**Algorithm Ciphers**

Justin Starr

Department of STEM

CS 305 – Software Security

Dr. Sarah North

September 24, 2023

**Algorithm Ciphers**

Artemis Financial is a consulting company that develops individualized financial plans for its customers, which include savings, retirement, investment, and insurance plans. Artemis Financial is looking to archive its files as a part of its continued growth. The following is a short analysis of the recommended encryption algorithm cipher that Artimus Financial should deploy, and the justification for such deployment.

The recommended encryption algorithm cipher for Artimus Financial is AES 128-bit with CRT (short for counter) mode. Because Artimus Financial stores sensitive information as it relates to their customers, they are highly susceptible to attackers whose aim is to extract such sensitive information (Manico & Detlefsen, 2014, Chapter 6). Security best practices to defend against various types of security attacks are taken into consideration. Such considerations include protecting data stored at rest as well as data in transit. While the goal here is to protect an archive of data that is stored at rest, we must consider the use of data in transit because one way we can help to protect the data is to keep it stored separately from the keys used to encrypt and decrypt the stored data (Manico & Detlefsen, 2014, Chapter 6). To secure data in transit, we can make use of known cryptographic protocols such as Secure Sockets Layer (SSL) and Transport Layer Security (TLS); always being sure to use the latest secure protocols (Manico & Detlefsen, 2014, Chapter 6). Also, because impersonation is a risk, we want to ensure that certificate verification is performed which verifies that the server’s hostname matches the hostname on the certificates and that the certificates have not expired (Manico & Detlefsen, 2014, Chapter 6). Knowing that data at rest must be protected with encryption, this encryption must be reversible; the data should be wrapped or encrypted when it is not being used and when we need to access the data, we must be able to unwrap or decrypt the data (Manico & Detlefsen, 2014, Chapter 6). While a widely adopted standard, such as TLS exists for data in transit, one does not yet exist for data at rest, which necessitates the need for developers to implement encryption on their own (Manico & Detlefsen, 2014, Chapter 6). This is typically done by making use of toolkits, such as Keyczar, that enable developers to easily encrypt and sign data, and manage the keys associated with that encrypted data (Manico & Detlefsen, 2014, Chapter 6).

When considering the most current government regulations and how they will be met, it is important to realize that the reason why government regulations apply to Artimus Financial is because they store sensitive data that pertains to consumers. Consumer data must be protected to every extent possible because it places consumers at higher levels of risk for identity theft and privacy concerns. Specifically, the government passed the Gramm-Leach-Bliley Act in 1999, which requires financial institutions to implement safeguards for the protection of consumer information (Federal Trade Commission, 2021). Since then, updates have been made to these requirements, ensuring a stricter set of standards are applied to companies that store consumer’s sensitive data (Federal Trade Commission, 2021). To help ensure these regulations are met, we are incorporating the use of secure protocols such as SSL and TSL, which are used during the transmission of consumer data between servers. Additionally, Artimus Financials’ application will make use of the recommended encryption algorithm cipher. This will ensure, to the best of our ability, that consumer information will be protected and only accessible to authorized individuals/services.

The recommended algorithm cipher will be used to encrypt Artimus Financials’ archived data. The AES 128-bit encryption is a type of symmetrical cryptography, which is used for encryption or encrypting data rather than signing (Manico & Detlefsen, 2014, Chapter 6). To encrypt data, information is scrambled with the use of a secret key, and the encrypted data cannot be unscrambled without it (Manico & Detlefsen, 2014, Chapter 6). It is because the same key is used for the encryption and decryption of data that the cryptography is considered symmetrical (Manico & Detlefsen, 2014, Chapter 6).

Another type of cryptography is asymmetric cryptography or public key cryptography. This type of cryptography uses two different keys, a private key which is a secret key used to sign data, and a public key that can be made public and known to anyone and is used to validate signatures (Manico & Detlefsen, 2014, Chapter 6). This type of cryptography is not widely used for the encryption of data, which is one reason why we have chosen not to use an asymmetrical cipher. A second reason is that it is possible for the private key to become compromised or exposed due to certain security vulnerabilities, by trusted system administrators who have access to private keys, or even by a warrant that compels a system administrator to relinquish private keys (Manico & Detlefsen, 2014, Chapter 6).

A cipher that is better than AES 128-bit could be AES 256-bit, which makes the key size larger, providing higher encryption and making the possibility of an attacker decrypting the data even more improbable. When choosing an algorithm cipher, it is important to consider what the cipher is being used for. Various ciphers are used for various reasons, and many have been created or adapted due to previously identified security vulnerabilities. While 256-bit encryption is better because of the larger key size, 128-bit encryption is sufficient for most purposes. One would likely decide to choose the lower encryption because of the amount of computing resources required of systems that implement higher encryption. AES is the algorithm that is most likely to be implemented correctly, is widely adopted, and has been found to be six times faster than the triple DES algorithm (Tutorialspoint, 2023).

An important aspect of cryptography and other security-related operations is the ability to access a system that generates random numbers (Manico & Detlefsen, 2014, Chapter 6). The reason why random numbers are important is because it is essential that a system that creates cryptographic keys can generate completely random numbers. While randomness is somewhat determined by the type of platform that is being used, such as Windows, Linux, or MacOS, Oracle has improved Java so that the SecureRandom class includes a getInstanceStrong() method, “which returns the cryptographically strongest random number generator offered by the platform” (Manico & Detlefsen, 2014, Chapter 6).

As it concerns the current state and history of encryption algorithms, one could argue that encryption is still in its infancy or early stages. In 1975, DES (Data Encryption Standard) became the standard in encryption as a block cipher that utilized a 56-bit key (Axel, 2021). This no longer remains true as AES (Advanced Encryption Standard) is now the most common and widely used algorithm, which was built off the DES algorithm and is significantly more secure (Axel, 2021). AES 256-bit encryption has even become widely accepted by agencies such as the United States National Security Administration (Axel, 2021). While modern encryption does provide the protection that is required by current technology and common methods used to attempt cryptography cracking, the future remains uncertain, especially with the rise of quantum computing (Axel, 2021). If quantum computing continues to grow, today's methods of cryptography will prove to be insufficient. However, the technology has yet to prove its usefulness, but in the event quantum computing powers become standard, quantum-safe encryption algorithms are already being developed leaving only a need for developers to upgrade their products accordingly (Axel, 2021).

**References**

Axel. (2021, May 28). *Encryption: The past, Present, and Future*. AXELorg.

https://www.axel.org/2021/05/28/history-of-encryption/

Federal Trade Commission. (2021, October 27). *FTC Strengthens Security Safeguards for*

*Consumer Financial Information Following Widespread Data Breaches*. <https://www.ftc.gov/news-events/news/press-releases/2021/10/ftc-strengthens-security-safeguards-consumer-financial-information-following-widespread-data>

Manico, J., & Detlefsen, A. (2014). *Iron-Clad Java: Building Secure Web Applications*. McGraw

Hill Computing. <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/?sso_link=yes&sso_link_from=SNHU>

Oracle. (2023, July 10). *API reference for Java Platform, Standard Edition*. Java Security

Standard Algorithm Names. https://docs.oracle.com/en/java/javase/11/docs/specs/security/standard-names.html#cipher-algorithm-names

Tutorialspoint. (2023). *Advanced Encryption Standard*. Online Courses and eBooks Library.

<https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm>