****

# 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 5](#_Toc102040761)

[5. Secondary Testing 5](#_Toc102040762)

[6. Functional Testing 7](#_Toc102040763)

[7. Summary 8](#_Toc102040764)

[8. Industry Standard Best Practices 8](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **04/16/25** | **Alexander Freeders** |  |

## 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

Alexander Freeders

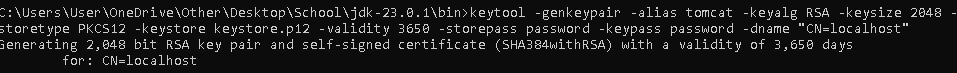
## Algorithm Cipher

The most appropriate algorithm cipher to use in this case is Advanced Encryption Standard (AES). AES is a symmetric key block cipher, standardized by NIST and generally regarded as the gold standard for secure encryption; with its high-complexity, 256-bit keys, and no known weaknesses (Portnox, 2025), it is a great choice for Artemis Financial.

AES offers up to a 256-bit key size and multiple rounds of encryption which makes it resistant to brute-force attacks and helps maintain confidentiality. Hash functions such as SHA-256 are used for integrity verification and authenticity. By using higher-bit algorithms, we are making the keys more complex and therefore more secure. In this instance, we are utilizing the symmetric key AES, which is much better suited for data storage. In regard to RNG or random number generation, RNG is extremely important in ensuring that our cipher is secure. RNG allows us to generate keys that are unpredictable and cryptographically random. RNGs are also responsible for Initialization vectors and salting hashes, which further help verify that data integrity and confidentiality are secure (Barker & Kelsey, 2015).

The history of encryption can date back to ancient times (simple transposition cipher, the Caesar cipher) through medieval times as well, but did not see leaps and bounds until the beginning of WWI. The first rotor machines used for cryptography were developed, and during WWII the enigma machine was cracked by allied codebreakers (Schneider, 2024). Modern day cryptography is generally noted as beginning in 1977 with DES, followed by 3DES in and AES in 2001. Post-Quantum Cryptography is the emerging field which many believe will be refined and streamlined enough for application soon (Chen & Scholl, 2022).

## Certificate Generation



## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

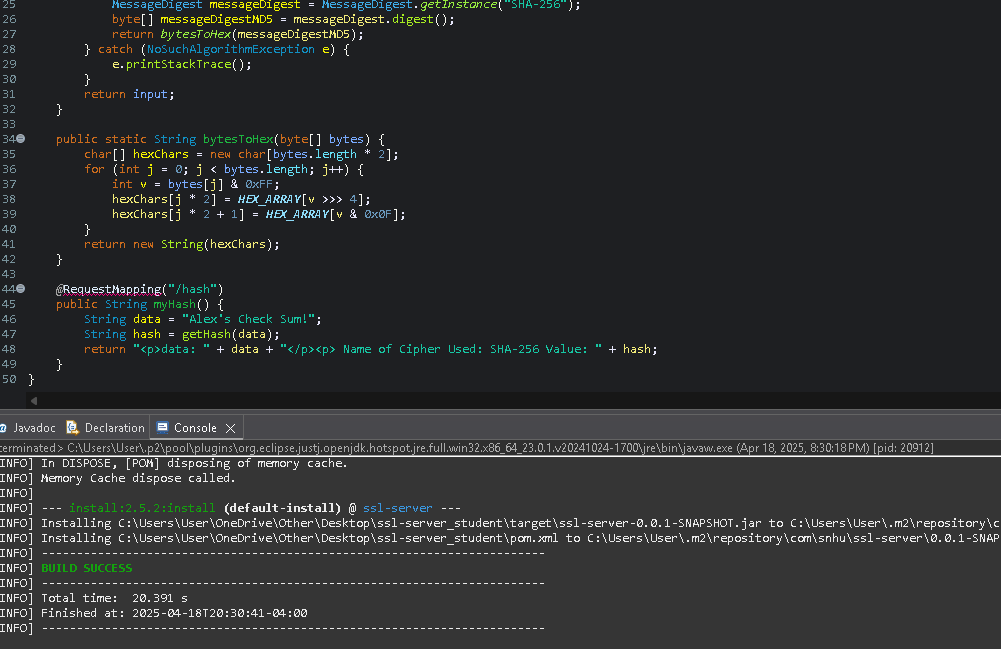
Insert a screenshot below of the web browser that shows a secure webpage.

