****

# 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** | **2/20/2023** | **Kristen Hawkins** |  |

## 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

Kristen Hawkins

## Algorithm Cipher

## Provide a brief, high-level overview of the encryption algorithm cipher.

## Discuss the hash functions and bit levels of the cipher.

## Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.

## Describe the history and current state of encryption algorithms.

I would consider using SHA-256 encryption for this project because there is money and personal data involved when working with this company. We need a secure way to access files and encrypt them when we send data back and forth. SHA-256 is used when you convert user passwords into hash value. This is done before storing them in the database. SHA-256 algorithm is used to verify the data integrity of files on the internet. When a file is uploaded to the internet, a hash value is set that is associated with that file. When a user downloads that file, the hash is recalculated to establish its’ data integrity. The SHA hash functions extensively creating and verifying SSL and TSL digital certificates. These certificates enable an encrypted connection between the web servers and we browser. SSL certificates are required to secure online transactions and to protect customer information (Lepcha, 2023). According to Prey Project, symmetric encryption uses the same key for decryption and encryption. Asymmetric encryption uses a pair of keys: a public key and a private key for decryption. The reason for the random numbers and letters in an encryption is to make it very hard to hack into or nearly impossible. Math is also used as a tactic in cryptography algorithms. The earliest evidence of cryptography was found in ancient Egypt. This dates back nearly 4,000 years ago from the tomb of Khnumhotep II. Ancient encryption was used for just about the same reason we use it now, to keep information safe. Military encryption was first recorded around the time 500 BC. This allowed the Spartan Military to send and receive secret messages among themselves (Team, 2022). Encryption has grown in 4,000 years and we get more and more advanced in cryptography as we grow in technology (Team, 2022).

# References

Lepcha, M. (2023, July 26). *What Is SHA-256?* Retrieved from www.techopedia.com: https://www.techopedia.com/definition/sha-256#:~:text=SHA%2D256%20is%20used%20to,user's%20input%20during%20login%20verification.

Team, T. (2022, Jan 14). *The history of encryption: the roots of modern-day cyber-security*. Retrieved from tresorit.com: https://tresorit.com/blog/the-history-of-encryption-the-roots-of-modern-day-cyber-security/

POGGI, N. (2021, June 15). *Types of Encryption: Symmetric or Asymmetric? RSA or AES?* Retrieved from preyproject.com/: https://preyproject.com/blog/types-of-encryption-symmetric-or-asymmetric-rsa-or-aes#:~:text=Symmetric%20vs%20Asymmetric%20Encryption,-Asymmetric%20and%20symmetric&text=Symmetric%20encryption%20uses%20the%20same,a%20private%20key%20for%20decryption.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

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

## Secondary Testing

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

Refactored code:

A screenshot of a computer code

Description automatically generated

Dependency check:



## Functional Testing

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

Code Running with no errors:

A screenshot of a computer

Description automatically generated

Summary

After refactoring this project, I implemented the client and server side making sure that they are secure using the SHA-256 algorithm. I also used cryptography to encrypt vulnerabilities withing the code. This is also where SHA-256 comes in to hash the information that we don’t want seen by outside users. APIs were also used to bring the hash to a secure API web server. This was done by also using a digital certificate. Lastly, I used code quality, I secured the code by using best practices and patterns. The project was running more than two times making sure that everything came out as it was planned to. The layers of security that was used in this process was making sure that the information and web browser was secure for the client and server. Applying an extra layer of security inside the code while creating a certificate for the browser.

## Industry Standard Best Practices

I used industry standard best practices in this project by finding out where errors were in the code, then I implemented an API into the code, along with the hash functions and HTTPS server. I then ran regular dependency checks on the project to make sure that I could point out the vulnerabilities within the code. The best practice for securing this code and for the company’s wellbeing is to run dependencies checks and suppressing checks to find hidden vulnerabilities.