****

# 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** | **[Date]** | **[Your Name]** |  |

## 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 encryption algorithm cipher that I recommend is TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305\_SHA256. This is a cipher suite. Attacks can come in various types with different goals in mind. One type of attack is when the attacker attempts to receive information that they shouldn’t or send information out impersonating a different sender. One best practice to combat this is authentication. The system should verify that the correct person is sending the information and that the intended user is the one that receives it. This is being authenticated by this cipher suite. Another attack would be to enter the system through malicious software. One best practice to prevent this is to validate the security certificate and verify it is not expired. Another feature that should be considered is forward secrecy. This protects past communications even if an attacker does manage to get the private encryption key. This is provided by this cipher suite.

One risk with using a cipher suite is that the other endpoint (the user’s system) may not support it. Testing would need to be confirmed to make sure it works. Another risk is that the system may slow considerably. While we are looking to keep all information safe, the drawback to encryption is that it will slow communication. There are always other risks that could come up but this appears to be the best option for this request.

Because Artemis Financial is a company that accesses and controls individuals’ money, there are government regulations that they must comply with. One of these is the Sarbanes-Oxley (SOX) act of 2002. This framework looks to keep investors safe. There are numerous requirements to this act. For example, any breeches or breech attempts must be tracked and the attempts to fix the problem must be recorded as well. The Gramm–Leach–Bliley Act is another regulator that must be obeyed. Under this act, (Upguard, 2025) “ financial entities must establish security controls to protect customer information from any events threatening data integrity and safety.” Failure to comply with government regulations can result in penalties including loss of licenses, monetary penalties and potentially even shut downs. All government regulations will need to be monitored and verified to have been met. One key to that is verifying that all security certificates are in place and up to date. The encryption algorithm also needs to be monitored to make sure that it stays compliant.

This cipher suite will be entered into the code so that all communications will be appropriately encrypted. It will be used to confirm all messages sent between the user and the sender stay between those two endpoints. Each cipher has different qualities that are appropriate for different situations. The reasons for using this cipher are not including anon which then means that the users are authenticated. It also does not allow for the suite to be exported which maintains a safety level. It has a larger key size of 256. The ECDHE indicates that the suite does provide forward secrecy. It also has an 18 octet authentication tag which does have less risk of forgery. In order to keep speed faster, the ECDHE was chosen instead of the DHE. There is a fine balance between security and performance that needs to be evaluated.

The purpose of the hash function in a cipher algorithm is to take an input and transform them into a larger fixed length. If an attacker was able to get to the data, the more characters that are encrypted, the harder it would be to decrypt that information. One place this can play a key role is in password verification. The cipher suite recommended has poly1305 which is one hash function with a 16 byte secret key which verifies the message has not been tampered with. The SHA-256 indicates a hash function which transforms input into 256 bit. The larger the bit the more complex and stronger the security is for the program.

This cipher suite uses ECDSA which is an elliptic Curve Signature Algorithm. This is an asymmetric encryption algorithm. This is a positive thing because the user’s information is sent via a public key encryption but then it needs a private key to decrypt it. Since it takes two keys, it can be harder for an attacker to decrypt the message.

While one can trace encryption itself all the way back to the Greeks, the first modern encryption algorithm was developed in the 1970’s as DES which is Data Encryption Standard. This was developed by the National Bureau of Standards. This was only a 56 bit key. DES is currently still used but has also been expanded on. This developed into the public key. Triple DES was the next development. AES grew out of that which has much larger keys. Now there are a large number of variants of the encryption algorithm. Different companies are building their own algorithms which have different strengths and weaknesses. The goal is to find the most suitable algorithm for the software’s needs. Again, there is a balance that needs to be obtained between security and performance.

## Certificate Generation

Insert a screenshot below of the CER file.

A black and white line

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screen shot of a computer program

AI-generated content may be incorrect.

## Secure Communications

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

A screenshot of a computer

AI-generated content may be incorrect.

## Secondary Testing

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

I continue to get the same error that we have been discussing via email for Module 6. I am not sure how to get around this. I have spent hours searching for a solution and trying different things to no avail. I would love to see if I can get this working so I could perform the rest of the project.

## Functional Testing

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

I continue to get the same error that we have been discussing via email for Module 6. I am not sure how to get around this. I have spent hours searching for a solution and trying different things to no avail. I would love to see if I can get this working so I could perform the rest of the project.

## Summary

I continue to get the same error that we have been discussing via email for Module 6. I am not sure how to get around this. I have spent hours searching for a solution and trying different things to no avail. I would love to see if I can get this working so I could perform the rest of the project.

## Industry Standard Best Practices

I continue to get the same error that we have been discussing via email for Module 6. I am not sure how to get around this. I have spent hours searching for a solution and trying different things to no avail. I would love to see if I can get this working so I could perform the rest of the project.

**Citations**

CS 305 Project Two Guidelines and Rubric

<https://learn.snhu.edu/d2l/le/content/1918286/viewContent/40563418/View>

Java Security Standard Algorithm Names. (n.d.). <https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html#cipher-algorithm-names>

Al Tamimi. (n.d) Performance Analysis of Data Encryption Algorithms. <https://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/#2>

Manico, J. & Detlefsen, A. (2014) Iron-Clad Java. <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/?sso_link=yes&sso_link_from=SNHU>

Kost, E. (2025, January 02). Top 9 Cybersecurity Regulations for Financial Service. UpGuard. <https://www.upguard.com/blog/cybersecurity-regulations-financial-industry>

Public Key Cryptography. (n.d.). <https://cs.stanford.edu/people/eroberts/courses/soco/projects/public-key-cryptography/main.html>