****

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[CS 305 Project Two 1](#_Toc101116951)

[Document Revision History 3](#_Toc101116952)

[Client 3](#_Toc101116953)

[Developer 3](#_Toc101116954)

[1. Algorithm Cipher 3](#_Toc101116955)

[2. Certificate Generation 6](#_Toc101116956)

[3. Deploy Cipher 6](#_Toc101116957)

[4. Secure Communications 7](#_Toc101116958)

[5. Secondary Testing 7](#_Toc101116959)

[6. Functional Testing 9](#_Toc101116960)

[7. Summary 9](#_Toc101116961)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **4/17/2022** | **Mark Holden** |  |

## Client



## Developer

Mark Holden

## 1. Algorithm Cipher

The Encryption Algorithm that I recommend that Artemis Financial use to encrypt data at rest is the AES 256 Encryption. AES stands for Advanced Encryption Standard, and according to the National Institute of Standards and Technology (NIST) (2021), it was developed from the Rijndael encryption technique, which was selected out of a pool of submissions for the AES. After years of intense scrutiny, the AES remains secure and is still the standard for United States Government. Another advantage of AES is that according to the NIST, one of the requirements of a submission to the AES selection process is that submissions must be royalty-free, so there is no cost to use the encryption algorithm itself.

One of the most basic rules of cryptography is to not create your own algorithm. As Schneier (2011) says, “Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can’t break” (para. 5). A home-grown algorithm might seem secure to you when you create it, but even for an expert in cryptography, one should not be fooled by one’s own hubris to think that an attacker with the skill and the will to defeat it does not exist. Secure cryptographic systems have been developed by teams of skilled cryptographers over decades, and the ones that have stood the test of time can be trusted with financial records.

The currently known risks of this recommendation are the attacks that target gaining access to the key and attacks that use unauthenticated or unauthorized requests for which the server allows access and decrypts the data. Attacks that target access to the key are a risk because if the key is known, then any data encrypted can be deciphered with the key. These attacks could be anything from a phishing scam to gain access to a machine with access to the keys, or a physical breach that involves bribery or social engineering to gain access to such systems. Attacks that try to use unauthenticated or unauthorized calls will target weak security on web requests to bypass the encryption and let the server which has access to the encryption key do the work of decrypting the data on behalf of the attacker. Currently, there is no known way to decrypt data encrypted with the AES 256 algorithm without the key, but that does not guarantee that in the future some advance in technology will not make it possible.

There are still attack vectors to consider when using AES 256. Because the encryption has not been broken, attackers will most likely target the more vulnerable parts of the application rather than to try to decrypt the data directly. That means that it is of paramount importance to protect the encryption key from being exposed. To that end, the encryption key should not be stored with the data which would make it an easy target. The key should be stored securely in an encryption key manager which requires specific authorization to access. According to Townsend Security (n.d.), the data encryption key is stored encrypted by a Key Encryption Key (KEK) and is only decrypted and sent to the application to decrypt data after a call from an authenticated user with proper authorization requests access to encrypted data (section How Encryption Key Systems Work, para. 2).

This recommendation meets the current government regulations set forth by complying with the definition of encryption given in federal regulations and the implementation requirements. Standards for Safeguarding Customer Information (2002) gives the following definition for encryption:

Encryption means the transformation of data into a form that results in a low probability of assigning meaning without the use of a protective process or key, consistent with current cryptographic standards and accompanied by appropriate safeguards for cryptographic key material. (16 CFR 314.2(f))

Provided that the solution also complies with the mandate that the safeguards “Protect by encryption all customer information held or transmitted by you both in transit over external networks and at rest” then the requirements for encryption for data at rest (16 CFR 314.4(c)(3)).

The implementation of the cipher should be on a secure system that end users do not have direct access to, and care should be taken to ensure that keys are maintained in a secure key vault and unauthorized access is not granted via other interfaces to the system that could allow attackers to retrieve unencrypted (or encrypted) data. The best cipher for a financial company to use is one that is both convenient to use and sufficiently secure to prevent unauthorized decryption. The only truly unbreakable encryption is known as the One-Time Pad. The One-Time Pad uses an encryption key made of random noise and the XOR function to encrypt and decrypt data. The problem with the one-time pad according to Rijmenants (2022), is that the key must be the same length the message or longer, each key can only be used a single time, and the key must be truly random. For Artemis Financial, the burden of generating, distributing, and maintaining all the one-time use keys would be a nightmare. Ultimately, the added security of using a one-time pad and having truly unbreakable encryption for a financial services company is not worth the added overhead. That extra overhead is the main reason to not use the “most secure” cipher.