A screenshot of a computer

AI-generated content may be incorrect.

## Secondary Testing

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



A screenshot of a computer error

AI-generated content may be incorrect.

## Functional Testing

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

A computer screen shot of a program

AI-generated content may be incorrect.

A screenshot of a computer error

AI-generated content may be incorrect.-

## Summary

The big changes that were made to the code were the implementation of HTTPS and SHA-256, which is made possible by our generated self-signed certificate that was created at the beginning of this project. After the self-signed certificate was generated, the Hash function was factored into the code. In order to ensure that it is working properly we implemented the checksum verification to receive the unique Hash value. The self-signed certificate is what allows us to use HTTPS, and while the hash value does not require a self-signed certificate, these blanketed layers of security are what is going to help ensure security for the client.

In regard to the vulnerability assessment process flow chart diagram, there were several areas of improvement, consolidated into a chart for ease of information:

|  |  |
| --- | --- |
| Security Areas addressed by refactored code | Explanation |
| Cryptography | Implemented SHA-256 to ensure data integrity with a unique hash |
| Client / Server | Implemented HTTPS on port 8443, establishing a secure client / server communication via TLS |
| Code Error | The try-catch block in the hashing function allows for secure error handling |
| Encapsulation | The hashing algorithm is encapsulated and separated from the rest of the application, all that needs to be entered is a string and a hash is generated, much the same with bytesToHex, which encapsulates the conversion logic into one separate method |
| Code Quality | Clear methods, constants, input sanitization via hash encoding |

## Industry Standard Best Practices

In refactoring this code, I implemented several industry standard best practices which align with OWASP top 10 and NIST guidelines. Using SHA-256 to generate a hash value allows us to ensure data integrity, which mitigates risks of data modification or tampering. HTTPS was utilized via a self-signed certificate to ensure secure communication between the client and server via TLS, which prevents eavesdroppers, MITM attacks and potential data leaks over unsecured networks. Error handling and minimal exposure of sensitive information are also best practices that were utilized; the hashing logic is encapsulated in a private method and the application avoids logging or displaying sensitive information. Lastly, Secure coding patterns are used which display clear method separation, clean coding standards and input sanitization (NIST, 2025).

There is tremendous value in applying security best practices for Artemis Financial. By ensuring that communications are secure, and data is hashed and stored securely, Artemis Financial is taking the proper steps to effectively position themselves for an application’s healthy future. The long-term value that is realized by implementing these best practices include risk mitigation, code maintainability, developer awareness, compliance and customer trust. Being able to reduce exposure to service disruptions and exploitation of vulnerabilities builds trust in the application with consumers—as well as the knowledge that their data and personal information are safely stored with integrity. Developers understanding a security-first mindset allows the code to refactored or maintained in the future while continuing to remain secure. Lastly, by utilizing security best practices, the company sets itself for success with potential audits by remaining in compliance with the necessary organizations.

**References:**

Barker, E., & Kelsey, J. (2015, June 24). *Recommendation for random number generation using deterministic random bit generators*. CSRC. https://csrc.nist.gov/pubs/sp/800/90/a/r1/final

Chen, L., & Scholl, M. (2022, May 26). *The cornerstone of cybersecurity – Cryptographic Standards and a 50-year evolution | nccoe*. NIST - National Cybersecurity Center of Excellence. <https://www.nccoe.nist.gov/news-insights/cornerstone-cybersecurity-cryptographic-standards-and-50-year-evolution>

NIST. (2025, March 20). *Security best practices*. NIST - Security Best Practices. https://www.nist.gov/itl/voting/security-best-practices

Portnox. (2025). *What is Advanced Encryption Standard (AES)?* <https://www.portnox.com/cybersecurity-101/what-is-advanced-encryption-standard-aes/#:~:text=AES%20is%20the%20gold%20standard,importance%20in%20protecting%20sensitive%20data>.

Schneider, J. (2024, November 25). *The history of Cryptography*. IBM. hSurettps://www.ibm.com/think/topics/cryptography-history