****

# 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** | **04/16/23** | **Deonne Ludwig** | **COMPLETED 1-8** |

## 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

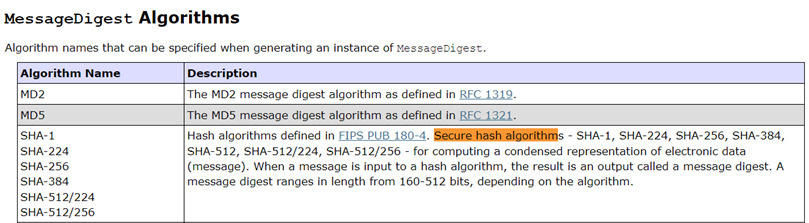
Deonne Ludwig

## Algorithm Cipher

## Recommend an appropriate encryption algorithm cipher to deploy, given the security vulnerabilities, and justify your reasoning. Review the scenario and the supporting materials to support your recommendation. In your practices for secure software report, be sure to address the following:

Secure Hash Algorithm (SHA-256)

<https://docs.oracle.com/en/java/javase/11/docs/specs/security/standard-names.html>



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

The best encryption algorithm for financial businesses depends on a variety of factors, like the security level required, the amount of data being encrypted, and the regulatory requirements of the industry, so to develop our security plan we would need to consider the threats that our application could potentially face and then work to address those issues. Some external threats that need to be assessed would include web application attacks, third-party services, DDoS attacks, backdoor attacks, injection attacks, phishing, spoofing, ransomware, etc. Modernization requirements will need to be considered to address the added security risks of open-source libraries and ever-evolving technologies like blockchain and IoT that might pose future risks. Since governments worldwide consider cryptographic algorithms as covert technology, care will need to be taken to address current government regulations for cyber security.

Our recommendation for Artemis Financial is, Advanced Encryption Standard (AES), a symmetric-key algorithm that is widely used for encrypting data in a variety of industries, including finance. It’s a fast and efficient high-security encryption algorithm that is currently one of the most secure available and a trusted standard used by the United States government. It uses a block size of 128 bits, but also supports 192 and 256 bits for even greater security. RSA is a public-key algorithm, is also often used in financial transactions that require strong authentication and data integrity. In cases where an organization is still using older systems, Triple Data Encryption Standard (3DES), a symmetric-key algorithm could be used to provide a high level of security. SHA-256 is the most secure hashing algorithm (scrambling of data beyond recognition) that is irreversible and trusted by the U.S. government. Although SHA-512 may offer a lesser chance of collision, it requires a 640 bytes lookup table compared to SHA-256’s 256 bytes lookup table which may save on storage costs. If that is not an issue the client could move to the SHA-512 algorithm.

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

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

Cipher hash functions and bit levels play an important role in the security of encrypted data. Hash functions take input data of any size and produce a fixed-size output (hash value) to verify the data has not been tampered with. Bit levels are the number of bits used to represent data in an encryption algorithm. The higher the number of bits, the more secure the encryption because it increases the complexity of the encryption and makes it harder for an attacker to break it. Java offers a SecureRandom class that will return a strong randomly generated number to use for securing your data. Symmetric cryptography uses a single secret key to encrypt and decrypt the data and non-symmetric cryptography uses one secret key and one public key.

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

Throughout history, as far back as Julius Caesar to World War II, cryptography has evolved from mechanical to digital and has become more complex and advanced to protect sensitive data. In the 1970s IBM developed Data Encryption Standard (DES) and it was adopted by the US government only to be replaced later by Advanced Encryption Standard (AES), which is currently the most widely used encryption algorithm. Encryption algorithms play a critical role in protecting sensitive data and ensuring the privacy and security of digital communication, but as technology continues to evolve, encryption algorithms will also need to evolve to keep pace with new threats and vulnerabilities.

## Certificate Generation

Insert a screenshot below of the CER file.

Text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

From module 5:

Graphical user interface, text, application

Description automatically generated

## Secure Communications

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

Graphical user interface, text, application, email

Description automatically generated

B: Couldn’t get the certificate to verify:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generatedC: I tried adding it to the trusted root certifications for chrome:

## Secondary Testing

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

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Functional Testing

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

Graphical user interface, text, application

Description automatically generated

## Summary

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

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

Since Artemis Financial exchanges sensitive information including savings, retirement, investments, and insurance the following areas of security would need to be addressed:

Input Validation – to ensure only properly formed data is entering the workflow in an information system.

API – to securely allow the capabilities of one computer program to be used by another.

Cryptography – to ensure that what is received is genuine, from the intended sender and data has not been tampered with during transmission.

Client/Server – to allow data transmission and protect the data.

Code Error – to diagnose and repair technical problems.

Code Quality – to prevent technical debt and security issues.

Encapsulation – to protect sensitive data and reduce unintentional errors.

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

Added RestAPI, encryption algorithm, static testing, manual code review, refactored code, checksum verification, utilized certificate authorities (CAs), and updated dependency checks.

## Industry Standard Best Practices

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

**Explain how you used industry standard best practices to maintain the software application’s current security. Explain the value of applying industry standard best practices for secure coding to the company’s overall wellbeing.**

Regular software updates, authentication and authorization, secure coding practices and error handling.

Applying industry standard best practices for secure coding is crucial for a company's overall wellbeing as it helps protect against cyber-attacks and data breaches, which can result in loss of data, damage to the company's reputation, and financial losses. Especially in the financial industry, specific regulations and standards are necessary for conducting business.