****

# Practices for Secure Software Report

Table of Contents

[Practices for Secure Software Report 1](#_Toc153148730)

[Document Revision History 3](#_Toc153148731)

[Client 3](#_Toc153148732)

[Developer 4](#_Toc153148733)

[1. Algorithm Cipher 4](#_Toc153148734)

[2. Certificate Generation 7](#_Toc153148735)

[3. Deploy Cipher 8](#_Toc153148736)

[4. Secure Communications 9](#_Toc153148737)

[5. Secondary Testing 11](#_Toc153148738)

[6. Functional Testing 12](#_Toc153148739)

[7. Summary 13](#_Toc153148740)

[8. Industry Standard Best Practices 14](#_Toc153148742)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **12/10/2023** | **Tomas Estanislao** | **1st publish; Initial report.** |

## Client



## Developer

Tomas Estanislao

## Algorithm Cipher

My recommendation for an algorithm cipher is AES-256, specifically Advanced Encryption Standard 256 implementation. There are various areas of concern for a web-based software application including Input Validation, APIs, and Cryptography. Algorithm ciphers are in the domain of Cryptography. The main risk of algorithm ciphers is brute force attacks, where malicious actors try to guess the right secret key through all the possible key combinations. There are also vulnerabilities to consider on whether the data being protected is at rest or in transit. Artemis Financial is looking to encrypt long-term archive files— at rest. From *TechTarget*, “a major risk to AES encryption comes from [side-channel attacks](https://www.techtarget.com/searchsecurity/definition/side-channel-attack)... Side-channel attacks involve collecting information about what a computing device does when it is performing cryptographic operations and using that information to [reverse-engineer](https://www.techtarget.com/searchsoftwarequality/definition/reverse-engineering) the device's cryptography system” (Bernstein & Cobb, 2021). Despite the risks, this is considered the currently best cipher available, “The AES encryption standard was approved by the National Security Agency (NSA) to protect both secret and top-secret government information. It has since become an industry standard for encrypting information. It is an open standard, which means it can be used for public, private, commercial, and non-commercial implementations” (Kiteworks). AES-256 will be used to encrypt and decrypt sensitive data within the organization; AES complies with the U.S. government’s regulations and is the most applicable for Artemis Financial's particular use case. For data that needs to be protected in transit, or where there are limits on computational power, other cipher may be more applicable.

           AES-256 uses a hashing algorithm, on a given password, to produce a secret key 256 bits long. The hashing extends the length, and the bits determine the length— the longer being the most secure. AES uses the Rijndael cipher for symmetric encryption, from *Medium*, “symmetric encryption uses the same key for encryption and decryption and asymmetric encryption uses different keys” (Wagner, 2019). Because the same key is used in symmetric, the use case would be for encrypting for yourself or within an organization. Asymmetric encryption is for outside users sending data, and because the keys are separate the algorithm is slower and much more complex. The AES ciphering process uses round keys to encrypt the password. From*tutorialspoint*, the password goes through 14 rounds of randomization where operations are performed on each round key, including byte substitution, row shifting, and column mixing. For centuries, governments and militaries have been using cryptography to send secret messages. From simple ciphers performed on wooden rods to rotar machines. According to *Thales* (2023), “{in 1945} Claude E. Shannon of Bell Labs published an article called ‘A mathematical theory of cryptography.’ It's the starting point of modern cryptography.” Modern cryptography consists of computer encryption, and nowadays keys are shorter and faster than ever with Elliptic-curve cryptography. However, with the advent of cloud services, there is an increasing concern about protecting data both at rest and in transit. Quantum computing poses a significant risk to the future of encryption, with its significant computing power able to break cryptographic algorithms.

My recommendation for a hashing algorithm cipher that avoids collisions is SHA-256. To recommend an algorithm cipher for performing a checksum, I needed to first understand the difference between encryption and hashing. From *NordVPN*, “encryption is the process of scrambling plaintext into unreadable ciphertext, which you can decrypt with a relevant key, while hashing turns plain text into a unique code, which can’t be reverted into a readable form” (Higgins, 2023). This hash is used to compare between two values: the generated hash and the stored hash. If these two values are the same, then the data is validated. Similarly, I also needed to understand what collisions are, which is a vulnerability for when two different keys use the same hash value. This is problematic, as malicious files could be substituted with the originals— with users being none the wiser. With these two pieces of information, I was able to research potential algorithms. MD5 was one potential algorithm, however according to *Baeldung*, “In 2011, Internet Engineering Task Force (IETF) published RFC 6151, describing possible attacks on MD5. Some attacks could generate collisions in less than a minute on an average computer... Thus, the MD5 is no longer recommended for solutions requiring a high level of security” (Stec, 2023). Alternatively, SHA-2 has a 256-bit subvariant that is collision resistant; from *Technopedia*,” SHA-256 was developed by the U.S. Government’s National Security Agency (NSA)” (Lepcha, 2023). SHA-256 is slower than MD5; however, since speed was not specifically stated as a concern, reliability and collision resistance win out.

