****

# 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** | **10 Oct 2023** | **Benjamin Dowell** |  |

## 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

Benjamin Dowell

## Algorithm Cipher

Advanced Encryption Standard (AES) is my recommended algorithm cipher. AES is the federal government standard as of 2002 and is the standard encryption algorithm established by the National Standards and Technology (NIST). Because this has been adopted as the standard, there has been extensive analysis done and is generally considered to be quite resistant to many types of attacks.

This algorithm cipher allows for secure communication of up to 256 bit but provides flexibility for less is if more performance is desired without unnecessarily detracting from security. The block cipher creates a fixed length string of characters, referred to as a hash value. The same input should always produce the same output, which makes for a great way to authenticate the encryption. It is not time-feasible to reverse the hash as the computations involved are significant to say the least. The hash is collision resistant as it would be near impossible to find two inputs that equaled the same hash value. The algorithm also has the versatility to encrypt keys in 128, 196, and 256 bit. There is a direct link to the level of security to the bit size of a key.

To fill the algorithm with new ciphers per key, the use of random numbers is necessary. Though it is said that computers are unable to truly produce random numbers, there are various levels to how good a computer can fake knowing how. That being said, an algorithm cipher that is able to do this well reduces the chance for any patterns and further reduces the off chances of anyone cracking a key. Symmetric cryptography employs the use of a single key for encryption and decryption. This is commonly used for data that is stored, files and databases. Non-symmetric cryptography on the other hand uses a key pair, a public and a private key. The public key is used for encryption and can be freely distributed. The private key is used for decryption and should be kept secure. This method is generally used when securing communication channels, establishing secure connections, and signatures.

Encryption algorithms have been around for a very long time, though not in the same capacity. From the Caesar cipher in 60 BC to the Enigma machine used by the Germans during WWII. Encryption algorithms use some kind of cipher to decode messages with the medium being non-important. Today, these algorithms have come a long way and are used heavily in securing data over the internet. Though the Caesar cipher may not seem all that impressive, lower forms of encryption can seem equally as unimpressive today. It all comes down to computing power, in Caesar’s time, one may only have an uneducated mind to crack his cipher and today we have computers. Encryption algorithms must be quite large, large enough so that a computer cannot even decipher through iteration in a reasonable amount of time.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with text and numbers

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A black and white text

Description automatically generated with medium confidence

## Secure Communications

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

I have made this certificate available to Windows and elevated it to a trusted CA. I can see the application offering the certificate when navigating to 8443 but my browser will not accept it for a secure connection.

A screenshot of a computer

Description automatically generated

## Secondary Testing

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

A white background with black text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A screenshot of a computer

Description automatically generated

## Summary

The program will need input validation once it requires input from the user. Secure API interactions should be achieved by creating a self-signed certificate allowing for HTTPS connections further allowing secure communications. The algorithm cipher allows for encryption of information passed through the application as shown with the checksum verification. The algorithm cipher and certificates allow for client/server to be secure. Code quality was enhanced by updating various dependencies that were previously vulnerable to various attacks.

The layers that were added to this application include encryption for secure communication between the client and the server, trusted certificate authority for the web browser to determine the site that is being visited is in fact the site you navigate to, and software updating to ensure known vulnerabilities are coded out.

## Industry Standard Best Practices

I utilized the industry-recognized secure coding standard OWASP (Open Web Application Security Project) to write secure code. I performed a code review scanning for common security issues such as injection attacks and authentication errors. Secure data transmission is used by encrypting communications. The dependencies used and the spring framework have been updated to newer versions.

The value added for applying industry standard best practices is customer trust. Having robust security profile in your application makes it so customers do not have to worry about their sensitive information while conducting business online. Just one data breach can severely impact a company’s reputation. It is important that, should this happen, you can at a minimum prove your company operated on industry standards. There is such a wide range of tools out there for secure coding, there is no reason or excuse to not follow best practices.