****

# 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** | **6/17/23** | **Austin Henley** | **Creation of document** |

## Client



## Developer

Austin Henley

## Algorithm Cipher

My recommendation for Artemis Financials’ algorithm cipher needs is AES256. AES is widely adopted for its security and efficiency. It is considered secure and efficient and is thoroughly tested to ensure so.

Hash functions are essential for data integrity, and for this purpose, I would propose the use of the SHA256 hashing function. It has a large output size of 256 bits, therefore providing a large range of potential hash values. This reduces the chance of two different inputs resulting in the same hash value. It is important for functions like this to be implemented to secure the data being moved within the program.

Random numbers are important when used in encryption algorithms. Their main function in cryptography is with encryption keys, where their randomness is essential for producing unpredictable and secure keys.

There are two different types of key encryption and they both have their advantages. Symmetric key encryption, such as AES, is effective for encryption of large amounts of data utilizing the same key for both encryption and decryption. Asymmetric key encryption provides confidentiality through using public and private key pairs, utilizing a public key that is shared openly and a private key that is kept secure.

Currently the focus of modern encryption algorithms is on the strength and compatibility with current systems. AES256 along with SHA256 (for data integrity) provides a combo that will surely keep Artemis Financials’ system secure.

## Certificate Generation

Self-signed certificate generation:

A screenshot of a computer

Description automatically generated

## Deploy Cipher

SHA256 hashing implementation:

A screen shot of a computer program

Description automatically generated with low confidence A picture containing text, screenshot, font, software

Description automatically generated

AES256 encryption and decryption function:

A picture containing text, screenshot, software

Description automatically generatedA screen shot of a computer code

Description automatically generated with low confidence

## Secure Communications

Screenshot of web browser:

A screenshot of a computer

Description automatically generated

The certificate:

A screenshot of a computer

Description automatically generated with medium confidence

## Secondary Testing

OWASP Dependency check

Before adding new code:  
A screenshot of a computer

Description automatically generated with low confidence

After adding new code:

A screenshot of a computer

Description automatically generated with low confidence

No issues were introduced from the added code for SHA256 hashing function and AES256 cipher.

## Functional Testing

Code executing without errors:

A screen shot of a computer program

Description automatically generated with low confidence A picture containing text, screenshot, font, software

Description automatically generated

A picture containing text, screenshot, software

Description automatically generatedA screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

A picture containing screenshot, text, font

Description automatically generated

## Summary

The original code has been refactored with security enhancements. It addresses the cryptography area of security, since the code now utilizes AES-256 encryption for data confidentiality and uses an SHA-256 hashing function for data integrity.

Secure code handling was also practiced in the code. Error handling is important to the integrity of the program, and throughout the code try and catch statements were used along with exception throwing to properly inform when an error has occurred. Different functions are also encapsulated inside of classes, with private variables used for encrypting data, and methods used for separating separate functions. This provides secure code by ensuring each function and class has its own tools for doing the job.

## Industry Standard Best Practices

In the code, industry standard practices were implemented to ensure the code is secure and efficient. A keystore was incorporated to secure cryptographic keys, so that the program can establish trust with clients through a digital certificate. HTTPS is used to ensure that client and server data is encrypted and private. The implementation of HTTPS helps maintain compliance with regulations, such as Payment Card Industry Data Security Standard (PCI DSS) or General Data Protection Regulation (GDPR), who require secure communication.

The code also demonstrates best practices in the code itself, with encapsulated and well commented methods. Methods are implemented within classes with private variables and are also commented thoroughly to explain what is happening within each method.