Journal of AHIMA (American Health Information Management Association)

Encryption Basics

Kevin Stine
Information Security Specialist
Security Management and Assurance

Quynh Dang
Computer Scientist
Cryptographic Technology

National Institute of Standards and Technology
Information Technology Laboratory
Computer Security Division

Healthcare and health information technology professionals are entrusted with patient data which, because of its personal nature, requires protection to ensure its confidentiality. To provide this protection, these professionals frequently look to commonly accepted technologies and methodologies to safeguard this data while at rest and in transit. One technology capable of providing this type of protection is encryption.

The Health Insurance Portability and Accountability Act (HIPAA) Security Rule has long identified encryption, an addressable implementation specification, as a mechanism to safeguard electronic protected health information. More recently, the standards and certification criteria for electronic health records (EHR) technology have specified that EHR technology must be able to encrypt and decrypt health information in order to qualify for stage 1 of the Medicare and Medicaid EHR Incentive Program (commonly known as the Meaningful Use program).

Similarly, guidance accompanying the Breach Notification Interim Final Rule, issued by the Department of Health and Human Services' (HHS) Office for Civil Rights (OCR), identified encryption as one technology that can render protected health information "unusable, unreadable, or indecipherable to unauthorized individuals." Protected health information that is encrypted in accordance with this guidance is not subject to breach notification requirements under the Breach Notification Interim Final Rule. The guidance discusses encryption as a mechanism to protect data in transit and data at rest.

Implementing and managing an encryption solution can certainly be complex. However, an understanding of basic encryption processes, an awareness of the security properties provided by encryption, and knowledge of important requirements for effective encryption are essential for anyone entrusted to protect patient data.

Encryption Basics

Encryption, a security control used primarily to provide confidentiality protection for data, is simply a mathematical transformation used to scramble data requiring protection (plaintext) into a form not easily understood by unauthorized people or machines (ciphertext). The plaintext, after being transformed into ciphertext, appears random and does not reveal anything about the content of the original data. Once encrypted, no person (or machine) can discern anything about the content of the original data by reading the encrypted form of the data.

Encryption is a reversible transformation. It is useful only when encrypted data (ciphertext) can be reversed back to its original, unencrypted form (plaintext). If not reversible, the encrypted data is considered unreadable and unusable. This reversal process is referred to as decryption. An encryption process has one corresponding decryption process, which is used to reverse the encrypted data (ciphertext) back to its original content (plaintext).

Each encryption and decryption function requires a cryptographic key. A cryptographic key is a string of binary digits used as an input to encryption and decryption functions. In order for the encryption function to transform the plaintext into ciphertext, and for the decryption function to reverse the ciphertext back to its original form, the encryption and decryption functions must use the same cryptographic key. This is referred to as symmetric key. The encryption functions specified in the Advanced Encryption Standard (AES) are widely supported in current systems and software.

As depicted in figure 1, the encryption function requires two inputs, *Plaintext* and *Cryptographic Key*, in order to output *Ciphertext*.

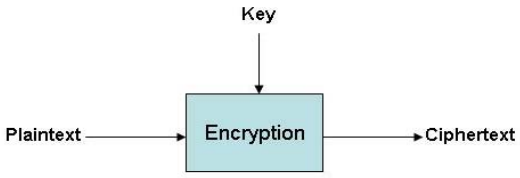

Figure 1. Encryption Function

As depicted in figure 2, the decryption function requires two inputs, *Ciphertext* and *Cryptographic Key*, in order to output *Plaintext*.

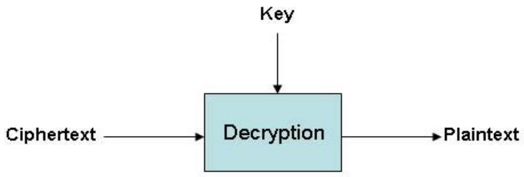

Figure 2. Decryption Function

Encryption in widely used Computer Applications

Encryption is widely used in many computer applications to protect data in transit and at rest. User involvement in the encryption process may vary for each application. For example, in some applications of secure web browsing using Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols, the use of encryption may be transparent to the user. In other implementations, the user may be required to enter a password in order to encrypt or decrypt the protected data if the cryptographic key is derived from the password.

Using Secure/Multipurpose Internet Mail Extensions (S/MIME), email messages can be encrypted by the sender and then decrypted by the intended recipient. These email messages can only be read by the sender and the intended recipient(s).

Internet Protocol Security (IPsec), network layer security protocols frequently used to establish virtual private networks (VPNs), enables only the two ends of a communication in a computer network to understand the encrypted messages exchanged between them.

The SSL protocol and its successor, TLS, are the primary end-to-end security protocols used to protect information traversing the Internet. The most common usage scenario for these protocols is a web browser, acting as client for the human user, interacting with a web server. Using SSL and TLS, encrypted messages between a web browser and a web server cannot be decrypted by any unauthorized party.

In addition to protecting data being transmitted over computer networks, encryption is also used to protect data at rest such as data stored on hard drives, USB drives, and other end user storage devices. When an encrypted hard drive, for example, is stolen or is in possession of an unauthorized user, the encrypted data in the hard drive is useless to the unauthorized user because the unauthorized user cannot reproduce the plaintext from the hard drive without the cryptographic key.

Important Requirements when Implementing Encryption

Encryption is an important security control to provide confidentiality protection for data. For encryption to be effective and to provide data confidentiality, it is important for the following requirements to be met.

