

[Developer 4](#_Toc111229010)

[1. Algorithm Cipher 4](#_Toc111229011)

[2. Certificate Generation 4](#_Toc111229012)

[3. Deploy Cipher 4](#_Toc111229013)

[4. Secure Communications 4](#_Toc111229014)

[5. Secondary Testing 5](#_Toc111229015)

[6. Functional Testing 6](#_Toc111229016)

[7. Summary 7](#_Toc111229017)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **August 14 2022** | **Apurva Shukla** |  |

## Client



## Developer

Apurva Shukla

## 1. Algorithm Cipher

The algorithm cipher that was recommended to be used for the client for their respective purposes is SHA-256 with RSA encryption. This algorithm is widely used by almost everyone in the world, including the Department of Homeland Security of the United States (*Compliance with cybersecurity and privacy laws and regulations*, 2021). It is also considered to be one of the most secure encryption algorithms being used, since it is not easily decrypted by current technology. The hash function converts an arbitrary amount of data into a fixed length, so the data is always reliably transferred. The bit level recommended for use with the client is 256-bit and that is applied in the program as well. The use of symmetric encryption is useful for transferring smaller, less resource-intensive data, since it is faster and

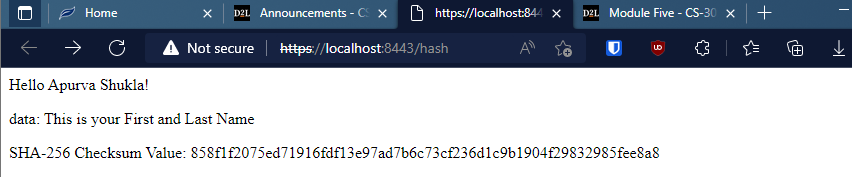
only requires the use of a single key to decrypt. Asymmetric encryption requires the use of two keys to encrypt the data (*Symmetric vs asymmetric encryption: A guide for non-techies*, 2021). The randomly generated bits are created to ensure that the bits are completely unpredictable, and therefore extremely impractical to guess using today’s standards. This technique of encryption has been in use since 2001 and is yet to be broken in security, hence remains the standard for encryption. These reasons are why this specific algorithmic cipher was chosen, as this matches the security requirement of the client with ease and reliability.

