****

# 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

Robert Szabo

## Algorithm Cipher

“I have chosen to use SHA-256, which, while not an algorithm cipher, is a hash function that can do a good job securing an application. SHA-256 is a good cryptographic algorithm because it always produces a 256-bit output, making it easier to use. Additionally, because it is a one-way function, you cannot reverse the hash to obtain the original data. Also, it is collision resistant, meaning it is nearly impossible for two different inputs to produce the same hash output, ensuring the integrity of your data. Finally, this is considered secure because as of now, there are no known ways to attack and defeat this hash function (Szabo, 2025).”

Cryptography uses something called cryptographically secure random number generators (CSPRNG’s) to ensure unpredictability of the encrypted data. In a symmetric encryption, both the sender and receiver use the same key to encrypt and decrypt the data. It is know to be faster and more efficient than asymmetric keys, but less secure. Asymmetric keys use a public key to encrypt and a private key to decrypt the data. This solves a few problems that occur in symmetric keys in that you don’t need to distribute keys in advance because the public key can be shared with anyone, while only the private key needs to remain secure.

Throughout history, cryptography has evolved from ancient times with examples including the Caesar Cipher, known for shifting each letter in a message a fixed number of cases. During the time it was used, it was very effective, but with today’s technology it is easy to crack with brute force or frequency analysis. You then started seeing other ciphers grow throughout the middle age and renaissance periods that worked well during their time, but again, were able to be cracked easily in the modern era. After the discovery and harnessing of electricity and the industrial revolution, ciphers like the Enigma Machine baffled allied scientists and mathematicians until Alan Turing was able to lead a team that was able to crack the code without even the use of computers. Turing went on to be a major figurehead in the world of early computing and is still well known today. As electronic computing rose, the Data Encryption Standard was developed by IBM using 56-bit symmetric keys, but was able to be cracked using brute force methods due to weak key length. Asymmetric cryptography was created with the invention of RSA, which is still used today. In the current state of affairs, you see the evolution of DES into the Advanced Encryption Standard (AES) in 2001 which exists in symmetric block ciphers of 128, 192, and 256 bit keys, as well as others like SHA-2 and 3. As we move into the future, quantum computing looms on the horizon as a threat to current cryptographical methods in that it can process data faster by leaps and bounds compared to traditional computers, but many leading researchers are working on encryption methods that will be resistant to these new computing technologies.

## Certificate Generation

A screenshot of a certificate

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Deploy Cipher

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

A screenshot of a computer

AI-generated content may be incorrect.

## Secondary Testing

A computer screen shot of text

AI-generated content may be incorrect.

Original Dependency Check:

A screenshot of a computer

AI-generated content may be incorrect.



Second Dependency Check:

A screenshot of a computer

AI-generated content may be incorrect.



## Functional Testing

A screen shot of a computer program

AI-generated content may be incorrect.

## Summary

Refactoring the code, I’ve created a secured RestController to secure my program. I chose to use the SHA-256 hashing cipher because of its high-level 256-bit security and its ability to avoid collisions. This helps ensure that we are meeting the standards of cryptography in the Vulnerability Assessment Process Flow. Ensuring there were no errors or quality issues in the code by running dependency checks helps to consistently ensure the security of the software. I would also recommend maintaining a cycle of running these checks to ensure that as time goes on, no new problems are identified that impact this software.

## Industry Standard Best Practices

Using industry standard best practices to secure your software and ensure secure code helps to maintain the company’s image and standing among other companies in the field. Security breaches can destroy the public’s trust in a company and if you do not respond effectively and ensure breaches of that nature do not occur again as well as to ensure the minimization of other breaches, you can lose business and ultimately impact the company’s bottom line and ability to do business with other businesses.

References

Szabo, R. (2025, June 8). *CS 305 Module Five Coding Assignment Checksum Verification* .