****

# 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** | **4/18/2025** | **Aidan Gorospe** | **Creation** |

## Client



## Developer

Aidan Gorospe

## Algorithm Cipher

Due to the nature of hackers and malicious users attracted to financial data, I would recommend a more robust encryption, such as AES128. The 128 in the name refers to the length of the encryption keys, which would correspond to 32 characters. This length is estimated to take thousands of years to be cracked or brute-forced by even the most powerful of computers. However, it uses a symmetric encryption technique, meaning that the encryption and decryption keys are the same, which would require extra care in protecting said keys. For that, RSA should be put into place to help security.

Time is also a huge factor in why I chose this scheme, as while there are more secure options to choose from, this is still a business that needs to run and respond to clients fast. This scheme is considered the best of both worlds, as the encryption method mentioned above shows how secure it could be but compared to longer but slower performance encryptions like 256-bit, this is what I consider the sweet spot between security and performance.

Lastly, I would choose the CTR method for decrypting and encrypting the AES128, it does this by turning everything into a stream cipher, while also allowing the parallelization of encryption and decryption. This would mean that in large amounts of data, this would decrease the time it would take to both encrypt and decrypt. This method is much preferred in the modern day, compared to the slower performing 3DES and DES, as well as the security-lacking ECB.

## Certificate Generation

A computer screen with white text

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

## Deploy Cipher

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

A screenshot of a computer

AI-generated content may be incorrect.

## Secondary Testing

A screenshot of a computer error

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

## Functional Testing

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 refactored the code to allow the traffic to be sent over HTTPs instead of HTTP, preventing people trying to spy on the website and steal information. Communication uses the AES 128-bit cipher to secure the privacy of the users accessing the website and is also signed with a personal certificate.

The purpose of using checksums is to check and verify the history and integrity of the digital object. That means creating a hash from the object, such as a string, and checking if any information or file someone is accessing has been manipulated. There can be an issue with security regarding hashes, where two objects can have the same hash, which is known as collisions, which can result in malicious users reverse-engineering the original file. This can be solved by using SHA-256, which has a much smaller probability of collisions compared to other hashing functions. Alongside that, the input was created to be a string of lengths 1-25 characters to prevent buffer overrun attacks.

Using both a dynamic scanner and a static analyzer would be best to catch any exposures or vulnerabilities that occur during the runtime. Dynamic is especially useful as it catches things that a static analyzer would not catch, such as HTTPs redirects and memory leaks.

## Industry Standard Best Practices

To maintain the software application’s current security, proper practices had to be put into place. This resulted in constant implementation for problems such as Input Validation to account for data corruption or code injection, hashing implementation to account for password strengthening, and Error Handling to prevent information being given to malicious users.

This would also mean constantly checking and testing to see if there are any vulnerabilities or errors happening during said implementation and responding accordingly. This would be a mixture of using maven for automated testing to find any errors among the dependencies the original code uses, as well as manually checking and testing either the code itself or what maven returns, as false positives are known to show themselves. This would happen every time something new gets implemented into the existed code.