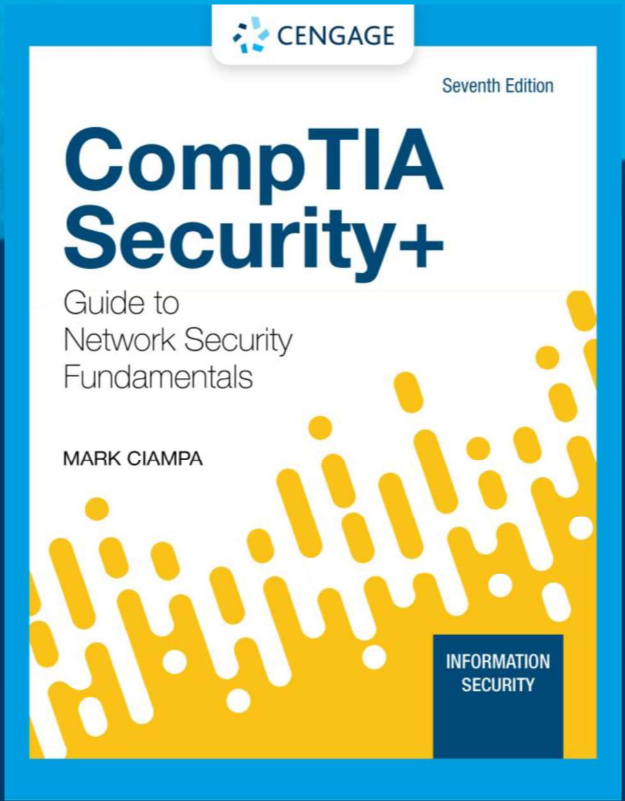

CompTIA Security+ Guide to Network Security Fundamentals, 7th Edition

Module 6: Basic Cryptography

CENGAGE

Module Objectives

By the end of this module, you should be able to:

1. Define cryptography
2. Describe hash, symmetric, and asymmetric cryptographic algorithms
3. Explain different cryptographic attacks
4. List the various ways in which cryptography is used

Defining Cryptography

- Defining cryptography involves understanding what it is and how it is used
- It also involves knowing the limitations of cryptography

What is Cryptography? (1 of 5)

- **Cryptography**
  - Scrambling information so it cannot be read
  - Transforms information into secure form so unauthorized persons cannot access it
- **Steganography**
  - Hides the existence of data
  - An image, audio, or video file can contain hidden messages embedded in the file
  - Achieved by dividing data and hiding in unused portions of the file
  - May hide data in the file header fields that describe the file, between sections of the *metadata* (data used to describe the content or structure of the actual data)

What is Cryptography? (2 of 5)

