Linux Security:

Encryption

This material is based on work supported by the National Science Foundation under Grant No. 0802551

National Science Foundation
WHERE DISCOVERIES BEGIN

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author (s) and do not necessarily reflect the views of the National Science Foundation

Lesson Overview

Not all data are equal. Some data can be made available to a large group of people, but some data are very sensitive and must be secured. Governments, militaries, corporations, academic institutions, medical institutions and even regular computer users have many reasons to keep information private and unavailable to others. One way to secure data is to encrypt it. Another method is to encrypt the entire media that contains the data. In this way, any information stored on the media is encrypted by default.

Whether on a flash drive, designated folder or on an entire operating system, file encryption allows users and administrators to protect sensitive data from prying eyes. Linux offers several different options for protecting data with minimum setup.

In this lesson, students will explore basic encryption terminology and techniques. Additionally, students will be shown how to configure remote flash drives using two popular file encryption mechanisms.

Understanding encryption techniques is an essential skill for any computer user who needs to keep data secure.

Select **PLAY** below for a video on the lesson overview:

View Video VideoLesson4Overview(C5L5S2).mp4

Objective

You should know what will be expected of you when you complete this lesson. These expectations are presented as objectives. Objectives are short statements of expectations that tell you what you must be able to do, perform, learn, or adjust after reviewing the lesson.

Lesson Objective:

Given the need to encrypt important data on Linux systems, the student will be able to create whole-disk or filesystem encrypted partitions using DM-Crypt or Loop-AES.

Lesson Outline

In this lesson, you will explore:

- Important Terms
- File Encryption History
- Need for Encryption
- Popular Encryption Tools
- Crypto Strength
- Private & Public Key
- Loop-AES
- DM-Crypt
- Summary

Resources and Notes

This lesson uses Ubuntu Linux for demonstration. To complete this lesson successfully, you must have access to:

- ❖ Ubuntu Linux on bare metal or as a virtual install
- ❖ 10 Gb of hard drive space dedicated to the operating system's use
- ❖ A command shell
- ❖ Internet for research
- ❖ Word processor

Use the resources on the right to configure your system for Ubuntu.

Resources:

- Download Virtualbox
- Using Virtualbox with Ubuntu
- Virtualbox for Linux Hosts
- Virtualbox manual

Glossary of Terms

- **AES – (Advanced Encryption Standard)** - The Advanced Encryption Standard (AES) specifies a cryptographic algorithm (approved by FIPS - Federal Information Processing Standard) that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.
- **Algorithm** – The result of encryption performed on plaintext using a mathematic formula called a cipher.
- **Cipher** - A cipher (pronounced SAI-fuhr) is any method of encrypting text (concealing its readability and meaning). It is also sometimes used to refer to the encrypted text message itself although the term "cyphertext" is preferred in those instances. Basically, a cipher is a way of scrambling content, usually a message or file, so that others cannot read the material.
- **Ciphertext** - text that has been encoded (ciphered) to hide its meaning and contents.
- **Cryptsetup** – An easy-to-use encryption utility that works at the block device level. This means you can mix encrypted and unencrypted partitions on the same drive. It is a great way to protect laptops, sensitive data on workstations, and removable media, such as backup drives and USB drives.

Information Source:

- FIPS & AES
- Cipher text
- Cryptsetup

Glossary of Terms

- **Decryption** - The process used to retrieve readable content out of a file or message by using an algorithm to decode the file.
- **Encryption** - The process of using an algorithm to code a message, file or text in order to "hide" the context of the original text or file. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext.
- **Hash** - also called a digest, is a unique string of data. A hash is created when a collection of information that you want to protect is run through a hash function.
- **Dm-Crypto** - An algorithm method used to cipher a message or file.
- **Key** - The key is used in conjunction with the algorithm to code or decode the file.
- **Loop-AES** - The Loop-AES provides a loadable Linux kernel module that has the AES cipher built in. However, with the recent progress of cryptosetup, dm-crypt, and TrueCrypt, AES is becoming obsolete.
- **Loop device** - Loop devices are block devices that do not store data directly but loop all reads and writes to underlying block device or file, possibly encrypting and decrypting data in the process.

Information Source:

- Hash
- DM-Crypt
- Loop AES
- Loop Device Primer

Glossary of Terms

- **LUKS** (Linux Unified Key Setup) - LUKS is the standard for Linux hard disk encryption. By providing a standard on-disk-format, it facilitates compatibility among distributions and provides secure management of multiple user passwords. In contrast to existing solutions, LUKS stores all necessary setup information in the partition header which enables the user to transport or migrate data seamlessly.
- **PGP** (Pretty Good Privacy) - A popular encryption process used to keep email and message traffic private between sender and recipient.
- **Plaintext** - Human readable unscrambled text. Text that can be read without decoding.
- **Private Key** - An encryption / decryption key known only to the party or parties that exchange data. Also referred to as symmetric cryptology.
- **Public Key** - In cryptography, a public key is a value provided by some designated authority as an encryption key that, combined with a private key derived from the public key, can be used to effectively encrypt messages and digital signatures. Also referred to as asymmetric cryptology.

Information Source:

- LUKS
- Public Key Encryption

History of Encryption

A course on encryption without reviewing the history of encryption is missing an exciting component of the lesson. Encryption has been around for thousands of years and research on the Internet will reveal some exciting information about important developments in history that were directly related to encryption to encryption.

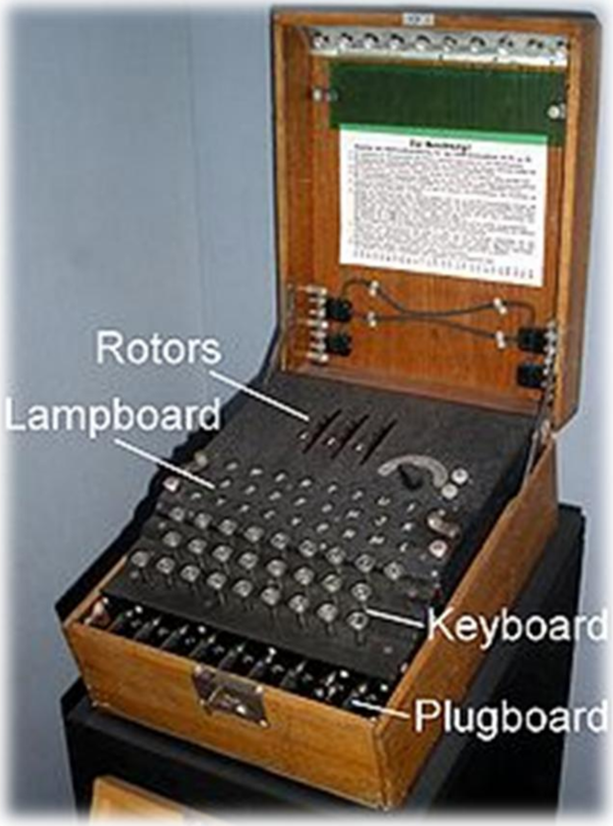
One excellent story is the *Enigma* machine and U-110. This story refers to the German encryption machine called *Enigma* that was discovered during World War II (see picture on next slide). The Allies captured a German U-boat and recovered the Enigma device. This recovery allowed the Allies to retrieve and decrypt important message traffic to U-boat commanders and resulted in a significant security breach to the Germans. Historians believe this was a major contribution to the Allies claiming victory in World War II.

The movie U-571, is fictional and claims the boat was taken by American Naval forces. However, the boat was actually U-110 and was captured by British forces.

Information Source:

- History of Encryption
- Think Quest: Encryption
- Evolution of Encryption

History of Encryption



Picture of Enigma Machine
Source: Wikipedia

Recommended Reading

- Enigma Machine
- How Enigma Worked

What is File Encryption?

File encryption is the process in which data files are run through a mathematical process (algorithm) that conceals or prevents the data from being readable in its regular form. In other words, the original readability of the file is hidden from the user. There are several encryption processes that can be used to conceal data, but most involve using a key in conjunction with an algorithm to scramble the data. File encryption is just one step of additional security towards making your system safe. Other security measures that should also be implemented are proper password policies, limited user access, physical security, backups, maintenance and training. These measures, along with good encryption, work together to protect your data.

