****

# 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** | **4-4-2023** | **James Soto** |  |

## 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

James Soto

## Algorithm Cipher

When considering security protection best practices, it is crucial to account for various types of security attacks, risks, and current government regulations. The chosen cipher algorithm will be used to encrypt and decrypt sensitive data transmitted over a network or stored within a system. The primary goal is to protect data confidentiality, integrity, and availability. The cipher should be robust against common attacks, such as brute force, side-channel attacks, and cryptanalysis.

The Advanced Encryption Standard (AES) is widely considered to be the best symmetric encryption cipher. AES is a 128-bit block cipher supporting keys of 128, 192, and 256 bits, providing a high level of security. It has been adopted by the US government and is approved by the National Institute of Standards and Technology (NIST). AES is efficient, secure, and widely supported across various platforms and libraries. It is essential to conduct a thorough risk assessment and weigh the trade-offs between security, performance, compatibility, and operational requirements when selecting a cipher algorithm.

Purpose of cipher's hash functions and bit levels Hash functions are one-way functions that take an input and produce a fixed-size output (called a hash). In the context of encryption, they are often used for data integrity checks, password storage, and digital signatures. Although AES is not a hash function itself, it can be used in conjunction with hash functions (such as SHA-256) to create secure communication protocols.

Bit levels in AES (128, 192, and 256 bits) represent the length of the encryption key. Longer keys provide stronger security, as they make it exponentially more difficult for an attacker to brute force the key. However, longer keys may also require more processing power and time for encryption and decryption.

Use of random numbers, symmetric vs. non-symmetric keys Random numbers play a crucial role in cryptography, as they are used to generate encryption keys, initialization vectors (IVs), and other parameters that ensure the security and unpredictability of encrypted data.

AES is a symmetric-key algorithm, meaning it uses the same key for both encryption and decryption. Symmetric-key algorithms are generally faster and more efficient than non-symmetric (asymmetric) algorithms. Asymmetric algorithms, such as RSA, use a pair of public and private keys for encryption and decryption, providing additional security features like digital signatures and secure key exchange. However, they are slower and require more computational resources compared to symmetric algorithms.

In practice, both symmetric and asymmetric algorithms are often used together. For example, a secure communication protocol may use asymmetric encryption to exchange symmetric keys, which are then used to encrypt and decrypt data.

History and current state of encryption algorithms Encryption algorithms have evolved significantly over time. Earlier algorithms, such as the Data Encryption Standard (DES), provided a lower level of security and were eventually broken. In response, the US government and NIST conducted a competition to develop a new encryption standard, which resulted in the selection of AES (originally called Rijndael) in 2001.

AES has been extensively reviewed and analyzed by the cryptographic community and is currently considered secure for most applications. However, ongoing research into quantum computing poses a potential threat to many encryption algorithms, including AES. As a result, there is ongoing research into post-quantum cryptography to develop new encryption algorithms that can withstand attacks from quantum computers.

## Certificate Generation

Insert a screenshot below of the CER file.

Text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application, email

Description automatically generated

## Secure Communications

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

Graphical user interface, text, application

Description automatically generated

## Secondary Testing

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

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## Functional Testing

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

Text

Description automatically generated

Text

Description automatically generated

## Summary

Refactoring the code, I added a secure RestController to the SSLServerApplication.java file to serve as the secure controller for the hash RESTful endpoint. This ServerController class addresses the secure coding concern in the Vulnerability Assessment Diagram and fulfills the necessary requirements. I’ve chosen to use SHA-256 as the hashing cipher for this function, and the code is very minimal so as to reduce the potential attack surface. I also updated the version of the Maven Dependency check version from 5.3.0 to 8.2.1, so that the static dependency check is updated to the newest available software version.

## Industry Standard Best Practices

To mitigate against known security vulnerabilities and maintain the software application's current security, I applied industry-standard best practices for secure coding. Apart from making the changes I explained in the Summary section, it is essential that the proper security measures are implemented. The following are what should be used to ensure that the software security is maintained properly.

* Input validation: Validating and sanitizing user inputs to prevent attacks like SQL injection, cross-site scripting (XSS), and command injection.
* Secure authentication and password management: Implementing strong password policies and hashing algorithms, and using multi-factor authentication (MFA) to strengthen user access control.
* Principle of least privilege: Ensuring that users and applications have the minimum necessary permissions to perform their tasks, thereby reducing the potential for unauthorized access or actions.
* Secure data storage and transmission: Encrypting sensitive data at rest using industry-standard encryption algorithms and using secure communication protocols like HTTPS.
* Regular security updates and patching: Keeping the software application and its dependencies up to date, applying security patches promptly, and addressing any known vulnerabilities to minimize the risk of exploitation.
* Error handling and logging: Implementing proper error handling to prevent the leakage of sensitive information and maintaining secure logging practices to detect potential security threats.

Applying industry standard best practices for secure coding brings significant value to a company by protecting sensitive information, complying with regulations, costs savings, and finally gaining the trust of customers, partners all while contributing to a positive brand image and enhancing the company's reputation.

Resources:

GeeksforGeeks. (2022, February 11). Advanced encryption standard (AES). GeeksforGeeks. Retrieved March 20, 2023, from <https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>

What is AES encryption and how does it work? | Cybernews. (n.d.). Retrieved March 20, 2023, from <https://cybernews.com/resources/what-is-aes-encryption/>

Manico, J., &amp; Detlefsen, A. (n.d.). Iron-clad Java - Chapter 6. O'Reilly Online Learning. Retrieved March 20, 2023, from <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch06.html#ch06lev2sec3>

baeldung, W. by: (2021, November 14). Java AES encryption and decryption. Baeldung. Retrieved March 20, 2023, from <https://www.baeldung.com/java-aes-encryption-decryption>

Types of encryption: Symmetric or asymmetric? RSA or AES? Prey Blog. (2021, June 15). Retrieved March 20, 2023, from <https://preyproject.com/blog/types-of-encryption-symmetric-or-asymmetric-rsa-or-aes>

Bernstein, C., &amp; Cobb, M. (2021, September 24). What is the Advanced Encryption Standard (AES)? definition from searchsecurity. Security. Retrieved March 20, 2023, from <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard>