****

# Practices for Secure Software Report

**Dhiraj Gurung**

**SNHU CS305**

**Professor Wayne Morris**

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** | **4/12/2025** | **Dhiraj Gurung** |  |

## Client



## Developer

Dhiraj Gurung

## Algorithm Cipher

The first standardized cipher was called DES. Over time, it evolved into what is now known as 3DES encryption. In its peak, it provided good security and an industry standard for some time. However, as technology has continued to evolve and computers have gotten much faster, 3DES is now considered outdated. The current industry standard for encryption and security is AES encryption. This is also the same encryption that I recommend Artemis Financial use.

I recommend using an Advanced Encryption Standard (AES) cipher because it provides a high level of data security while the data is being transmitted. The AES encryption utilizes a symmetric encryption key to protect data. This means that when data is sent between two users, the data is encrypted, and access to that data is denied until the correct encryption key is used to decrypt the data. The secret encryption key is a key part of the security equation when it comes to AES because the data “can't be understood by anyone who does not possess the secret key to decrypt it” (Cryptomathic, 2020). The secret key prevents any unauthorized access to data. However, this could be a potential issue as well because poor key management could lead to data leaks. Using key hierarchies and the best practices for key management can help mitigate and manage a lot of the potential issues with using AES encryption.

The hash function serves two main purposes: data integrity and identity verification (Crane, 2024). The first purpose, data integrity, just means that the data is kept secure and unchanged until it finds it way to the right user. Identity verification is the process of ensuring that the data is only accessible to people with the correct keys. The size of the keys can vary depending on the bit level of the cipher, and different ciphers utilize different bit sizes. For example, AES-128 uses 128-bit encryption keys while AES-256 uses 256-bit encryption keys. A 128-bit encryption key means that the length of the key is 128 bits, and the higher the bit number, the longer it will take computers to brute force their way through these encryption programs. Thus, a higher bit size generally means a higher level of security. However, the main drawback of going to a higher bit encryption is that performance speeds are slower. As the bit size increases, the performance speed generally decreases. As a result, choosing an encryption with both the right level of security and necessary speeds is essential for systems to operate smoothly.

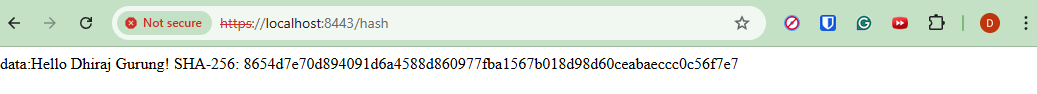
An encryption using a symmetric key means that the same key is used to both encrypt and decrypt the data. This means that both parties will need the same key in order to access the data. The benefit of symmetric key lies in quicker performance speeds and lower resource consumption (Poggi, 2021). However, it is less secure than using a two-key encryption system, also known as an asymmetric key. In asymmetric encryption systems, a public key that everyone has access to is used to encrypt the data, and a private key that is unique to each user. While both types of encryption can provide strong security, asymmetric encryption is generally considered slower but more secure. Additionally, when creating the keys, the generated keys are done so using random numbers because they provide unpredictability and security against brute force attacks. If an encryption is created with truly random or close to random numbers, then two messages that have 1 letter difference could produce wildly different ciphertexts, the encrypted form of the data. This way, it is a lot more difficult to guess the encryption key from the ciphertexts.

## Certificate Generation

A screenshot of a computer

AI-generated content may be incorrect.

## Deploy Cipher



Commands Used:

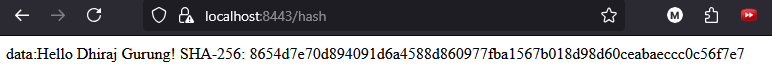
"C:\Program Files\Java\jdk-23\bin\keytool.exe" -genkey -keyalg RSA -alias selfsigned -keypass passcs305 -keystore keystore.jks -storepass passcs305 -validity 360 -keysize 2048

"C:\Program Files\Java\jdk-23\bin\keytool.exe" -export -alias selfsigned -storepass passcs305 -file server.cer -keystore keystore.jks

"C:\Program Files\Java\jdk-23\bin\keytool.exe" -printcert -file server.cer

"C:\Program Files\Java\jdk-23\bin\keytool.exe" -export -alias selfsigned -storepass passcs305 -file server.cer -keystore keystore.jks

## Secure Communications



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a certificate

AI-generated content may be incorrect.

## Secondary TestingA computer screen shot of a program code AI-generated content may be incorrect.

## A screen shot of a computer AI-generated content may be incorrect.

A close-up of a white background

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Functional Testing

A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Summary

I have refactored the code to have a higher level of code quality and security. I used the SHA-256 bit encryption to create a more secure environment for the data. This makes the data a lot less susceptible to brute force attacks and thus more secure. Additionally, I used a RestController to help manage the HTTP response from the Restful API. My process for adding layers of security first started off with creating a code with the basic elements, like having a @RestController and a @Requestmapping. Then I made sure to import all the packages and different resources that I was using, but only on an as-needed basis. This is because importing unnecessary packages or resources would be inefficient and cause a variety of issues, like causing confusion when trying to debug or maintain the code. I also tried to ensure that the code compiled without any issues, and if any issues were caught, they would be resolved.

## Industry Standard Best Practices

I used the standard best practices, like using an appropriate security encryption. I used a SHA-256 bit encryption because it was an encryption cipher that best combines security without slowing down the performance too much. Choosing the right encryption is crucial because using too weak of an encryption could cause sensitive data to be breached, and too strong an encryption could cause slower performance. I also used only the necessary amount of imports of different objects and made sure to implement line spacing and indents in a way that the code is more easily readable. These changes help to ensure that the code can be reviewed at a later stage with minimal confusion, as there won’t be unnecessary imports for unused files.

The industry standard for best practices exists because of trial and error. Other companies and developers have written code for many years and have realized through trial and error what type of code works the best and what does not. Following industry standard guidelines for coding will help the company at the very least have a base level of security that is at par with the rest of the industry, if not higher. Additionally, industry standards are often maintained because they comply with any government regulations that may be in effect. Following the standards may help the company stay in compliance with government regulations. Another advantage of maintaining industry standards is that they could help the company earn the trust of its customers by using modern-day technology instead of outdated software.

**Citations:**

Crane, C. (2024, November 5). *What is a hash function in cryptography? A beginner’s guide*. Hashed Out by The SSL StoreTM. https://www.thesslstore.com/blog/what-is-a-hash-function-in-cryptography-a-beginners-guide/#:~:text=Hash%20functions%20are%20a%20way,a%20means%20of%20identity%20verification.

Cryptomathic. (2024, December 5). *Symmetric Key Encryption: Uses in banking explained*. Symmetric Key Encryption: Uses in Banking Explained. https://www.cryptomathic.com/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking#:~:text=crypto%2Dagile%20infrastructure.-,AES,in%20U.S.%20FIPS%20PUB%20197.

Poggi, N. (2021, June 15). *Encryption Choices: RSA vs. AES Explained*. Prey Project. https://preyproject.com/blog/types-of-encryption-symmetric-or-asymmetric-rsa-or-aes#:~:text=Symmetric%20encryption%20involves%20using%20a,specific%20needs%20of%20the%20user.