Encryption is an additional measure to protect valuable data. Other uses include:

- Protect information stored on your computer from unauthorized access - even from people who otherwise have access to your computer system. (System administrators have access to all files on a computer, unless it is encrypted)
- Protects information while it is in transit from one computer system to another whether via email or through file services. (Data cannot be changed unless someone has the appropriate key.)
- Used to prevent and detect accidental or intentional changes in your data. (The key must be used to open the data file)
- Can be used to verify whether or not the author of a document is really who you think it is. (Only the originator has the key to the file.)

Encryption Does Not:

- Prevent an attacker from eliminating your data all together. A disgruntled employee or competitor could just delete all encrypted files on your system
- Protect the encryption mechanism itself. In other words, the attacker might modify the program to use a key different from the one you provide or might record all the encryption keys in a special file for later retrieval
- Prevent a vulnerability from being discovered which would allow the encryption to be used in a negative manner
- Prevent an attacker from gaining access to your files before they have been encrypted or after they have been decrypted

Encryption should not be viewed as the only source of protection for your data. It should be considered just one step of a security process.

Popular Linux Encryption Tools

GnuPG / PGP : PGP is the famous encryption program by Phil Zimmerman which helps secure your data from eavesdroppers and other risks. GnuPG is a very well-regarded open source implementation of the PGP standard (the actual executable is named GPG). While GnuPG is always free, PGP costs money for some uses.

OpenSSL : The premier SSL/TLS encryption library - The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and open source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

Source: http://sectools.org/crypto.html

Recommended Reading

- GNUPG
- PGP & GPG
- OpenSSL

Popular Linux Encryption Tools

Tor: An anonymous Internet communication system - Tor is a toolset for a wide range of organizations and people that want to improve their safety and security on the Internet. Using Tor can help you anonymize web browsing, publishing, instant messaging, IRC, SSH, and other applications that use the TCP protocol. Tor also provides a platform on which software developers can build new applications with built-in anonymity, safety, and privacy features. For a free cross-platform GUI, users recommend Vidalia

OpenSSL : The premier SSL/TLS encryption library - The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and open source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

Source: http://sectools.org/crypto.html

Recommended Reading

- Tor
- Vidalia

Popular Linux Encryption Tools

Stunnel : A general-purpose SSL cryptographic wrapper - The stunnel program is designed to work as an SSL encryption wrapper between remote client and local (inetd-startable) or remote server. It can be used to add SSL functionality to commonly used inetd daemons like POP2, POP3, and IMAP servers without changes in the programs' code. It will negotiate an SSL connection using the OpenSSL or SSLeay libraries.

OpenVPN : A full-featured SSL VPN solution - OpenVPN is an open-source SSL VPN package which can accommodate a wide range of configurations, including remote access, site-to-site VPNs, Wi-Fi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls. OpenVPN implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or two-factor authentication and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN uses OpenSSL as its primary cryptographic library.

Source: http://sectools.org/crypto.html

Recommended Reading

- Stunnel
- OpenVPN

Popular Linux Encryption Tools

TrueCrypt: Open-Source Disk Encryption Software for Windows and Linux –

TrueCrypt is an excellent open-source disk encryption system. Users can encrypt entire filesystems, which are then encrypted / decrypted on-the-fly as needed without user intervention beyond entering their passphrase initially.

A clever hidden volume feature allows you to hide a 2nd layer of particularly sensitive content with plausible deniability about its existence. Then if you are forced to give up your passphrase, you give them the first-level secret. Even with that, attackers cannot prove that a second level key even exists.

