****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Instructions 3](#_Toc102040756)

[Developer 4](#_Toc102040757)

[1. Algorithm Cipher 4](#_Toc102040758)

[2. Certificate Generation 4](#_Toc102040759)

[3. Deploy Cipher 4](#_Toc102040760)

[4. Secure Communications 4](#_Toc102040761)

[5. Secondary Testing 4](#_Toc102040762)

[6. Functional Testing 4](#_Toc102040763)

[7. Summary 4](#_Toc102040764)

[8. Industry Standard Best Practices 4](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **6/18/2023** | **Adela Martinez** |  |

## Client



## Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

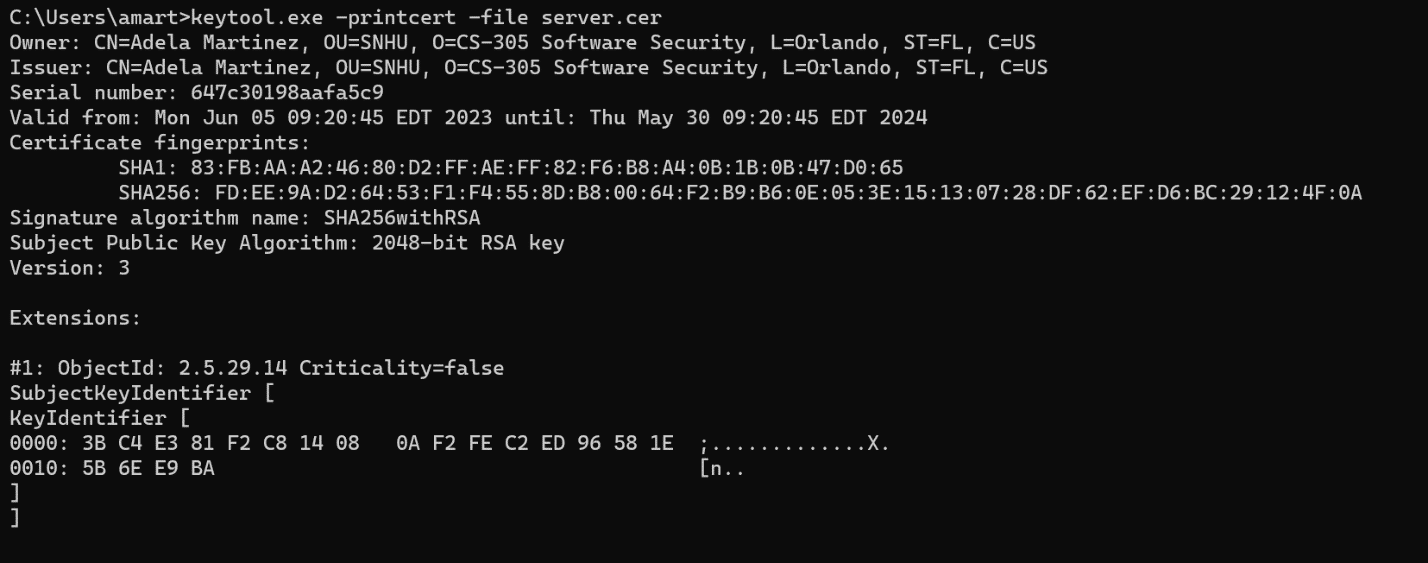
Adela Martinez

## Algorithm Cipher

The encryption algorithm cipher I would recommend for this project would be SHA256(Secure Hash Algorithm). SHA256 is a cryptographic hash function that takes an input and converts it into an output that is 256 bits hash value. SHA256 was a newer version of SHA1, that was originally 160 bits, with a larger number of bits it made it more secure.

## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated with low confidence

## Secure Communications

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

A screenshot of a computer

Description automatically generated with low confidence

## Secondary Testing

Insert screenshots below of the refactored code executed without errors and the dependency-check report.

A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with low confidence

## Functional Testing

Insert a screenshot below of the refactored code executed without errors.

A screenshot of a computer program

Description automatically generated with low confidenceA close-up of a computer code

Description automatically generated with low confidence

## Summary

To add a route to enable the check sum I added a rest controller to serve as the server control for the programs hash stop. I used the SHA-256 hashing cipher since there’s most likely a small chance of collision and it is very secure.

## Industry Standard Best Practices

To keep up with maintenance of the security of the software I would suggest performing dependency checks every few months to make sure no new vulnerabilities pop up. It will also help identify if anything needs to be updated in the pom.xml.