****

# 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** | **2/17/2025** | **Justin Leger** |  |

## 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

Justin Leger

## Algorithm Cipher

The most current and effective software security is AES-128 (Advanced Encryption Standard) which provides strong encryption for secure data transmissions at extremely high speeds. Given that Artemis Financial is a consulting company and develops individualized financial plans including savings, retirement, investments, and insurance; considering that Artemis Financial wants secure communications – AES-128 is appropriate. Why not AES-256? Because AES-256 has known weaknesses in its key schedule and is slower and less efficient than AES-128.

A brief overview of AES is that it’s a symmetric block cipher algorithm that secures electronic data by converting it into an unreadable format without using the proper key. AES operates on bytes rather than bits, is widely used in areas requiring secure communication, and is a cornerstone of modern cryptography.

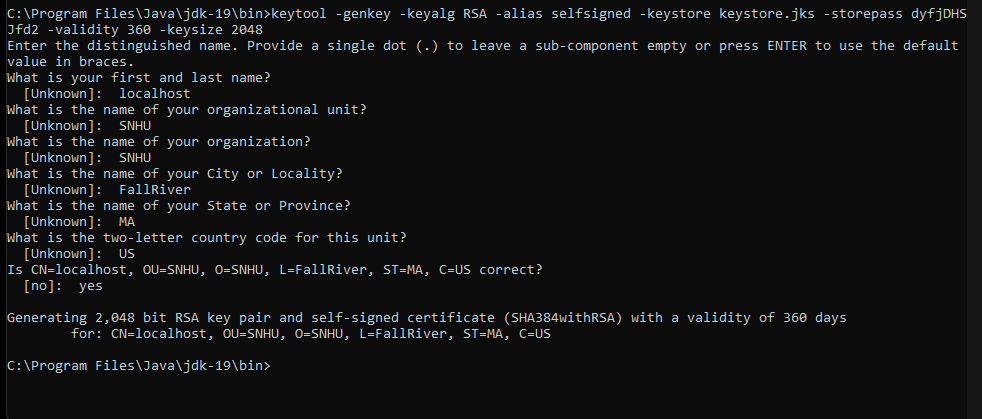
Bit levels of the cipher means that it takes 128bits of input and delivers 128bits of encrypted output, and it does this by performing operations on bytes of data. AES encrypts in several steps starting by substituting each byte for another byte; next each row is shifted several times; next matrix multiplication mixes columns; finally, the resultant output is XORed.

Symmetric cryptography uses a single key for both encryption and decryption the same way a single key can be used to lock and unlock a house door. Symmetric encryption is suitable for large data sets and efficient with computational resources, but the key must be shared between both parties and if the key is compromised then the entire communication is at risk. With asymmetric cryptography, a public and private key is used; the use of two keys makes it more secure. However, asymmetric cryptography is not suitable for large data sets and requires more computational power.

The history of encryption algorithms goes as far back as 1900 BC when a scribe used unexpected hieroglyphic characters instead of the real ones. Around 600 BC, the ancient Spartans used a scytale device to send secret messages during battle; 60 BC Julius Caesar invents a simple cypher that shifts characters three places in the alphabet. The first proper encryption key was envisioned in 1553 AD by Giovan Battista Bellaso; in 1854 Charles Wheatstone invents the Playfair Cipher which encrypts pairs of letters instead of single ones. Today encrypting data is crucial, especially as more information moves to the cloud; AES was invented in 1997.

## Certificate Generation

Insert a screenshot below of the CER file.



A computer screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Deploy and implement the cryptographic hash algorithm by refactoring code. Demonstrate functionality with a checksum verification.

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

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

A screenshot of a computer

AI-generated content may be incorrect.

Unable to get the site secured.

## Secondary Testing

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

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Functional Testing

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

User input sanitization could be done to prevent web attacks, such as improving upon weak input fields.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Summary

Discuss how the code has been refactored and complies with security testing protocols. In the summary of your practices for secure software report, be certain to address the following items:

1. Refer to the vulnerability assessment process flow diagram in the Supporting Materials section. Highlight the areas of security that you addressed by refactoring the code.
2. Discuss your process for adding layers of security to the software application.

I refactored the code so that it was secured over HTTPs using RSA encryption which prevents unauthorized access to the site. API returns checksum using SHA-256 that checks the fingerprints of the object and ideally, different objects should have a different hash.

Additionally, the code was put through a secondary static testing using the dependency check tool which shows common vulnerabilities. Finally, the input is sanitized, validated, and limited in length using a regex to match data with.

## Industry Standard Best Practices

Explain how you applied industry standard best practices for secure coding to mitigate known security vulnerabilities. Be sure to address the following items:

1. Explain how you used industry standard best practices to maintain the software application’s existing security.
2. Explain the value of applying industry standard best practices for secure coding to the company’s overall well-being.

Applying industry standard best practices develops much more resilient applications and customer satisfaction by streamlining security measures. Industry standard best practices are implementing robust authentication and access control mechanisms, keeping the application updated and patched, monitoring and auditing the application, establishing a software development life cycle, and understanding the importance of application security. I secured the webpage using RSA encryption, used secondary testing with the dependency check tool, and validated and sanitized input.

*13 Application Security Best Practices - OPSWAT*. (2023, April 12). OPSWAT. <https://www.opswat.com/blog/application-security-best-practices>

‌

*How to Create a Self Signed Certificate using Java Keytool. (n.d.). Www.sslshopper.com.* [*https://www.sslshopper.com/article-how-to-create-a-self-signed-certificate-using-java-keytool.html*](https://www.sslshopper.com/article-how-to-create-a-self-signed-certificate-using-java-keytool.html)

*keytool-Key and Certificate Management Tool. (n.d.). Docs.oracle.com.* [*https://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html*](https://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html)

*‌*

*‌*

‌