****

# Practices for Secure Software Report

Table of Contents

[Practices for Secure Software Report 1](#_Toc180347465)

[Document Revision History 3](#_Toc180347466)

[Client 3](#_Toc180347467)

[Instructions 3](#_Toc180347468)

[Developer 4](#_Toc180347469)

[1. Algorithm Cipher 4](#_Toc180347470)

[2. Certificate Generation 5](#_Toc180347471)

[3. Deploy Cipher 5](#_Toc180347472)

[4. Secure Communications 6](#_Toc180347473)

[5. Secondary Testing 6](#_Toc180347474)

[6. Functional Testing 7](#_Toc180347475)

[7. Summary 7](#_Toc180347476)

[8. Industry Standard Best Practices 8](#_Toc180347477)

[References 9](#_Toc180347478)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **Oct 17, 2024** | **Kenneth Pinkerton** |  |

## 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

Kenneth Pinkerton

## Algorithm Cipher

1. Since Artemis Financial is a financial institution there is a bare minimum level of encryption that is required to do business as a financial institution by the government and federal regulators. (Federal Trade Commission, 2024) The accepted industry standard algorithm is 256-bit AES encryption. (Goodwin, 2021) This algorithm is nearly uncrackable by normal means, but it does have some vulnerabilities using a related keys attack. (Kiteworks, 2024)
2. An encryption algorithm cipher allows for messages to be securely transmitted and stored so that sensitive data or information is not disclosed to unauthorized parties, such as hackers. This is done through the processes of signing, to verify that data has not been compromised since it was transmitted, such as someone grabbing the file that was originally safe, altering it to allow for code execution on the system, and sending it on to the system with a “poison pill” attached. This is often done using a private key that only the company knows, not the customer, if the checksum is different than what is on file for the file that is transmitted then the system can reject this as having been tampered with without being exposed to any potential dangers inside. The other main encryption means is to have a public key that is used to encrypt information on both the customer and companies side, but in order to decrypt it a private key is used that are often held by a third party certified authority. This key is not transmitted, and is able to decrypt the information once both the company and customer have proven that they are authorized to see the information.
3. Ciphers work by taking plain text data and other forms of information using a programmatic method to convert each of the characters used to a unique hash value. This cannot be reversed without having the specific key that is used to decrypt the message. In AES the information to encrypt is broken into block segments of a set length and converted to a hash value that is so many bits long. AES-256 means that the information is broken into segments that are 256 bits in length. This is then converted to a hash value using a key that can be of different sizes from 128, 256, or 2048-bits.
4. In order for these systems to work a key is needed to act as a value to be entered into the system that will convert the information into both encrypted and decrypted formats. This key is used to create a seed that is then used to decrypt or encrypt the information being considered. Random numbers are often used to prevent the key from being guessed or easily cracked by attackers. This has been tied to system hardware such as timing events in a computers normal operations but these events may not come up frequently so using these for encryption could allow for someone to have sufficient time to come to the encryption that is being used. Another option that I came across in preparing this was seemingly off the wall, but works well is using a wall of 100 lava lamps to determine the keys used in SSL certificates at Cloudflare. (Cloudflare, 2024) Symmetric keys use the same key to encrypt and decrypt information this is often used in signing to confirm that data has not been changed since it was sent. Asymmetric keys work by using a public and private key. The public key is known by all parties and allows the information to be encrypted, but in order to decrypt the private key must be used in order to accurately reveal the data. This private key is usually held by either the company themselves or to prevent the private key from being discovered if there is a breach this key is often held by Certificate Authorities that are well known and trusted. In this way even if the data is breached hackers would not be able to tell what the information says because they don’t have the way to decrypt this to know what it says.
5. Encryption has been used since ancient times, and more modernly was made famous by the enigma machine used by the Germans during WW2. The Allies were finally able to crack the code, but it was thought at the time to be uncrackable, it wasn’t until a phrase was found to be repeated frequently that they were able to use this as a key of sorts to break the code. Currently AES-256 has been thought to be uncrackable, but because of advances in quantum computing that allow for thousands of computations to be run simultaneously this has shortened the time that was estimated to be millions of years down to days or weeks. (Hirlikar, 2024) However as of just 1 day ago there have been reports that a team of researchers in China has now used their quantum computer to crack AES, while it needs to be verified, this brings the thought of this being years or a decade away to immediately that this can be the case that AES is not as safe as it was. (McCartney, 2024)

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a certificate

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A computer screen with a black background

Description automatically generated with medium confidence

## Secure Communications

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

A computer screen with a black background

Description automatically generated with medium confidence

## Secondary Testing

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A screen shot of a computer code

Description automatically generated

## Summary

* 1. This code was refactored to include the SHA-256 algorithm cipher to comply with AES-256 protocols. This involves calling the private function “generateChecksum” by passing in a data object that is a string value of the hello world sample text. This value is then run through an iteration to convert each character to a hash format in compliance with SHA-256 and this variable is then returned as a string that is saved as “checksum”. This is then compiled with the original message and presented to the webpage as a series of paragraph tags so that the information is conveyed to the user for demonstration purposes. This works to ensure that data is encapsulated and that the code is kept clean to prevent unintentional errors from arising. The system uses a keystore that is password protected so that the only people that are able to use the program are those with the correct password “oPENsESAME123”. This keystore helps to protect user data by alerting the user if information is off from what is reported in the security certificate.
  2. My process for adding layers of security for the application involved ensuring that the base program was working correctly and then incrementally adding additional layers of security, verifying these were working as intended before proceeding to the next step.

## Industry Standard Best Practices

* 1. I strove to use industry standard best practices to ensure that the software application was on standard with what would be expected in the financial industry as per the clients market demographics. Some of this involved the selection of the correct type of algorithm cipher and the manner and structure of the code to adhere to to financial systems and software development best practices.
  2. By adhering to industry standard best practices it gives a clear picture of how the code should look and operate across many developers, this helps to both lower the time it takes to code a given solution as well as reduce the amount of errors that can arise. Even though there may be differences between how one developer an another may name a given variable the final output should ideally be quite similar. And as best practices improve over time the quality of work from all developers that adhere to this should likewise increase. This reduces cost for clients utilizing these developers, increases the value of these developers to the clients and companies they work for, and helps the industry overall to stay more secure and more cognizant of changes that arise in the industry, better allowing for vulnerabilities to be spotted early and mitigated against ideally before a breach is able to occur or propagate to minimize the damage to the given industry as a whole.

# References

Federal Trade Commission. (2024, October 17). *Gramm-Leach-Bliley Act*. Retrieved from Federal Trade Commission Protecting America's Consumers: https://www.ftc.gov/legal-library/browse/statutes/gramm-leach-bliley-act

Goodwin, M. (2021, February 11). *How Safe is Bank-Level Encryption*. Retrieved from Alarm New England: https://alarmnewengland.com/blog/bank-level-encryption/

Kiteworks. (2024, October 5). *Everything You Need to Know About AES-256 Encryption*. Retrieved from Kiteworks: https://www.kiteworks.com/risk-compliance-glossary/aes-256-encryption/#:~:text=AES%2D256%20encryption%20uses%20the,to%20transform%20it%20into%20ciphertext.