## Certificate Generation

A screenshot of a computer

Description automatically generated

## 

## Deploy Cipher

A computer screen shot of a program

Description automatically generatedA screenshot of a computer

Description automatically generated

## Secure Communications

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Secondary Testing

A computer screen shot of a program

Description automatically generatedA screenshot of a computer

Description automatically generated

## Functional Testing

A screenshot of a computer

Description automatically generatedA computer screen shot of a program

Description automatically generated

## Summary

## As referenced in the algorithm cipher section, AES is in the domain of cryptography. This is a way of protecting sensitive data, such as certificates and files, in transit. SHA-256 is a hashing algorithm used to secure user input and data integrity. It is a way to ensure the authenticity of a user provided input by comparing the computed hash to that of the stored hash. When data is transmitted over a network, such as through a RESTful web-service using HTTPS, hashing and encryption are used to secure communications. Certificates provide trust and authenticity to a website, in addition to providing encrypted communications between client and server.

## Industry Standard Best Practices

A screenshot of a computer

Description automatically generated

I realized that a significant portion of my vulnerabilities were due to using outdated libraries. So I decided to use to use a newer version of Java, Spring Boot, and OWASP Dependency Check. The latest version of spring boot 3.2.0 requires at least Java 17, therefore I could no longer use Java 12. I changed my JRE to 21 in the build properties, and then updated the pom.xml file accordingly. The remaining vulnerabilities I suppressed. [CVE-2023-35116](https://nvd.nist.gov/vuln/detail/CVE-2023-35116) was advised by the vendor to be not a valid vulnerability due to not being accessible by an external attacker. The other is [CVE-2023-6378](https://github.com/advisories/GHSA-vmq6-5m68-f53m), a serialization vulnerability from a Denial-Of-Service attack by poisoned data. It is exploitable by the logback receiver component. I do not believe that I am using the ch.qos.logback:logback-classic or ch.qos.logback:logback-core packages in my project. It is important to not only use the latest libraries, but to research and understand why a vulnerability might exist. For a financial company dealing in extremely sensitive data, it is crucial to not only to design and architect “good code,” but it’s also important to understand what your code is being implemented upon. Just like how programming used to be at the bit level in assembly, with the advent of AI abstraction is only becoming greater— and it will be more important than ever for developers to understand what is being abstracted away underneath.

**References**

Boyini, K. (2020, June 26). *Convert byte array to hex string in Java*. Tutorialspoint. <https://www.tutorialspoint.com/convert-byte-array-to-hex-string-in-java>

Bernstein, C., & Cobb, M. (2021, September 24). *What is the Advanced Encryption Standard (AES)? Definition from SearchSecurity*. Security. <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard>

Ellis, S. (2018, December 13). *The beautiful hash algorithm*. Medium. <https://steviecellis.medium.com/the-beautiful-hash-algorithm-f18d9d2b84fb>

Higgins, M. (2023, July 12). *Hashing vs. encryption: What is the difference?* NordVPN. <https://nordvpn.com/blog/hashing-vs-encryption/>

Hungund, B. (2022, April 29). *SHA-256 Hash in Java.* GeeksforGeeks. <https://www.geeksforgeeks.org/sha-256-hash-in-java/>

Kiteworks. *Everything you need to know about AES-256 encryption.* <https://www.kiteworks.com/risk-compliance-glossary/aes-256-encryption/#:~:text=AES%2D256%20encryption%20is%20extremely,operating%20in%20highly%20regulated%20industries>.

Lepcha, M. (2023, July 26). *What is SHA-256? How it works in Blockchain & Cryptography -* *Techopedia.* Techopedia. <https://www.techopedia.com/definition/sha-256>

Stec, A. (2023, June 8*). MD5 vs. SHA Algorithms. Baeldung on Computer Science.* <https://www.baeldung.com/cs/md5-vs-sha-algorithms#:~:text=First%20of%20all%2C%20MD5%20produces,in%20terms%20of%20collision%20resistance>.

Thales Group. (2023, February 1). *A brief history of encryption (and cryptography).* <https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption>

tutorialspoint. *Advanced encryption standard.* <https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm>

Wagner, L. (2019, June 11). *(Very) basic intro to AES-256 cipher*. Medium. <https://medium.com/coinmonks/very-basic-intro-to-aes-256-cipher-a60104847776#:~:text=AES%2D256%20Secret%20Key&amp;text=In%20order%20to%20use%20a,attempting%20to%20decrypt%20the%20message>