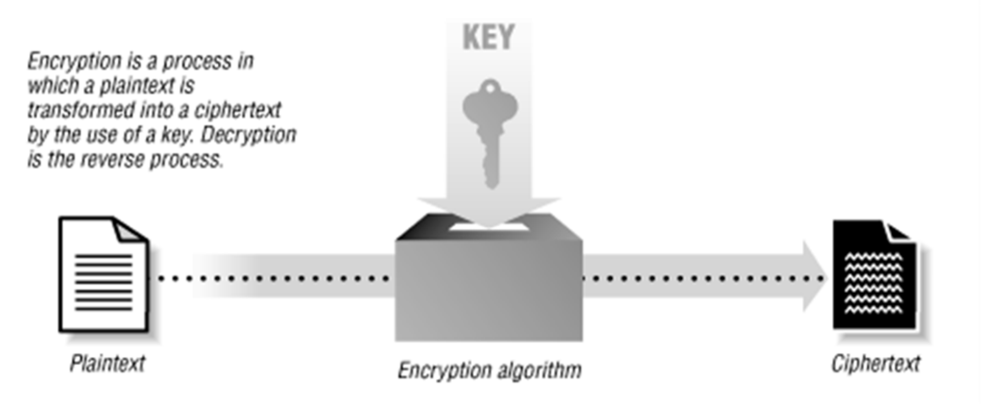
Source: http://sectools.org/crypto.html

Recommended Reading

- TrueCrypt

How File Encryption Work

Whole file encryption is the process of encrypting an entire file using a first-party or third-party software solution. The encrypting software rearranges the data of the file using an encryption algorithm, rendering it unable to be read or run until the file has been decrypted. The length of the algorithm determines the difficulty that another program or programmer would have in decrypting the data without the use of the encryption algorithm. The more complex the algorithm (in terms of its bit size), the more difficult it will be to break. The most common encryption types are 32-bit, 64-bit, 128-bit, and 256-bit.



Encryption is a process in which a plaintext is transformed into a ciphertext by the use of a key. Decryption is the reverse process.

Diagram showing encryption process from plaintext to ciphertext
Source: *Practical Unix & Internet Security*

Select **PLAY** below for a video on Encryption.

View Video VideoLesson4Encryption(C5L4S17V2.mp4

Encryption

View Video VideoLesson4Algorithm(C5L4S17v1).mp4

Algorithm

Cryptographic Strength

Different forms of cryptography are not equal. Some systems are easily circumvented or broken, others are quite resistant to even the most determined attack. The ability of a cryptographic system to protect information from attack is called its strength. Strength depends on many factors, including:

- The secrecy of the key
- The difficulty of guessing the key or trying all possible keys (a key search). Longer keys are generally harder to guess or find.
- The difficulty of inverting the encryption algorithm without knowing the encryption key (breaking the encryption algorithm)
- The existence (or lack) of back doors, or additional ways by which an encrypted file can be decrypted more easily without knowing the key
- The ability to decrypt an entire encrypted message if you know the way that a portion of it decrypts (called a known text attack).

Recommended Reading

- Content Source

Cryptographic Strength (Contd)

The strength of an encryption method should be carefully considered when selecting an encryption method. As the strength and process gets more complicated, more system resources will be utilized in performing the process of encoding or decoding. This may result in decreased system performance on machines that have limited resources (RAM).

Note: Students should carefully think about the hard drive before they encrypt the contents. Items such as CPU type, available RAM, frequency of the CPU, size of the target drive and length of the passphrase should all be taken in to consideration prior to encrypting a device.

Recommended Reading

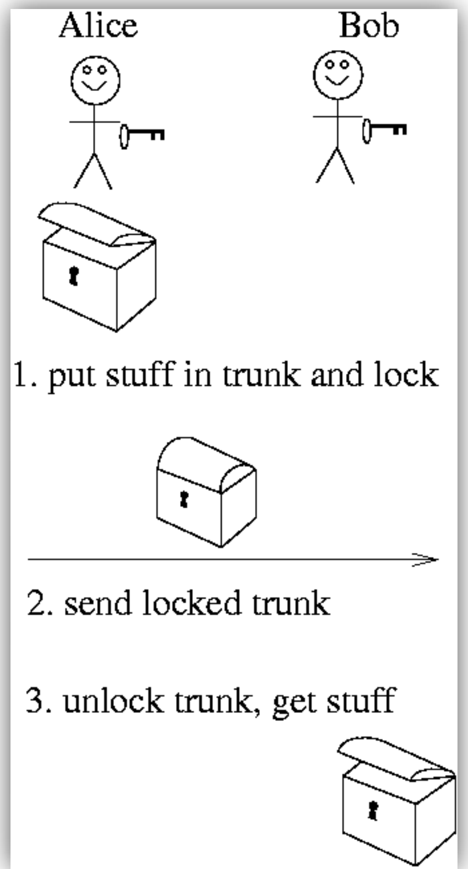
- Content Source

Private Key

With Private Key cryptography (secret key), a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver.

