****

# 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** | **4/21/23** | **Kevin Andryzak** |  |

## 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

**[Insert your name here.]**

## Algorithm Cipher

The most effective encryption algorithm cipher to use is AES cipher because it supports a variety of key sizes, and it is one of the best standards in place today. Furthermore, the 128-bit or the 256-bit are the best choices for this application, with 256-bit being difficult for attackers to crack. According to Humadi (2020), its symmetric key creation will enable the application to encrypt data as required and deliver keys to the customers who are considered the recipients of communications. Applying symmetric and non-symmetric keys varies depending on what an application must accomplish. In addition, symmetric keys are mostly shared between the server and the client, as a result non-symmetric keys consist of public and private keys. Meyer (2016) elaborates that both keys encrypt data and decrypt only when the correct one is used. A downside to this is if the key to any encrypted data is lost, the information is lost as well because there will be no way to decrypt data without the key. Random numbers of generators can be applied to provide a special identifier to transactions that may assist with determining events such as data transfer or communication. Currently, AES can be encrypted using up to 256-bit, virtually immune to being cracked as a few values can be created. For this application, 128-bit was used to encrypt, because it offers secure communication between the server and the client.

## Certificate Generation

## 

## Deploy Cipher

A screenshot of a computer

Description automatically generated with medium confidence

## Secure Communications

## 

## Secondary Testing

## 

## Functional Testing

## 

## Summary

The refactored code is the “CS 305 Project Two Code Base.zip\_expanded” file. As a way of refactoring the code, the key areas of security addressed include cryptography, code quality, client, and server. I created a signed certificate and generated keys for the application that allowed a connection of 128-bit AES encryption. This encryption was for a particular recipient to read the data, providing security to the application communications. Secure communications are critical to maintaining this application since unsecured one can cause data leak leading to the loss of personal information. The outcome of the security breach can cause the loss of trust between the application owner and client and financial loss because of fines imposed. Maintaining security in the application will safeguard the company products and assets and ensure that trust is built.

## Industry Standard Best Practices

The industry best practice is by frequently checking and accessing the code for any vulnerabilities, particularly after executing new functions or changing current ones before publishing it to the live code. If any new vulnerabilities are found during the development process, the team should work on removing them. However, those with no solutions should decide what will work best and determine if the vulnerabilities can impact any component the application is using. The team will consider regular update of the system for any vulnerabilities

## References

Yedakula, K. (2019, November 30). Exploring the Differences Between Symmetric and

Asymmetric Encryption | Cyware Hacker News. Cyware-Social-Nuxt.

https://cyware.com/news/exploring-the-differences-between-symmetric-and-asymmetric-

encryption-8de86e8a

Advanced Encryption Standard: Understanding AES 256. (2019, July 29). N-Able.

https://www.n-able.com/blog/aes-256-encryption-algorithm

## Meyer, S. (2016). *The History of Cryptography*. Rosen Publishing

# Humadi, A. (2020). *Symmetric and Asymmetric Encryption*. Retrieved from

## <https://www.researchgate.net/publication/347494177_Symmetric_and_Asymmetric_Encryption>