****

# 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** | **8/13/2024** | **Shae Machlus** |  |

## 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

Shae Machlus

## Algorithm Cipher

* 1. At Global Rain, we are partnering with Artemis Financial. Artemis Financial is a consulting firm that manages portfolios for savings, retirement, investment, and insurance. They are trying to increase their security of client data. This is where Global Rain comes in. I recommend applying SHA-256 (Secure Hash Algorithm) [1] to client data of Artemis Financial. This algorithm is used by the National Security Agency and the National Institute of Standards and Technology, which makes it a trustworthy choice. Common uses of SHA-256 include creating and verifying digital certificates and generating checksums [2].
  2. A hash function is applied to the client data to encrypt it. We use a 256 bit level version of the SHA-2 algorithm for security purposes. There are 2^256 possible keys that could be used to decrypt the hash value. This is a level of security where the possibility of a collision is statistically insignificant, so it’s perfect for securing sensitive financial information [3].
  3. SHA-256 can be used with symmetric and non-symmetric keys. If a symmetric key is used, it is a single private key shared only by the two parties. If asymmetric keys are used, then a public key and a private key are both shared. Random (or pseudorandom) numbers are relevant since the square roots of prime numbers are used as seed values [3].
  4. The history of encryption goes back to Ancient Egypt in altering hieroglyphs used in tombs. In 100 BC Caesar used a substitution cipher to write letters to army generals. However, this system was easy to figure out. Vigenère also used the substitution method while using an encryption key. The wartime encryption machine Enigma used the initial settings of rotors as their key, which was changed daily. With the rise of quantum machines, cryptography is still being studied to make security algorithms which are safe against quantum computer attacks [4].

## Certificate Generation

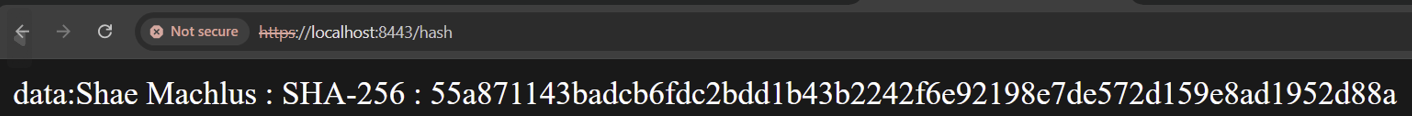
A screenshot of a computer

Description automatically generated

## Deploy Cipher

## 

## Secure Communications



Despite my refactoring of the application.properties to include a springboot.p12 certificate, I was unable to redirect the HTTP request to HTTPS.

## Secondary testing and 6. Refactored code executed without error

A computer screen shot of a program

Description automatically generated

A computer screen shot of a code

Description automatically generated

## Summary

I have developed a solution for secure data encryption of the sensitive financial data of Artemis Financial’s clients. The cryptography was successfully implemented using the SHA-256 algorithm. Errors in the code were mitigated using “throws” to make debugging a faster process. The code quality was ensured by the use of object-oriented programming and design patterns like singletons to create parent classes and iterators to loop through the digest to create the hexString. Code review was implemented using the Maven Dependency Check, which guided the updating of packages, which is a common way to remove vulnerabilities. Encapsulation was also used in keeping variables and methods public so that they may be used in the main code or other classes. These layers of security and best practices resulted in a robust codebase which would be trusted by data handlers largely thanks to the security of the encryption algorithm SHA-256.

## Industry Standard Best Practices

Functional spacing and indentation, concise comments throughout, and updated dependencies help bring the code to an industry-ready level of security. Artemis Financial can be a company that stands out for having almost no downtime and lightning-fast response speeds. Fast and easy security for the client will give Artemis Financial a good name overall, which will attract more business. If Global Rain continues to partner with Artemis as they need to scale up due to a larger customer base, then it’s a win-win. Global Rain will likely take their operations to the cloud using a service like AWS or Azure, while Artemis can hire more employees to manage their increase in clientele. Top-notch security gives consumers peace of mind and will encourage them to trust Artemis with more of their assets.

**Resources:**

[1] Oracle. (n.d.). Java Security Standard Algorithm Names. <https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html#cipher-algorithm-names>

[2] Wikipedia contributors. (2024, August 11). SHA-2. In *Wikipedia, The Free Encyclopedia*. Retrieved 22:20, August 18, 2024, from <https://en.wikipedia.org/w/index.php?title=SHA-2&oldid=1239730727>

[3] Gitlan, D. (n.d.). *What is the SHA-256 algorithm & how it works*. SSL Dragon. <https://www.ssldragon.com/blog/sha-256-algorithm/>

[4] Sidhpurwala, H. (n.d.). *A Brief History of Cryptography*. Red Hat. <https://www.redhat.com/en/blog/brief-history-cryptography>