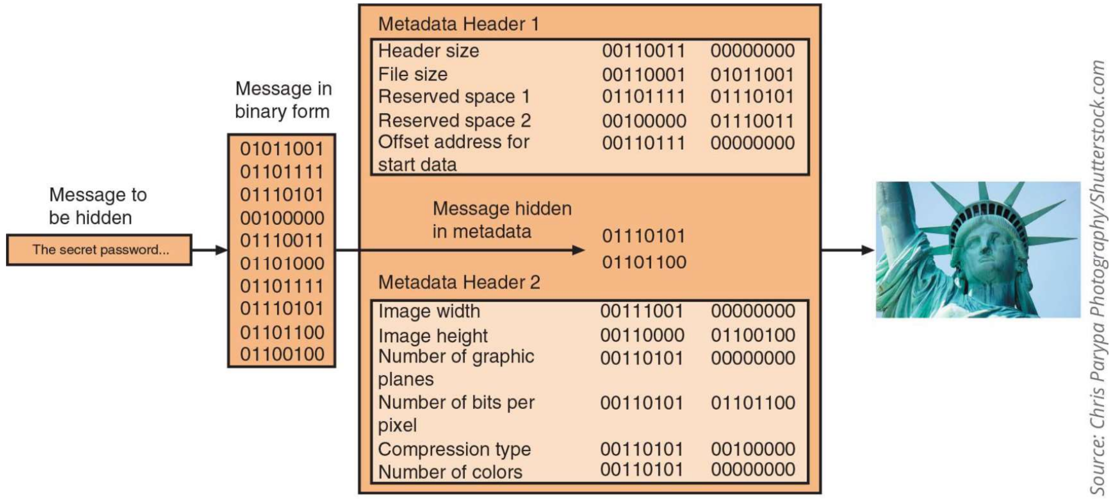

Figure 6-1 Data hidden by steganography

Figure 6-1 Data hidden by steganography

What is Cryptography? (3 of 5)

- **Encryption** is the process of changing original text into a secret message using cryptography
- Changing the secret message back to its original form is known as **decryption**
- *Plaintext* is unencrypted data to be encrypted or is the output of decryption
- *Ciphertext* is the scrambled and unreadable output of encryption
- *Cleartext* data is data stored or transmitted without encryption
- Plaintext data is input into a **cryptographic algorithm** (also called a *cipher*)
  - It consists of procedures based on a mathematical formula used to encrypt and decrypt the data

What is Cryptography? (4 of 5)

- A key is a mathematical value entered into the algorithm to produce ciphertext
  - The reverse process uses the key to decrypt the message
- A *substitution cipher* substitutes one character for another
  - One type is a ROT13, in which the entire alphabet is rotated 13 steps (A=N)
- An *XOR cipher* is based on the binary operation eXclusive OR that compares two bits
  - If the bits are different, a 1 is returned, if they are identical, a 0 is returned

What is Cryptography? (5 of 5)

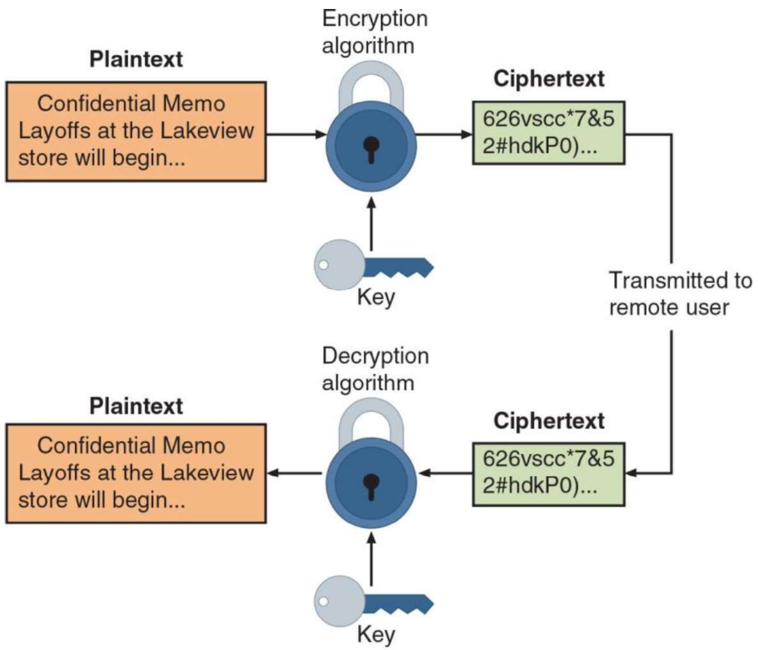

Figure 6-2 Cryptographic process

Figure 6-2 Cryptographic process

Cryptography Use Cases (1 of 2)

- Cryptography can provide several basic protections
  - *Confidentiality* ensures only authorized parties can view it
  - *Integrity* ensures information is correct and unaltered
  - *Authentication* ensures sender can be verified through cryptography
  - *Nonrepudiation* proves that a user performed an action
  - *Obfuscation is* making something obscure or unclear
- *Security through obscurity*
  - An approach in security where virtually any system can be made secure as long as outsiders are unaware of it or how it functions

Cryptography Use Cases (2 of 2)

