****

# 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** | **12/10/23** | **Antonio Colon** |  |

## Client



## Instructions

Submit this completed practice for a 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

Antonio Colon

## Algorithm Cipher

The appropriate encryption algorithm cipher I would recommend would be AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman known as cryptosystem). AES would be a wise choice as it is a widely used encryption standard. It was selected by NIST (National Institute of Standards and Technology) as a government standard and is associated with FIPS (Federal Information Processing Standards Publication). RSA would be used for the asymmetric key encryption algorithm to ensure security in its data and signature. AES has multiple key sizes of 128, 192, and 256 bits that encrypt into blocks of 128 bits for the bit level. AES deals with block ciphers and is not a hash type of encryption. RSA can be used with hash but is not a has function itself. RSA operates using different key sizes from 1024 up to 4096 bits.

The random number function is used to create a string of characters that generate an encryption piece for the data set. The difference between symmetric and non-symmetric is based on using secret keys. Symmetric uses the same type of key for decrypting and encrypting data. As for Asymmetric, it uses 2 different keys which are private and public. These can then be used along with a signature to encrypt or decrypt data. RSA would be best for this exact function. Ciphers in the past had many uses throughout history for wars, love, secret affairs, and much more. These secret messages were done back in early Egyptian times if I remember correctly. There would be hidden messages inside containers and on the outside of them hieroglyphics to encode them. These characters were long to ensure anybody had them in position. It would take a long time to open one of the containers. The current state of encryption is a lot harder to crack and has become standard since the brute-force attack made on DES (Data Encryption Standard). With the evolution of technology, we are going to have stronger encryption methods and even better security practices with key management.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a computer

Description automatically generated

Had to create an exception due to firewall and browser security.

## Secondary Testing

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

Before the code was entered into the program.

A screenshot of a computer error

Description automatically generated

Showing the refactored code executed without any errors.

A screen shot of a computer program

Description automatically generated

Screenshot showing the report of the dependency-check static tester after the code was written with no errors.

A screenshot of a computer error

Description automatically generated

## Functional Testing

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

A screen shot of a computer

Description automatically generated

## Summary

I have refactored the code to include the SHA-256 encryption and the library functions to help alongside them. This also includes the application.properties to allow the connection to the program using RSA. Have verified using the Vulnerability Assessment Process Flow to ensure API was functional along with Cryptography and Encapsulation. Each time a code was added in each section, I ensured that it had errors by running it multiple times. This way, it wouldn’t be hard to determine where an error came from along the way. Also using Code Quality to ensure the code worked properly when hashing onto the HTTPS page.

## Industry Standard Best Practices

When working on maintaining this software, would suggest adding time stamps with comments along the way. To ensure any issues that arise can be reverted to their original state. Backups are encouraged as well with cold and hot methods on the server. Also, communication with team members on the project can help with conflicts at work. That way, errors are less likely to occur when working on the backend of things. It is always best practice to follow along with standards and procedures when working on a project. This helps keep the scope of work running smoothly for any project manager keeping track of the work. Especially if a client or another associate is inquiring about a certain task along the way. Documented procedures and communication can be processed or discussed at that moment. Following through with the workflow can also be helpful if certain unforeseen events occur along the way. Then a delegation of tasks or even additional help can be changed around due to the nature of business.

**Reference**

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

Pietricica, Andreea-Catalina, et al. “Evaluation of AES Cryptographic Algorithm on a General-Purpose Map-Scan Accelerator.” 2023 International Semiconductor Conference (CAS), Semiconductor Conference (CAS), 2023 International, Oct. 2023, pp. 99–102. EBSCOhost, <https://doi-org.ezproxy.snhu.edu/10.1109/CAS59036.2023.10303705>

Berinato, S. (2000, October 16). AES standard proposal - NIST contest winner Rijndael may be the future encryption technology. eWeek, 52. <https://link-gale-com.ezproxy.snhu.edu/apps/doc/A66106423/GBIB?u=nhc_main&sid=ebsco&xid=c0c3d603>

MIL-OSI USA: NIST Updates FIPS 197, Advanced Encryption Standard (AES). (2023, May 10). ForeignAffairs.co.nz, NA. <https://link-gale-com.ezproxy.snhu.edu/apps/doc/A748706883/STND?u=nhc_main&sid=ebsco&xid=5f86bc10>

Manico, Jim, and August Detlefsen. “Iron-Clad Java: Building Secure Web Applications.” O’Reilly Online Learning, McGraw Hill Computing, <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch06.html#ch06lev1sec16>

Sheposh, R. (2023). RSA (cryptosystem). Salem Press Encyclopedia of Science. <https://search-ebscohost-com.ezproxy.snhu.edu/login.aspx?direct=true&db=ers&AN=87323266&site=eds-live&scope=site>