## 2. Certificate Generation

Text

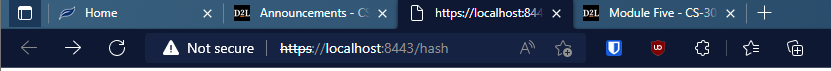
Description automatically generated

## 3. Deploy Cipher



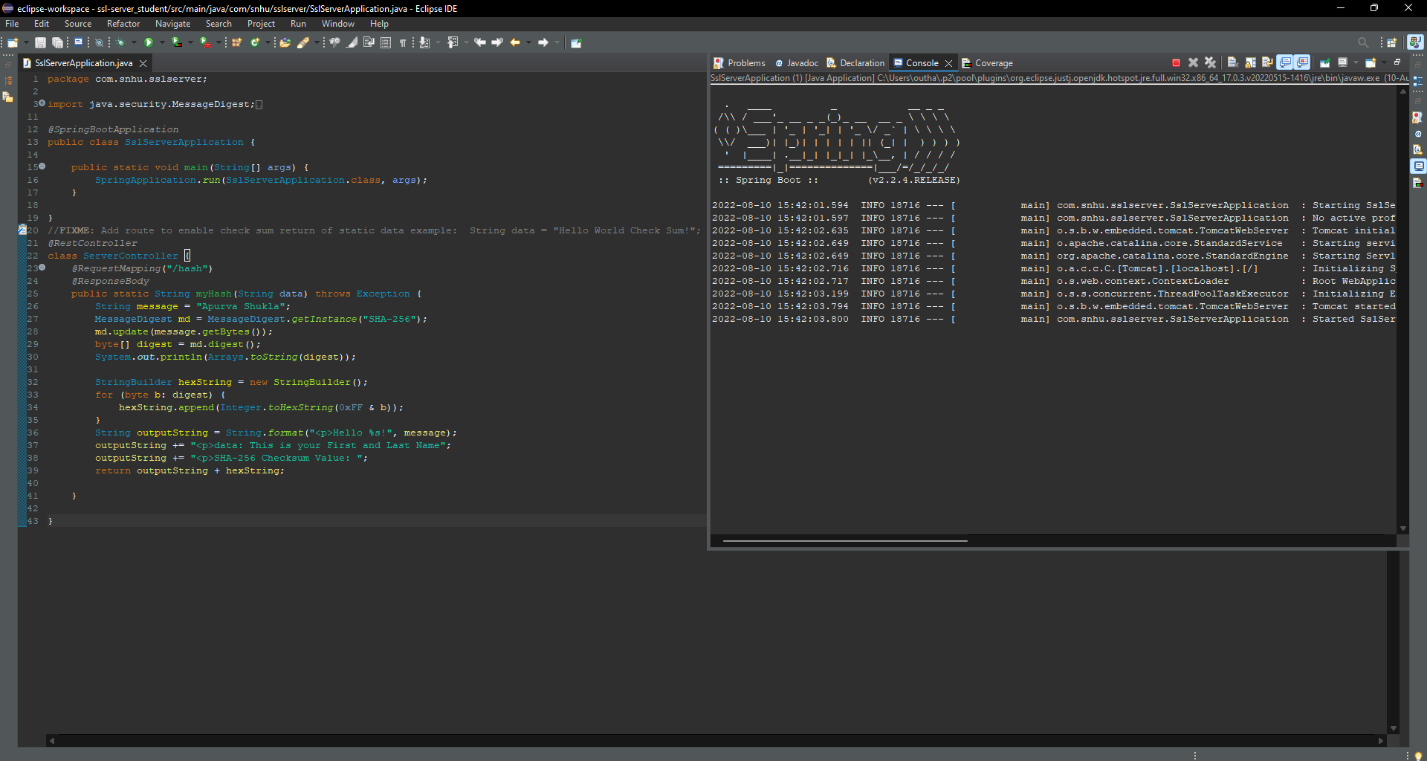
## 4. Secure Communications

The certificate is flagged only because the certificate was not installed in the Trusted Certificate Store on the browser and shows up as Not Secure, since the HTTP request of the website fails to process.



