****

# 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** | **12-22-2024** | **Joseph Thomas** |  |

## Client



## Developer

Joseph Thomas

## Algorithm Cipher

Since we will be storing highly sensitive financial information on our servers, we will need to use the strongest cipher algorithm possible. I would suggest the AES-256 algorithm cipher for Artemis Financial as it is typically considered to be the strongest algorithm cipher for encryption (Kananda, 2024). The main downfall to this type of strong encryption is speed, it is slower and requires more resources to properly utilize (Nagaraj, 2024). The only reason I would not use AES-256 is if higher performance was necessary or the number of resources needed to process the encryption were too financially intensive. Since we are concerned about archives being stored long term, the speed doesn’t matter as much for Artemis Financial, but the encryption strength does.

Since we will be working with customers’ sensitive financial data, we must adhere to the Gramm-Leach-Bliley Act. The Gramm-Leach-Bliley Act was enacted November 12, 1999, and is overseen by the Federal Trade Commission for enforcement compliance. This act enforces proper reporting and disclosure of sensitive financial data by institutions and proper safeguarding of nonpublic, private information (*Gramm-Leach-Bliley Act (GLBA) Compliance at ECU - University Data Governance*, 2023).

Hash functions take input data and convert it to a compressed value, known as the hash value. The bit level determines the length of the encryption, and generally the higher the bit level, the higher the level of security because it will take longer to brute force (Cryptographer, 2023). Random number generators are used to increase unpredictability. A symmetric key is used to encrypt and decrypt the information, whereas an asymmetric key needs two different keys, one to encrypt and one to decrypt. AES-256 is considered symmetric.

Cryptography can be dated back to 1900 BC, where non-standard hieroglyphs were discovered in the ancient wall of a tomb in Egypt (Schneider, 2024). Cryptography has been used widely throughout history to secure communications. For instance, Cryptography has been used by the Ancient Spartans to scramble the order of letters in their military communication (Schneider, 2024). Encryption today is typically used in modern computing for secure communication and to prevent “bad actors” from seeing sensitive data and thus preventing data from unauthorized access.

## Certificate Generation

## A screenshot of a computer program Description automatically generated

A screenshot of a computer

Description automatically generated

I have attached my .CER file with the project.

## Deploy Cipher

A screenshot of a computer

Description automatically generated

## Secure Communications

A screenshot of a computer

Description automatically generated

HTTPS works but the certificate isn’t official because it is self-signed.

## Secondary Testing

A screen shot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

## Functional Testing

A screen shot of a computer program

Description automatically generated

## Summary

I chose to work with the SHA-256 hash cipher because it is one of the most secure hash ciphers on the planet and has little chance of collision.

## Industry Standard Best Practices

I used industry standard best practices including a secured RESTController and SHA-256 encryption cipher. I have also ran a security vulnerability test using OWASP for an automated analysis of known code vulnerabilities as well as ensured a manual code review to make sure the code is functional and secure.

**References**

Kananda, V. (2024, November 13). Why you should use AES 256 encryption to secure your data. Progress Blogs. https://www.progress.com/blogs/use-aes-256-encryption-secure- data

Nagaraj, K. (2024, August 27). Advanced Encryption Standard (AES): a secure and efficient symmetric encryption algorithm. Medium. https://infosecwriteups.com/advanced- encryption-standard-aes-a-secure-and-efficient-symmetric-encryption-algorithm- 319eedb49905

Cryptographer, J. a. |. C. (2023, February 28). The Bit-Security of cryptographic primitives. AWS Wickr. https://wickr.com/the-bit-security-of-cryptographic-primitives- 2/#:~:text=In%20its%20most%20general%20sense%2C%20the%20bit,well%20studied %20and%20carefully%20implemented%20cryptographic%20primitives.

*Gramm-Leach-Bliley Act (GLBA) Compliance at ECU - University Data Governance*. (2023, May 23). University Data Governance. https://datagovernance.ecu.edu/gramm-leach-bliley-act-glba-compliance-at-ecu/

Schneider, J. (2024, November 25). *Cryptography History*. https://www.ibm.com/think/topics/cryptography-history