****

# 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** | **12/6/23** | **Curtis Felsher** |  |

## 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

Curtis Felsher

## Algorithm Cipher

1. Provide a brief, high-level overview of the encryption algorithm cipher.

AES encrypts data in blocks using symmetric keys. AES is known for its strength and speed which makes it versatile for software and hardware encryption. Since the algorithm is symmetric, the same key is used for encryption and decryption.

1. Discuss the hash functions and bit levels of the cipher.

The different key sizes determine the number of rounds of transformation that the data will undergo. A higher bit count equals more rounds which increases security at the slight expense of speed. 128 bit = 10 rounds 192 bits = 12 rounds 256 bit = 14 rounds

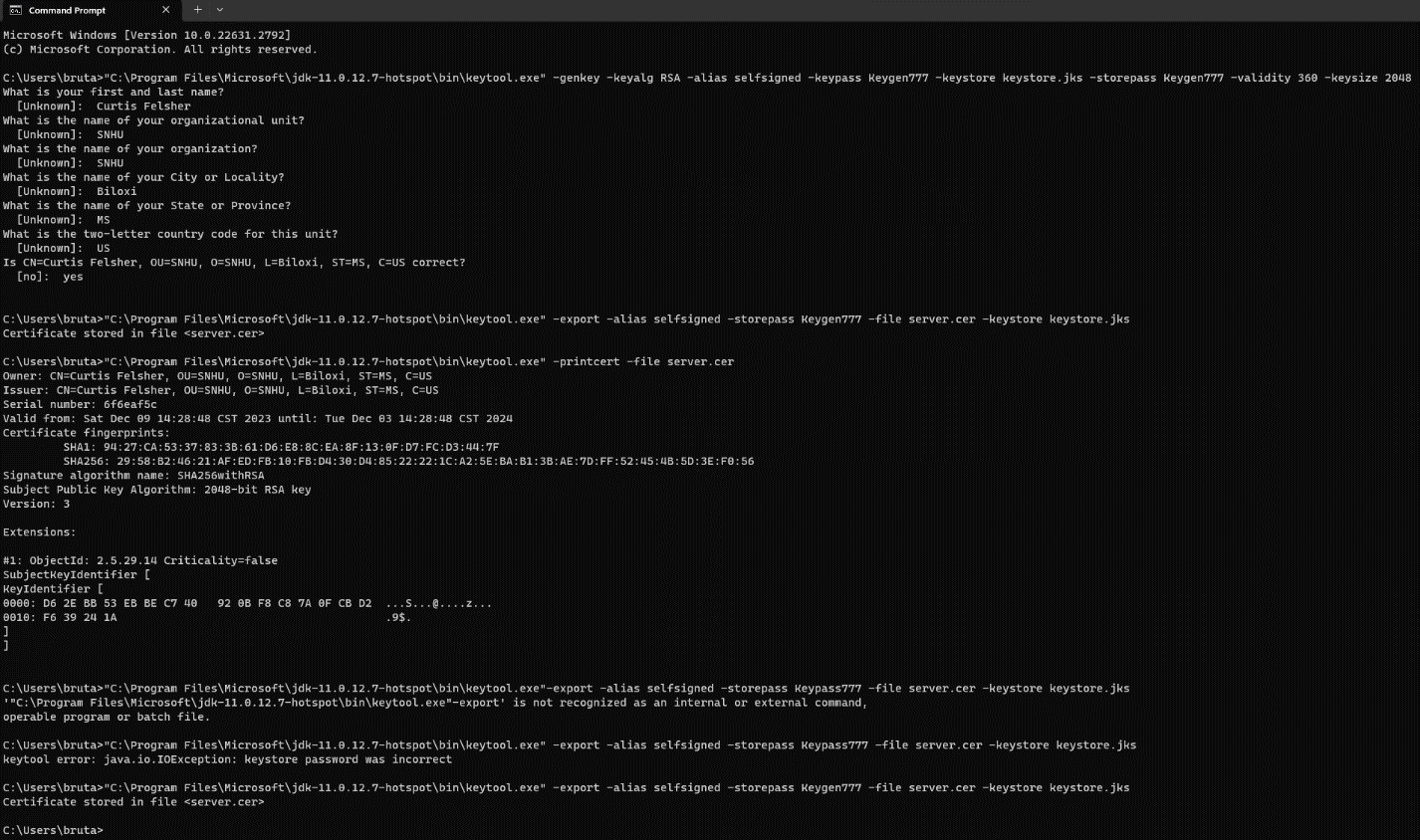
1. Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.

