****

# 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** | **10/12/2023** | **Justin Swinney** |  |

## Client



## Developer

Justin Swinney

## Algorithm Cipher

For the given security vulnerabilities, I recommend we use AES – Advance Encryption Standard, AES is encryption algorithm that uses blocks of data and support keys in sizes 128, 192, 256 bits, AES is symmetric-key encryption (the same key is used for encryption and decryption). AES essential takes plaintext and blends the information together several times to produce cipher text that may consist of bits of hex code.

The bit levels of AES are dependent on the key size chosen, 128-bit key, 192 bits key, 256-bit key. 256-bit key being the most secure but may produce a slower algorithm process time when compared to a 128-bit key the less secure key size. All keys are essentially uncrackable at this current moment in time.

Hash functions typically correspond to cryptographic functions like SHA-256.

Random numbers are critical when creating encryption keys, the random aspect is what ensures that the key is not cracked by unwanted guests and keeps any sensitive information it is securing safe.

Non-symmetric key encryption typically has a slower process time than symmetric key encryption due to the greater length of the key (non-symmetric key = 2048 bit or higher, symmetric key <= 256). Non-symmetric encryption requires two keys, a public and a private where one encrypts, and one decrypts whereas symmetric has one key that handles both actions.

Currently AES is the industry standard, AES was implemented around 2001 replacing DES, DES was deemed inadequate, and an actual contest was held to determine the final retirement of DES, DES is a symmetric block cipher that shared a secrete key with a length of 56 bits. Assuming history repeats itself, we may see AES retire as technology advances.

## Certificate Generation

A screenshot of a certificate

Description automatically generatedA screenshot of a computer

Description automatically generated

## Deploy Cipher

A close up of a message

Description automatically generated

## Secure Communications

A screenshot of a computer

Description automatically generated

## Secondary Testing

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

## No new vulnerabilities or errors found at this moment,

A black background with text

Description automatically generated with medium confidence

## Summary

After running dependencies checks prior to refactoring this code and after I introduced zero new vulnerabilities to this program, however many exist currently. Areas addressed in the refactoring of the code include, Cryptography (check sum), code error (exception handlers), I feel as well that code readability is okay and code quality is okay.The security layers I implemented are minimal but include the SHA algorithm for creating secure hash values as well as using javas messageDigest to perform the hash function, this is a standard secure method for hashing in java, The site also uses https however I was having issues with my certificate securing the site locally. There is okay error handling within the code as well but still minimal.

## Industry Standard Best Practices

I feel that I followed industry standard best practices by using HTTPS for secure communication and using a strong cryptographic hash algorithm. The program does not allow for user input therefore less security risk involved, and no input verification needed. I feel I followed standard practices with my certificate configuration as well. Although I’m sure I have room for improvement.

Source here

[AES vs DES Encryption: Why AES has replaced DES, 3DES and TDEA (precisely.com)](https://www.precisely.com/blog/data-security/aes-vs-des-encryption-standard-3des-tdea)