## 5. Secondary Testing

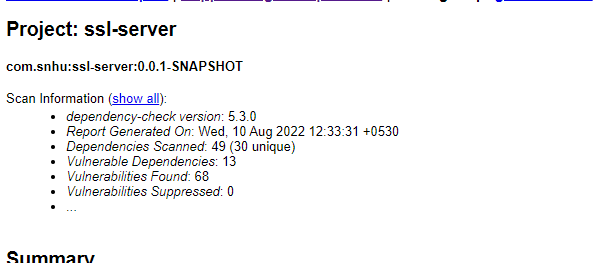
Refactored code executed without errors



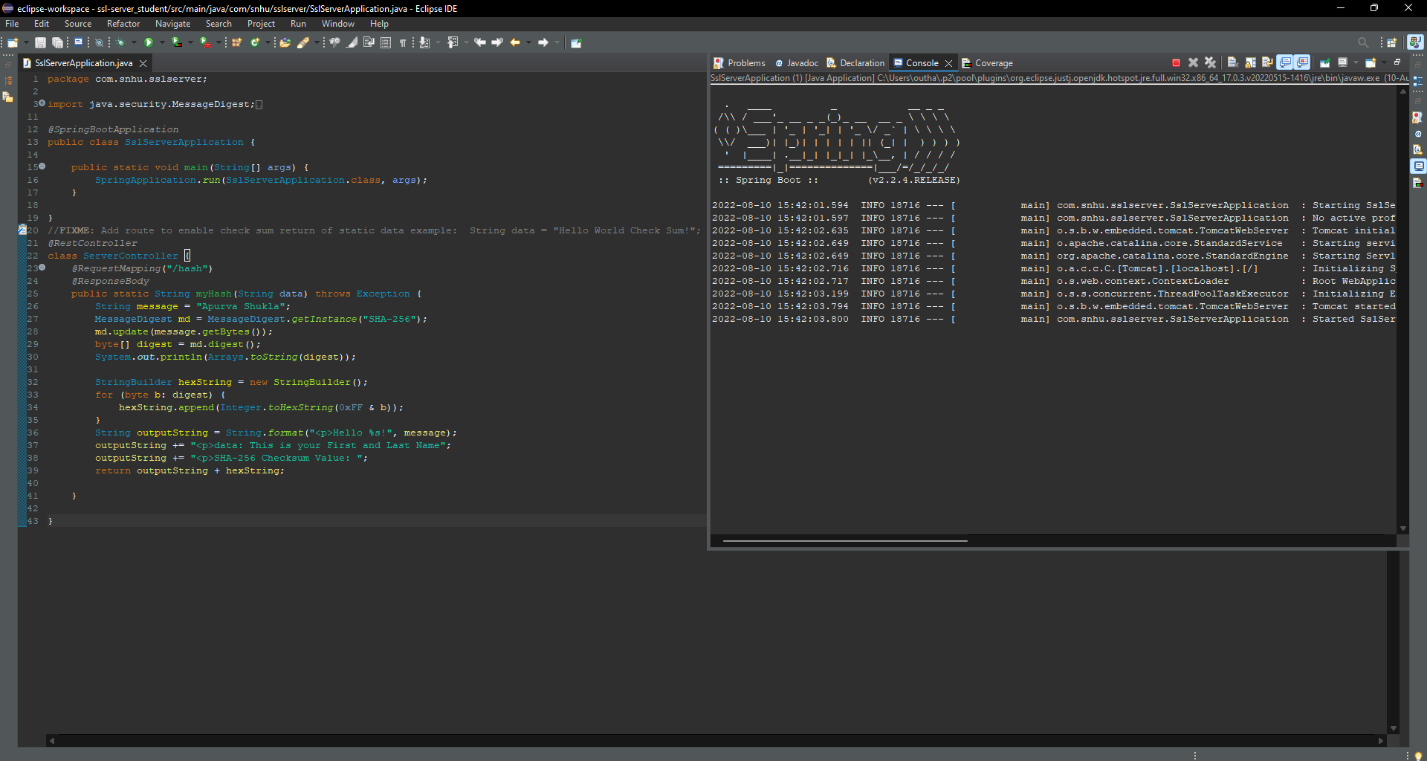
Dependency Check report after refactoringText

Description automatically generated

Dependency check report before refactoring



## 6. Functional Testing



There are not many obvious syntactical vulnerabilities in the code, since the application relies on mostly templated code and uses established APIs that are tested to be secure to deliver the required checksum value. There are some logical vulnerabilities, where the code is hardcoded into the application, which makes it a little difficult to optimize and clean which can lead to wrong usage of the code. There are security vulnerabilities as detected by the dependency check report generated, most of which can be solved by updating to the versions that do not have any currently identified vulnerabilities and have fixed their existing vulnerabilities, the rest can be covered by following industry standard coding practices or finding an alternative package with non-detected vulnerabilities for the packages that can be replaced. The last option should however be always used as a last resort.

## 7. Summary

The areas of security that are covered from the refactor of code is that APis are ensured to be secure and that all interactions are going through smoothly. The code was also ensured to be using encryption to protect and transfer data, while also ensuring that vulnerabilities are not exposed in the code. There is also use of Client/Server architecture to ensure delivery of the data in a secure and faster way. Code Errors and Code Quality were ensured to make sure that they were matching industry standards and followed secure coding practices. The layers of security added to the application ensures that the amount of sensitive data being handled by the application is all transmitted and stored securely, since any leak of this sensitive data will be catastrophic if lost, stolen, or deciphered. The application was also ensured to only transfer data if it was passed on by a secure and encrypted protocol and uses a cipher that is not very easily broken into by hackers. To upkeep and improve the level of security, the developer should update the algorithm cipher used as a newer, more-refined method of encryption is revealed to be reliable and secure enough to be changed.

**References**

*Compliance with cybersecurity and privacy laws and regulations*. NIST. (2021, August 4). Retrieved July 23, 2022, from <https://www.nist.gov/mep/cybersecurity-resources-manufacturers/compliance-cybersecurity-and-privacy-laws-and-regulations>

*Symmetric vs asymmetric encryption: A guide for non-techies*. HackerNoon. (2021). Retrieved August 10, 2022, from https://hackernoon.com/symmetric-vs-asymmetric-encryption-a-guide-for-non-techies-p03c316t