The purpose of a hash function in cryptography is to map an input of any size to a fixed length of bits. There are one-way hashes that can only be used to hash data, but there is no way to recover the original (such as password hashing). According to Alwen (2017), bit levels of a cipher is “the order of magnitude of the amount of resources needed to break a crypto primitives’ security” (para 1). Alwen also states that “the best symmetric key primitives (e.g. AES) the key length is essentially the same as the bits of security provided by the primitive” (Key Length section, para. 1). Because of such a high number of resources required to break the AES encryption, it is a good choice for Artemis Financial to use to encrypt data at rest.

“Random numbers”, or in what are sometimes referred to as cryptographically secure pseudo-random numbers, are used to generate keys. Symmetric keys are a type of key where the same key used to encrypt the data is the key used to decrypt it. This is the case in the AES algorithm. Asymmetric keys are used in cases where one key is used to encrypt data, but another, private key is used to decrypt the data. This way the public key can be shared with anyone and can be used to encrypt data, but only the holder of the private key can decrypt the data.

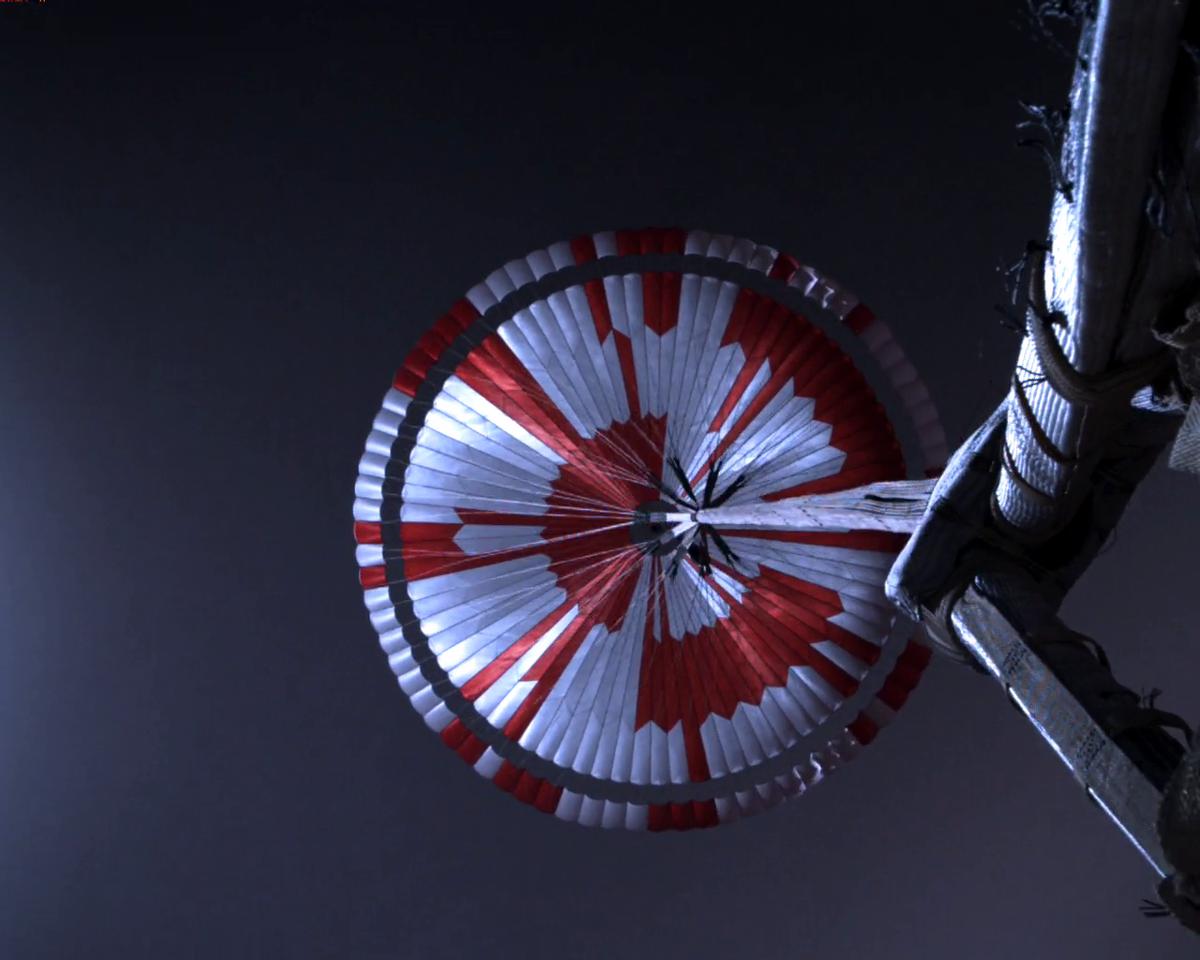
Encryption has been used from antiquity through ubiquity in modern cryptographic computer systems. As technology has improved, earlier methods of encrypting data have been broken, but the earlier methods are still sometimes used for fun or artistic expression, such as the parachute message from the NASA Perseverance rover parachute, which is a simple binary to decimal conversion followed by a decimal to corresponding letter of the alphabet. Without knowing the algorithm, the code in colored parachute gores may seem secure, but in practice it was basically cracked between the time the parachute opened and when the rover touched down on the surface of the red planet. From the simple systems used in ancient times to the more widely known example of the Enigma machine used by the Nazis in World War II, which was eventually cracked by Alan Turing and a team of cryptographers working for Allied intelligence agencies. According to Thales (pronounced tah-les) (2022), the starting point for modern cryptography was in 1945, when “Claude E. Shannon of Bell Labs published an article called “A mathematical theory of cryptography””. Thales also says that “Scientists cannot guarantee encryption beyond thirty years” and that “Experts foresee that RSA2048 can be broken by 2035”, so perhaps within the next 15 years Artemis Financial will need to upgrade their data at rest encryption.