Keep the Cryptographic Key Secret

Encryption algorithms are made public to allow for interoperability, ease of use, and more open and effective analysis. The security of the encryption depends on the secrecy of the cryptographic key. The cryptographic key must be kept secret from all entities who are not allowed to see the plaintext. Any person or machine that knows the cryptographic key can use the decryption function to decrypt the ciphertext, resulting in exposure of the plaintext. If a strong cryptographic key is generated, but is not kept secret, then the data is no longer protected. Keeping the cryptographic key secret ensures confidentiality protection of the protected data.

Protect the Cryptographic Key from Modification

The cryptographic key must always be protected from modification. For the ciphertext to be transformed to plaintext, the decryption function must use the same cryptographic key used by the encryption function to decrypt the ciphertext. If the cryptographic key is modified, the plaintext cannot be reproduced. When this happens, the plaintext (the protected data) is lost. It is very important to protect the cryptographic key from any modification (including being lost).

Like other files, cryptographic keys could be intentionally or unintentionally modified. For example, cryptographic keys could be unintentionally corrupted during transmission if an application or protocol using the cryptographic key does not operate as expected. A malicious user with access to the cryptographic key could intentionally modify the cryptographic key to prevent access to encrypted data. In either situation, plaintext data can not be reproduced by the modified cryptographic key.

Therefore, any system using encryption should have a key recovery mechanism to recover the cryptographic key if it is lost or modified. An example of such a recovery mechanism is to make multiple copies of the cryptographic key, and store them separate from each other in locations unknown to unauthorized parties. If the original key is modified or lost, a recovery copy can be used.

Importance of Cryptographic Key Length in Choosing an Encryption Algorithm

To decrypt the ciphertext, an attacker must search for the cryptographic key by decrypting the ciphertext with all possible keys until the correct key is found. For example, if the cryptographic key is two bits long, there are four possible keys that the attacker may try (00, 01, 10, 11). The longer the key, the more possible keys the attacker must try. Generally speaking, an encryption algorithm that uses a longer key provides greater level of confidentiality protection. For example, the Advanced Encryption Standard (AES) using a 192-bit key (AES-192) provides stronger protection than AES using a 128-bit key (AES-128) because there are more possible values for a 192-bit key than for a 128-bit key.

Generate a Strong Cryptographic Key and Transport it Securely

Generating a strong cryptographic key is very important when using encryption. If an attacker can get information about certain bits of the key, then the encryption function using this key does not provide the necessary level of protection. For example, if the key is two bits, and the attacker knows that the first bit of the key is equal to the second bit then the attacker only needs to try two possible keys, 00 and 11, instead of four combinations as explained above.

Ideally, a cryptographic key is a randomly generated string of bits that provides the attacker with no information about any bits of the key. Keys can be generated using a Deterministic Random Bit Generator (DRBG), a function used to generate high quality random bits for an encryption key. NIST Special Publication 800-90, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators,* recommends NIST-approved mechanisms for the generation of random bits using deterministic methods.

Cryptographic keys can be generated solely by the encrypting entity, or through cooperation between the encrypting and decrypting entities, depending on the usage scenario. NIST Special Publication 800-57, *Recommendation for Key Management - Part 1*, discusses approved cryptographic key generation methods when the key is generated solely by the encrypting party. In many secure communication protocols (ex, TLS), the cryptographic key may be generated through cooperation of the encrypting and decrypting entities. NIST Special Publication 800-57 (Parts 1 and 2) provides guidelines on these key agreement schemes.

In an application where the encrypting entity needs to share the key with a separate decrypting entity, the key must be transported to the decrypting entity in a secure manner.

This transportation can be done physically using an electronic device such as a USB drive which holds the cryptographic key. This transportation can also be done electronically over a computer network. NIST Special Publication 800-57 (Parts 1 and 2) provides guidelines on methods of secure key transport.

Encrypt All Copies of the Data

All data that requires confidentiality protection should be encrypted if there is a possibility that an unauthorized person could access it. Data at rest in an operational environment is frequently encrypted. However, all copies of data, including data in storage and backup environments, should also be encrypted to provide comparable protection.

Transition to NIST-approved encryption functions

Over time, changes in the use of encryption may be necessary because of cryptographic attacks on encryption algorithms or the availability of more powerful computing techniques and/or devices. Data encrypted in the past using a non NIST-approved encryption algorithm, or a NIST-approved encryption algorithm that has become obsolete, should be encrypted using a current NIST-approved encryption algorithm to ensure a strong level of protection for the data. NIST Special Publication 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, identifies current approved algorithms and timelines for acceptable use.

NIST Encryption Standards and Guidelines

For more information on encryption processes and technologies, including the following standards and guidelines developed by NIST, please refer to the NIST Computer Security Resource Center at http://csrc.nist.gov.

- Federal Information Processing Standard (FIPS) 197, *Advanced Encryption Standard*
- FIPS 140-2, *Security Requirements for Cryptographic Modules*
- NIST Special Publication (SP) 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*
- NIST SP 800-113, *Guide to SSL VPNs*
- NIST SP 800-111, *Guide to Storage Encryption Technologies for End User Devices*
- NIST SP 800-90, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*
- NIST SP 800-77, *Guide to IPsec VPNs*
- NIST SP 800-57, *Recommendation for Key Management, Part 1: General*
- NIST SP 800-57, *Recommendation for Key Management, Part 2: Best Practices for Key*

Management Organization

- NIST SP 800-57, *Recommendation for Key Management, Part 3: Application-Specific Key Management Guidance*

- NIST SP 800-52, *Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations*

NIST Overview

The National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Department of Commerce. NIST's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.