- Cryptography can provide protection to data as that data resides in any of three states:
  - *Data in processing* (also called *data in use*) is data actions being performed by “endpoint devices”
  - *Data in transit* are actions that transmit the data across a network
  - *Data at rest* is data that is stored on electronic media

Limitations of Cryptography (1 of 2)

- The number of small electronic devices (**low-power devices**) has grown significantly
  - These devices need to be protected from threat actors
- Applications that require extremely fast response times also face cryptography limitations
- **Resource vs. security constraint** is a limitation in providing strong cryptography due to the tug-of-war between available resources (time and energy) and the security provided by cryptography
- It is important that there be **high resiliency** in cryptography
  - High resiliency is the ability to quickly recover from these resource vs. security constraints

CENGAGE

Limitations of Cryptography (2 of 2)

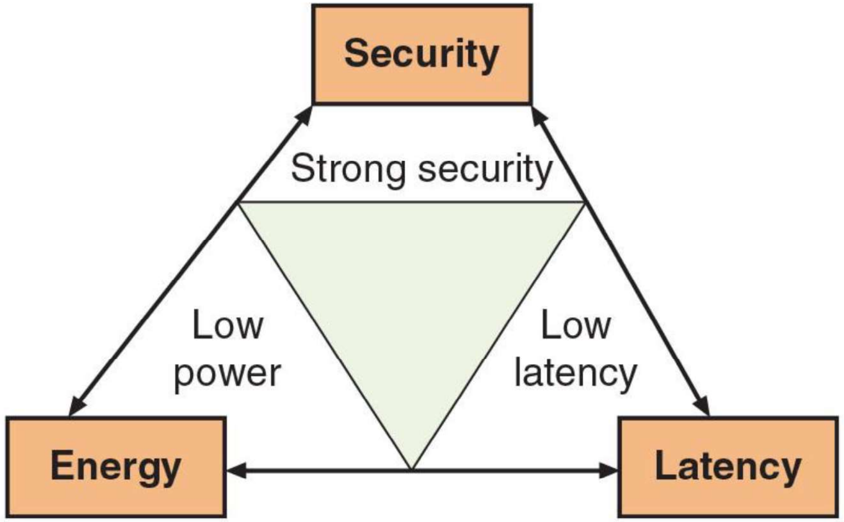

Figure 6-3 Resource vs. security constraint

Figure 6-3 Resource vs. security constraint

Knowledge Check Activity 1

Which of the following is a term that proves that a user performed an action with a computer or on data?

a. Confidentiality

b. Nonrepudiation

c. Obfuscation

d. Authentication

Knowledge Check Activity 1: Answer

Which of the following is a term that proves that a user performed an action with a computer or on data?

Answer: b. Nonrepudiation

Repudiation means denial. Nonrepudiation is the inability to deny, so in information technology, nonrepudiation is the process of pricing that a user performed an action such as creating a file or sending an email.

Cryptographic Algorithms

- A fundamental difference in cryptographic algorithms is the amount of data processed at a time
  - **Stream cipher** - takes one character and replaces it with another
  - **Block cipher** - manipulates an entire block of plaintext at one time
  - **Sponge function** - takes as input a string of any length and returns a string of any requested variable length
- Three categories of cryptographic algorithms
  - Hash algorithms
  - Symmetric cryptographic algorithms
  - Asymmetric cryptographic algorithms

Hash Algorithms (1 of 3)

- Hash algorithm creates a unique “digital fingerprint” of a set of data and is commonly called *hashing*
  - This fingerprint, called a digest (sometimes called a *message digest* or *hash*), represents the contents
  - Is primarily used for comparison purposes
- Hashing is intended to be one way in that its digest cannot be reversed to reveal the original set of data
- Secure hashing algorithm characteristics:
  - *Fixed size* - short and long data sets have the same size hash
  - *Unique* - two different data sets cannot produce the same hash
  - *Original* - data set cannot be created to have a predefined hash
  - *Secure* - resulting hash cannot be reversed to determine original plaintext

Hash Algorithms (2 of 3)