Image Credit: NASA

## 2. Certificate Generation

Text

Description automatically generatedArtemis Financial Certificate:

## 3. Deploy Cipher

Cipher Deployed:

Graphical user interface, application

Description automatically generated

## 4. Secure Communications

Attempt to use HTTP:

Graphical user interface, text, application

Description automatically generated

HTTPS working: Graphical user interface, application

Description automatically generated

## 5. Secondary Testing

Refactored Code:

Pom file:

Text

Description automatically generated

Updating spring-boot-starter-parent to 2.6.6 got rid of all Critical and High severity dependency alerts. The only remaining issues are two medium severity ones.

SslServerApplication.java:

Text

Description automatically generated

Text

Description automatically generated

## 6. Functional Testing

Upon refactoring the above SslServerApplication.java, there did not appear to be any syntactical, logical, or security vulnerabilities left in the program. The refactored code executed without errors.

Text

Description automatically generated

## 7. Summary

Input Validation

Currently, the GET methods on the ServerController do not accept input, so there is no input validation. When POST and PUT methods are added to Create and Update data, input validation should be added to ensure data is in an acceptable format and not malicious in nature.

APIs

The current codebase does not require authentication to call any of the endpoints. An authorization server should be developed, and strict authorization policies should be added to the controllers, and then overridden if necessary to allow additional user types to access necessary resources.

Cryptography

In the /hash endpoint, there is an example of the use of cryptography, but this will need to be taken further by the authorization service. The authorization service must securely salt and hash passwords, as well as encrypt JSON Web Tokens (JWT) to be used by services to authenticate users.

Encapsulation

At the moment, all of the functionality of the application is in the controller. Business logic should be moved to services leaving the controller with only the responsibility of Authenticating and Authorizing a user, passing the request to a service, and returning the response back to the user.

Additional Security Process

The recommended process to add additional layers of security to the application would be to:

1. Add an authentication service to grant bearer tokens to users, who are subsequently Authorized to only access specific parts of the application.
2. The authentication service should securely handle user passwords and encrypt tokens so they cannot be imitated or modified once granted.
3. Create role-based authorization policies and apply them strictly to the controllers.
4. Enhance the role-based authorization policy with attribute-based authorization to allow more granular control of user access.
5. Review security related code periodically to ensure it still adheres to current best practices.

The value of following this security process is that it will take the codebase from the current state that it is in and transform it into a secure system that customers can rely on.

**References**

Alwen, J. (2017, June 13). *The bit-security of cryptographic primitives.* Wickr. https://wickr.com/the-bit-security-of-cryptographic-primitives-2/

National Institute of Standards and Technology. (2021, August 23). *Cryptographic standards and guidelines.* CSRC. <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development>

Rijmenants, D. (2022, March 22). *The one-time pad.* One-time-pad. Retrieved March 27, 2022, from <https://www.ciphermachinesandcryptology.com/en/onetimepad.htm>

Schneier, B. (2011, April 15). *"Schneier's Law".* Schneier on security. <https://www.schneier.com/blog/archives/2011/04/schneiers_law.html>

Standards for Safeguarding Customer Information, 16 C.F.R. § 314 (2002).

Thales. (2022, March 24). *A brief history of encryption.* Thales Group. <https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption>

Townsend Security Inc. (n.d.). *The definitive guide to encryption key management fundamentals.* Townsend Security. <https://info.townsendsecurity.com/definitive-guide-to-encryption-key-management-fundamentals>