Key generation and selection are critical for AES security. Random numbers are a key component for generating a strong key that is unpredictable. Symmetric keys use the same key for encryption and decryption while asymmetric keys use a different key for each. Since this implementation is symmetric, key distribution and management are areas of concern.

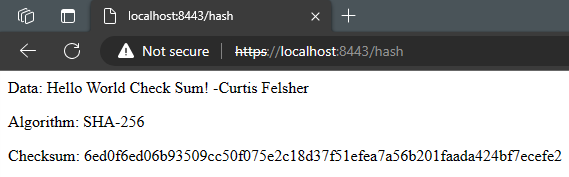
1. Describe the history and current state of encryption algorithms.

AES is the successor to DES and is designed to be secure against practical and theoretical attacks. AES is a great choice for this application as its efficiency in different platforms alongside its adaptability makes it a prime candidate for Artemis Financial. Its proven track record in resisting attacks adds a layer of trust to any financial data that Artemis Financial may handle.

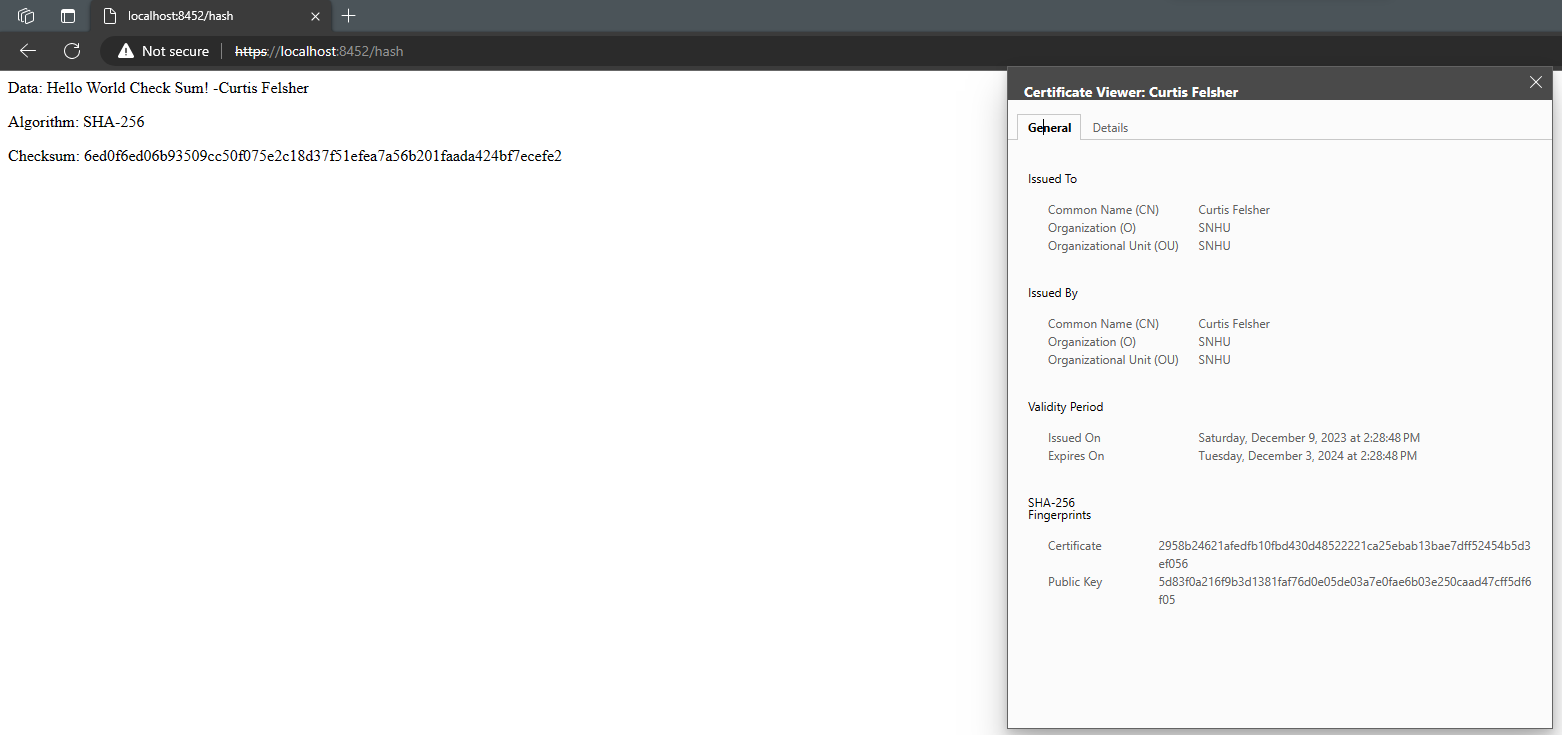
## Certificate Generation



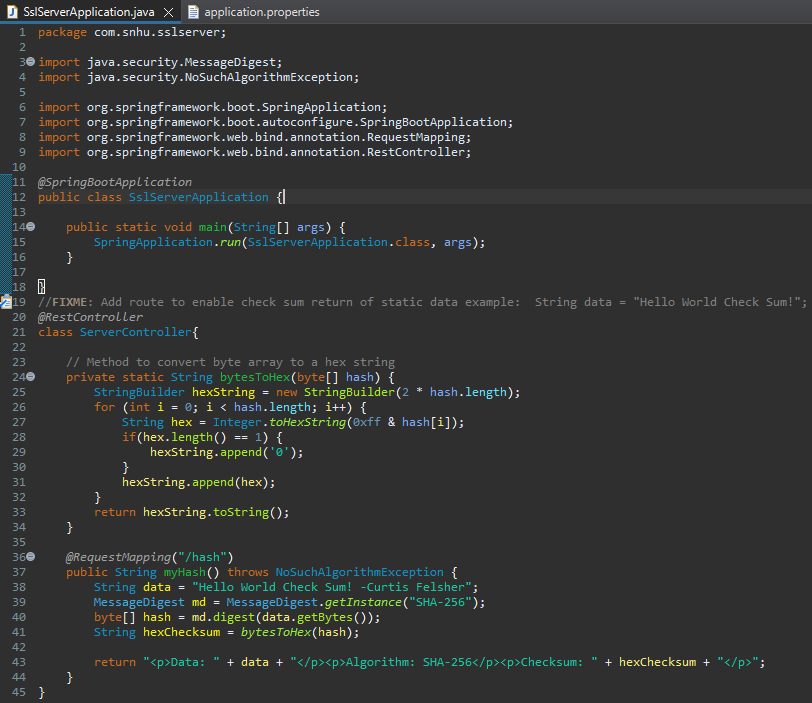
## Deploy Cipher

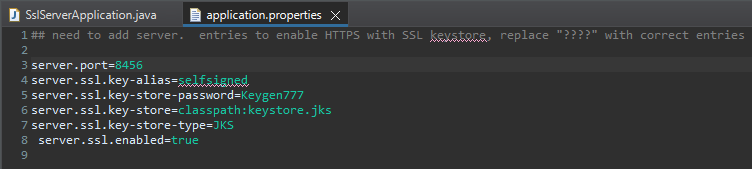


## Secure Communications



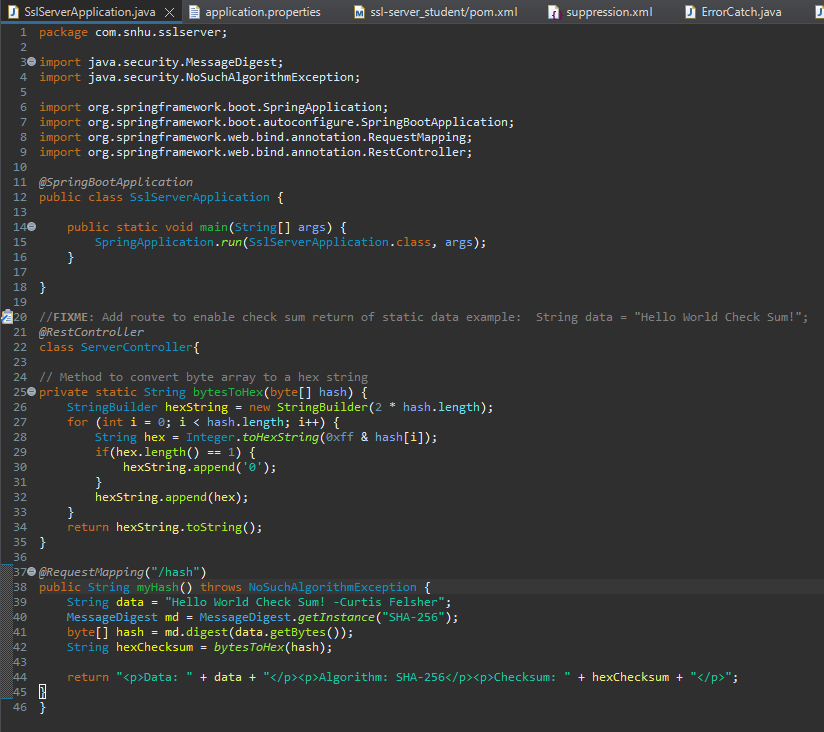
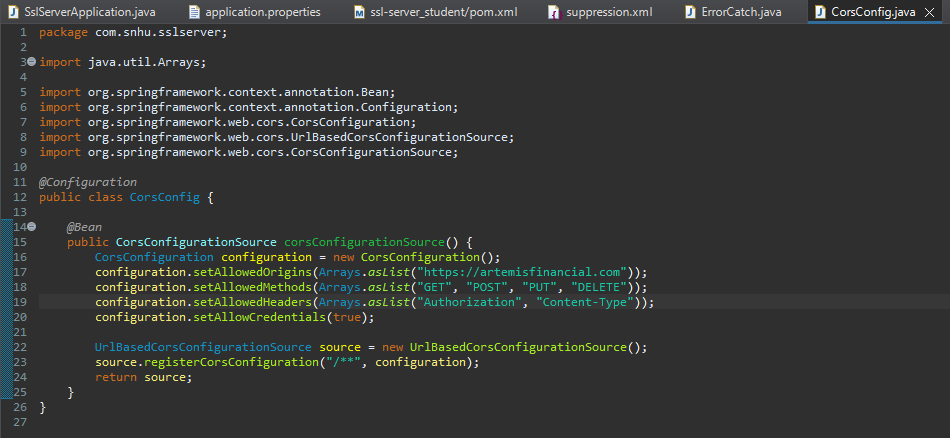
## Secondary Testing



A screenshot of a computer

Description automatically generated

## Functional Testing

A screen shot of a computer program

Description automatically generated

## Summary

1. Refer to the Vulnerability Assessment Process Flow Diagram. Highlight the areas of security that you addressed by refactoring the code.