Image Name	Torrent	Version	Size	SHA256Sum
Kali Linux 64-Bit (Installer)	Torrent	2020.2	3.6G	ae9a3b6a1e016cd464ca31ef5055506cecfc55a10f61bf1acb8313eddbe12ad7
Kali Linux 64-Bit (Live)	Torrent	2020.2	2.9G	e90e0cfb4bc8fc640219dba66c9fe4308c9502164e432c47a30af50ce9cb3ba2
Kali Linux 64-Bit (NetInstaller)	Torrent	2020.2	420M	def160159e12ff52fb5f4991240bd760500d7cd5ee38601a8bf35809a20f9450

Source: Kali Linux

Figure 6-4 Verifying downloads with digests

Figure 6-4 Verifying downloads with digests

Hash Algorithms (3 of 3)

- *Message Digest (MD)* is one of the earliest family of hash algorithms
  - Most well-known of the MD hash algorithms is MD5
  - Some security experts recommend using a more secure hash algorithm
- *Secure Hash Algorithm (SHA)*
  - SHA-2 is currently considered to be a secure hash
  - SHA-3 was announced as a new standard in 2015 and may be suitable for low-power devices
- *Race Integrity Primitives Evaluation Message Digest (RIPEMD)*
  - The primary design feature is two different and independent parallel chains of computation, the results are combined at end of process
  - There are several version of RIPEMD
    - RIPEMD-128, RIPEMD-256, and RIPEMD-320

Symmetric Cryptographic Algorithms (1 of 2)

- **Symmetric cryptographic algorithms** use the same single key to encrypt and decrypt a document
  - Original cryptographic algorithms were symmetric
  - Also called *private key cryptography* (the key is kept private between sender and receiver)
- Common algorithms include:
  - *Data Encryption Standard (DES)*
  - *Triple Data Encryption Standard (3DES)*
  - *Advanced Encryption Standard (AES)*
  - *Rivest Cipher (RC)*
  - *Blowfish*

Symmetric Cryptographic Algorithms (2 of 2)

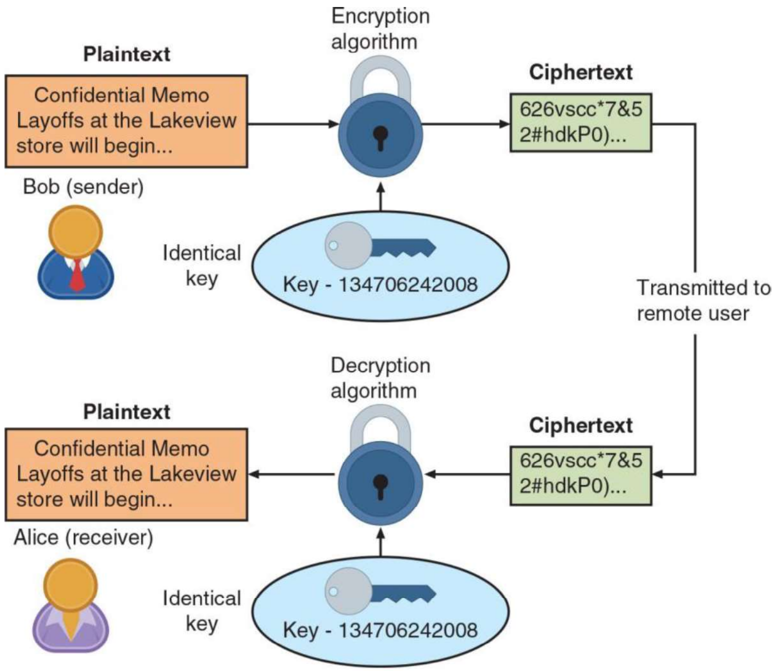

Figure 6-5 Symmetric (private key) cryptography

Figure 6-5 Symmetric (private key) cryptography

Asymmetric Cryptographic Algorithms (1 of 6)