The receiver applies the same key (or ruleset) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption.

Private key cryptography is most often used for protecting information stored on a computer's hard disk or for encrypting information carried by a communications link between two different machines.



Source: wellesley.edu

Recommended Reading

- Encryption
- Cryptography

Public Key

Two keys are used in Public Key encryption. One set of keys are distributed publicly or to anyone who wants to send you an encrypted message. The creator of the message will use your public key to encrypt the file or message and send it to you.

When you receive the message, you will use your private key (available only to you) to decode the message or file.

Public key cryptography is most often used for creating digital signatures of data, such as electronic mail, to certify the data's origin and integrity.

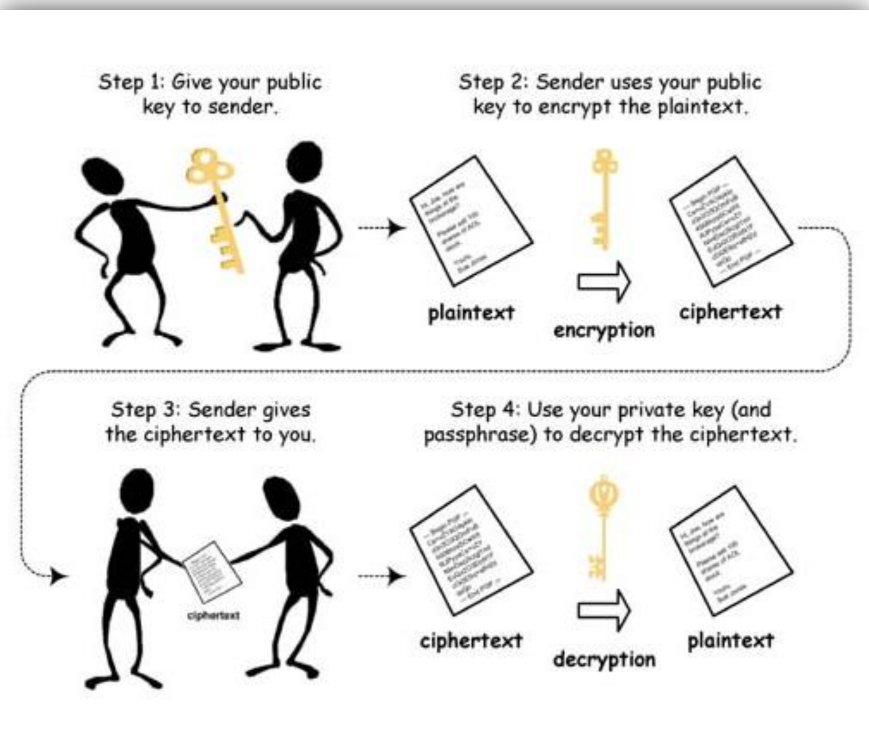

Step 1: Give your public key to sender.
Step 2: Sender uses your public key to encrypt the plaintext.
plaintext
encryption
ciphertext
Step 3: Sender gives the ciphertext to you.
ciphertext
Step 4: Use your private key (and passphrase) to decrypt the ciphertext.
ciphertext
decryption
plaintext

Image source: uic.edu

Select **PLAY** below for a video on security terms

View Video VideosLesson4SecurityTerms(C5L4S21).mp4

Recommended Reading

- PGP Security

Loop AES

Installation and Configuration

Getting Loop-AES

Enter only the commands in bold. You need to be logged in as root:

1. **apt-get install loop-aes-2.6.9 loop-aes-modules-2.6.18-4-686 loop-aes-utils**
2. **modprobe loop-aes**
3. **echo "loop-aes" >>/etc/modules**
4. **head -c 2925 /dev/random | uuencode -m - | head -n 66 | tail -n 65 | gpg --symmetric -a >/etc/fskey-hdd.gpg**
   (If it takes a very long to create this key, check the available entropy on your system with this command:)
   cat /proc/sys/kernel/random/entropy_avail
5. **fdisk /dev/sdb** *(create partition on /dev/sdb1)*
6. **losetup -e AES256 -K /etc/fskey-hdd.gpg /dev/loop0 /dev/sdb1**
7. **mkfs -t ext3 /dev/loop0**
8. **mkdir -p /media/sdb1**
9. **mount -t ext3 /dev/loop0 /media/sdb1**

Be sure to replace /dev/sdb1 with your partition name.

To Unmount:

- **unmount /dev/loop0**
- **losetup -d /dev/loop0**

To Mount Existing:

- **losetup -e AES256 -K /etc/fskey-hdd.gpg /dev/loop0 /dev/sdb1**
- **mount -t ext3 /dev/loop0 /media/sdb1**

Source:

- Loop AES on Debian

DM-Crypt

Installation and Configuration

Installing DM-Crypt

1. Enter: **sudo apt-get install cryptsetup**
2. Next partition the unmounted USB pen drive the way you want it.
3. Don’t mount the disk afterwards!

Unless you’ve rebooted your computer since you installed the cryptsetup package, you might have to load the device mapper crypt module first. Enter this command:

```
sudo modprobe dm-crypt
```

Next, you should encrypt the partition. Use this command:

```
$ sudo cryptsetup --verbose --verify-passphrase luksFormat /dev/sda1
```

This action will overwrite data on /dev/sda1 irrevocably:

1. Are you sure? [Type uppercase yes]: YES
2. Enter LUKS passphrase:
3. Verify passphrase:
4. Command successful.

Installing DM-Crypt (Contd)

If you get the error: “Failed to setup dm-crypt key mapping,” do the following:

1. Check kernel for support for the aes-cbc-essiv:sha256 cipher specification.
2. Verify that /dev/sda1 contains at least 133 sectors.
3. Ensure the disk is not mounted.
4. Make sure you are using the right device. Use **dmesg** to check the device to which the disks have been assigned.
5. Check that the module *dm-crypt* is loaded. Use **lsmod | grep dm**
6. Enter command: **$ sudo cryptsetup luksOpen /dev/sda1 sda1**
7. Enter LUKS passphrase:
8. Screen display: “key slot 0 unlocked”
9. Screen display: “Command successful.”

To create a filesystem

1. Enter this command: **sudo mkfs.ext3 /dev/mapper/sda1**
2. remove the temporary device mapped to the encrypted partition with the following cmd:

   sudo cryptsetup luksClose sda1

Pull your thumb drive from the USB socket and reinsert it.
If you are using Virtual Machine, you may need to recapture the USB device in the device panel.

Additional Resources

You will find the following links on Squid helpful for this lesson:

- DM-Crypt – A device-mapper support site
- DM-Crypt and LUKS Tutorial
- DM-Crypt / LUKS removable disk encryption
- Working with encrypted files on Ubuntu
- How encryption works
- What is Encryption?
- Overview of Encryption
- Cryptographic File System
- Simple Example of Encryption
- Ubuntu Full Disk Encryption

Lesson Summary

Encryption is an additional step towards keeping your valued data protected. This lesson explored the history of encryption, private and public keys, and the encryption process. You also learned basic cryptographic terminology.

Encryption is the process of taking plaintext files and rearranging them in a secure fashion using a complex algorithm so that the file is no longer readable without special tools. To access the file, a person or a program would need know the exact sequence of steps to undo the security changes and reveal the original message.

Some encryption schemes use one key to encode and decode messages and are called symmetric encryption. Other schemes use two keys—a public key (made available publicly) to encrypt messages, and a private key (known only to the owner) to decrypt or decode messages. A public/private key system is called asymmetric encryption.

Some encryption schemes are very difficult to break, while others have flaws or vulnerabilities that make them easier to access. Use encryption as only one of many options when securing sensitive data.

Select **PLAY** below for a summary video on encryption.

View Video VideoLesson4EncryptionSummary(C5L4S28).mp4