Cryptography:

The implementation of AES and SHA-256 ensures data confidentiality and integrity. AES is used for data encryption to protect against unwanted access while SHA-256 is a hashing algorithm confirms that data has not been altered in transit.

Secure Error Handling:

ErrorCatch.java was implemented to provide secure error handling and exceptions. This approach prevents the exposure of system information to users.

Secure API Interactions:

CorsConfig.java was added to allow the application to control how external domains interact with its API. Specific domains are allowed to make requests, what type of requests, and which headers can be used to prevent XSS and other cross-origin attacks. Placeholders are currently being used.

1. Discuss your process for adding layers of security to the software application.

For this application, I took a multi-layered approach. After recognizing the need to protect sensitive financial data, AES and SHA-256 were chosen for this task as they are robust and efficient. Secure error handling was implemented with ‘ErrorCatch’.java to mitigate any risks of information leakage via error messages. This makes sure that errors are handled securely by providing necessary info to users with exposing any system details. Since the application is web-based, ‘CorsConfig.java’ was added to manage cross-origin requests to protect against cross-site scripting and potential data breaches.

## Industry Standard Best Practices

1. Explain how you used industry standard best practices to maintain the software application’s current security.

By utilizing a checksum, data integrity is verified every step of the way. This ensures that data remains the same and consistent from endpoint to endpoint. Error handling was implemented to obscure any system data transmission from end users to ensure system info confidentiality. A strict CORS configuration was implemented to control how external domains interact with the web application and prevent cross-site attacks. While this was implemented, strict testing was used to ensure that additional vulnerabilities were not introduced.

1. Explain the value of applying industry standard best practices for secure coding to the company’s overall wellbeing.

Companies benefit greatly from applying industry standard best practices to their applications. It helps ensure the security of both client and business data to maintain integrity and confidentiality. It also greatly enhances customer trust, which is essential for a business’s reputation and longevity. Compliance with any regulations helps avoid any penalties or legal issues that may arise. Preventing security breaches is also key as a data breach in the financial sector could spell the end of a company. Secure coding practices also promote increased application reliability and stability which leads to higher user satisfaction.