- The primary weakness of symmetric algorithms: distributing and maintaining a secure single key among multiple users distributed geographically poses challenges
- Asymmetric cryptographic algorithms use two mathematically related keys
  - Also known as *public key cryptography*
  - Public key available to everyone and freely distributed
  - Private key known only to individual to whom it belongs
- Important principles
  - *Key pairs*
  - *Public key*
  - *Private key*
  - *Both directions* - keys can work in both directions

Asymmetric Cryptographic Algorithms (2 of 6)

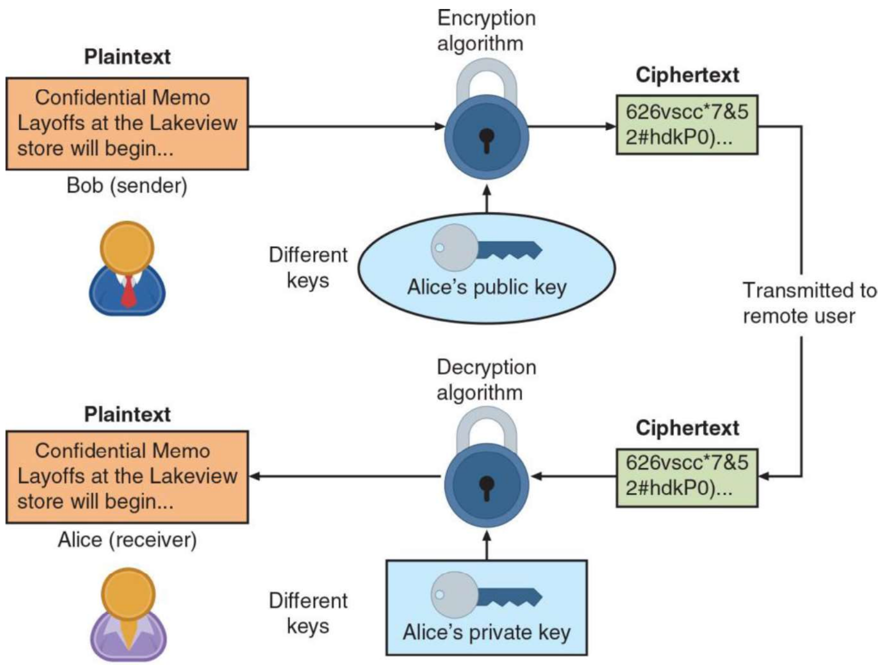

Figure 6-7 Asymmetric (public key) cryptography

Figure 6-7 Asymmetric (public key) cryptography

Asymmetric Cryptographic Algorithms (3 of 6)

- **RSA**
  - Published in 1977
  - Multiplies two large prime numbers
  - The basis of RSA encryption security if factoring
- **Elliptic curve cryptography (ECC)**
  - Users share one elliptic curve and one point on the curve
  - Uses less computing power than prime number-based asymmetric cryptography
    - Key sizes are smaller
  - Considered as an alternative for prime-number-based asymmetric cryptography for mobile and wireless devices

Asymmetric Cryptographic Algorithms (4 of 6)

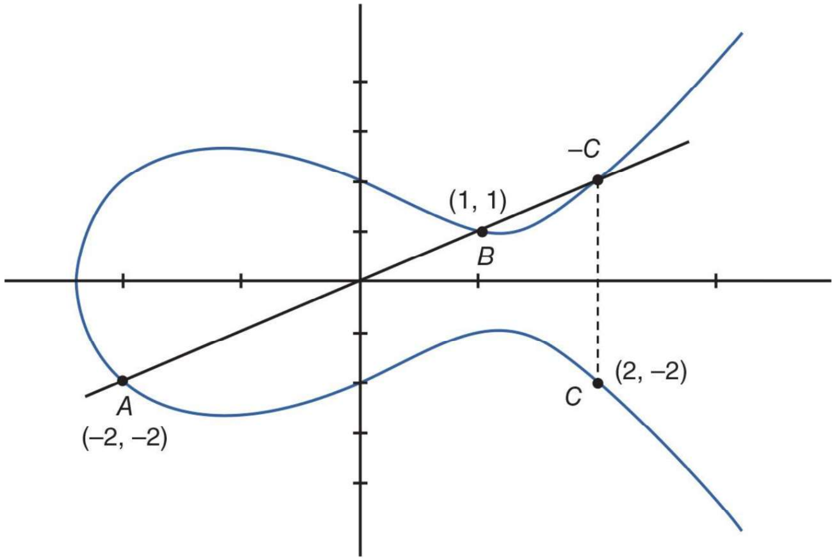

Figure 6-8 Elliptic curve cryptography (ECC)

Figure 6-8 Elliptic curve cryptography (ECC)

Asymmetric Cryptographic Algorithms (5 of 6)

- **Digital Signature Algorithm (DSA)**
  - Creates a digital signature - an electronic verification of the sender
  - A digital signature can:
    - *Verify the sender*
    - *Prevent sender from disowning the message*
    - *Prove message integrity*
- **Key Exchange**
  - There are different solutions for a key exchange that occurs within the normal communications channel (in-band) of cryptography:
    - *Diffie-Hellman (DH)*
    - *Diffie-Hellman Ephemeral (DHE)*
    - *Elliptic Curve Diffie-Hellman (ECDH)*
    - *Perfect forward secrecy*

Asymmetric Cryptographic Algorithms (6 of 6)

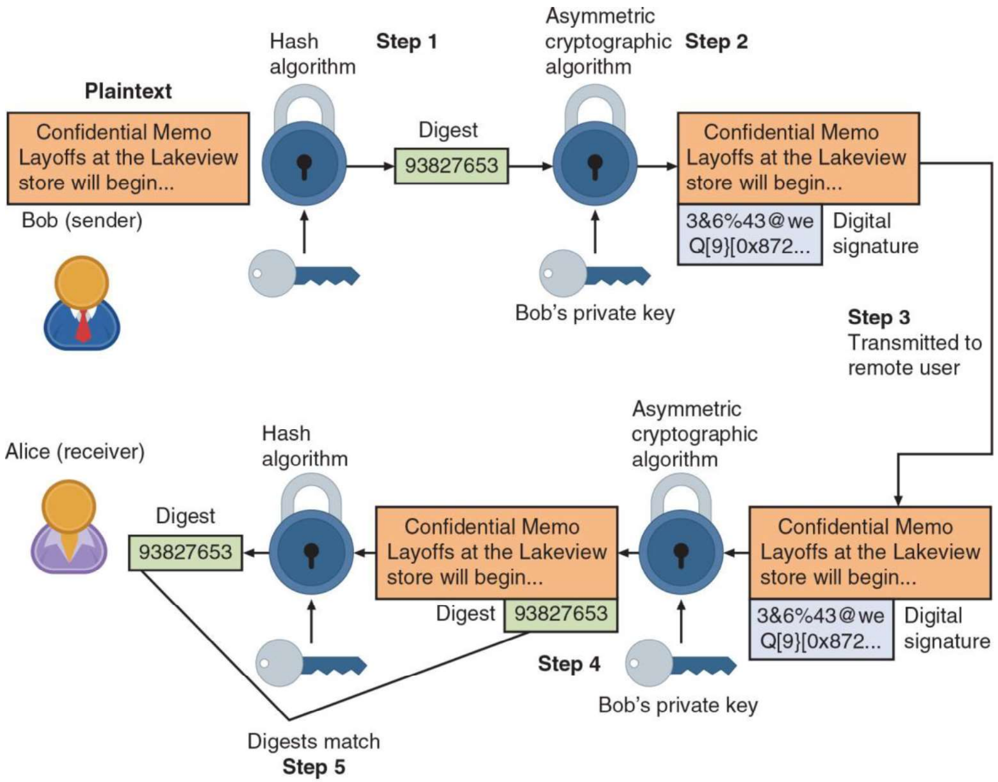

Figure 6-9 Digital signature

Figure 6-9 Digital signature