System and Network Administration

Cryptography

Secure Systems

1. Security policy
   - What needs to be protected
   - Kinds / level of protection
   - Responsibilities
   - Auditing policy
2. Security environment
   - Physical environment
   - Physical security
   - Hardware, operating system
   - firewalls, etc
3. Security mechanisms
   - cryptography
   - authentication
   - security protocols
4. Monitoring and auditing procedures
   - monitor access
   - audit trails
   - feedback on failures, security breaches
   - containment & recovery

Security Management

- The ***security policy*** defines what information is to be protected and from whom
- ***Security mechanisms*** implement aspects of the **security policy**, and their **effectiveness** must be monitored

Security Protocols

In practice, no single mechanism is adequate to address all goals, so a mix of mechanisms will be required to enforce security policies.

A protocol is an orderly sequence of steps that two or more parties follow in order to accomplish some joint task

e.g. protocols for

- authentication of participants in an exchange of messages
- data integrity checks

Encryption is a mechanism that can be incorporated into security protocols

Cryptography

The science of hiding information, most commonly by encoding and decoding a secret code used to send messages.

- Based on mathematics and computer science.
- Protects **data in transit** and **data at rest.**

data in use *is in memory or being processed by the CPU, so it cannot be encrypted*

Greetings, Mr. Logawps,

We received your request for information and will be happy to oblige. Here is your user name and password:

G7JDZL	L59CZ2	AA9CZ1
ZPQ12G	93D2BA	LP7FFH
18ABHF	UJ14A3	34FYO5
K71TYP	CS1314	566HXH

Cryptographic techniques

- **Encryption**: Scrambling a message
- **Algorithm**: Method used for scrambling
- **Key**: the output of certain encryption algorithms is controlled by a value called an encryption key

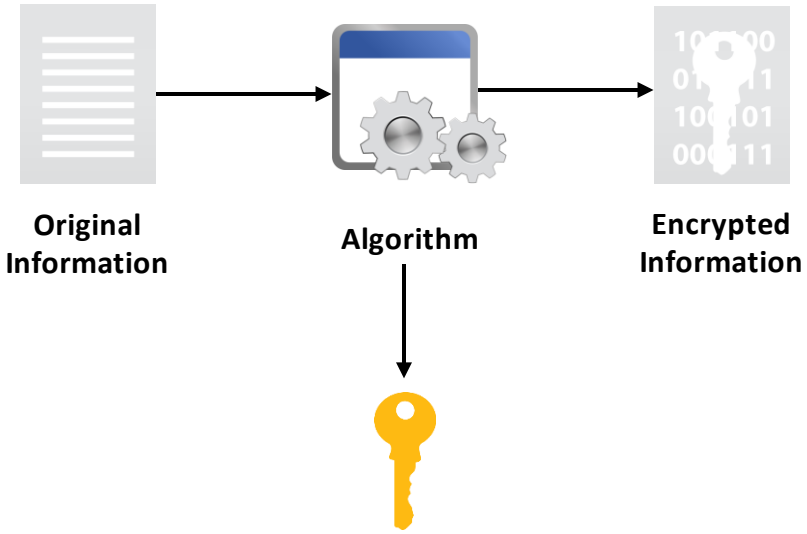
More formally: The algorithm is used to transform plaintext into cyphertext

Algorithms

Confusion: encrypted data looks much different than the data we started with

Diffusion: changing one character of the original data makes the encrypted data look much different

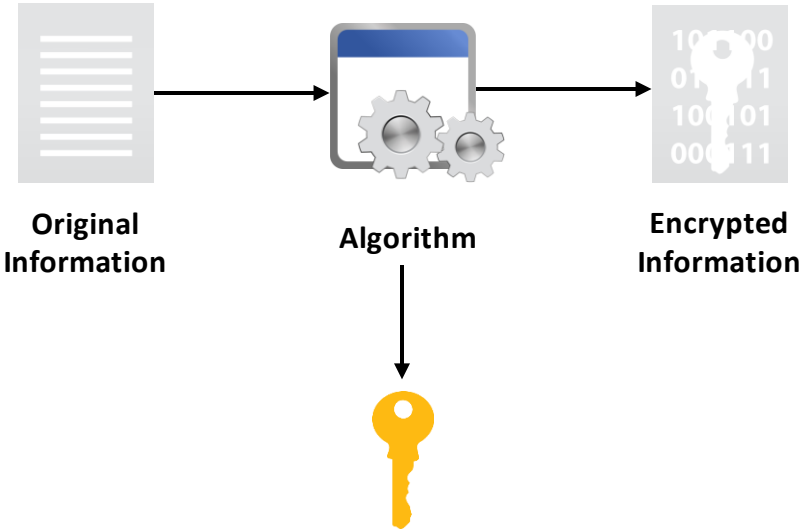
- An **encryption key** is used to help strengthen these mathematical properties of an algorithm



A Key

A specific piece of information that is used in conjunction with an algorithm to perform encryption and decryption.

- Different keys produce different ciphertext.
- For each algorithm, longer keys provide stronger encryption.

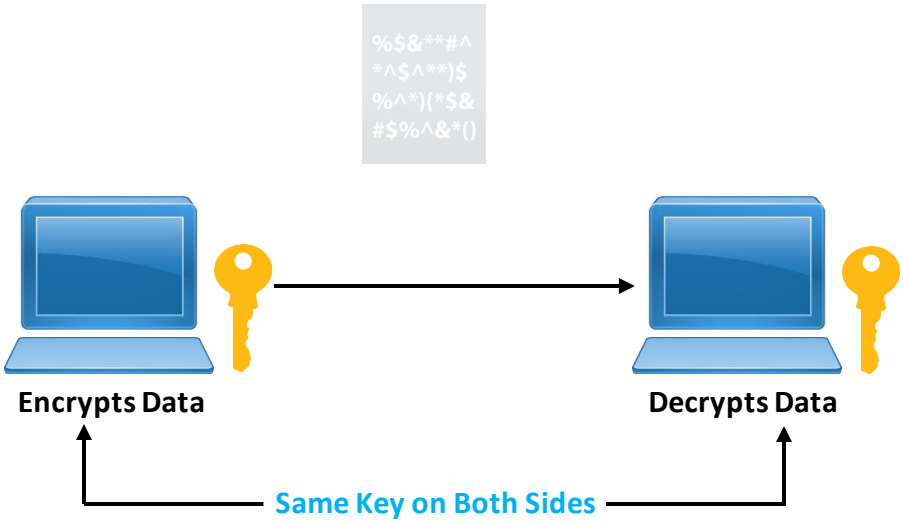


A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Symmetric Encryption

A two-way encryption scheme in which encryption and decryption are both performed using the same key (shared key encryption).

- Before encrypted communications begin, the key must be securely shared.
- Fast, but vulnerable if the key is lost or compromised.
- Common alternate names
  - Secret key
  - Shared key
  - Private key



Asymmetric Encryption

Asymmetric encryption: A two-way encryption scheme that uses paired public and private keys.

Private key: The component of asymmetric encryption that is kept secret by one party during two-way encryption.

Public key: The component of asymmetric encryption that can be accessed by anyone.

Key generation: The process of producing a public and private key pair by using a specific application.

Asymmetric Encryption (Cont.)


Public key


Private key

Uses a key pair (*a*,*b*)

- What one key does, the other can undo
  - Either key can be used to encrypt a message - the other key is then used to decrypt it
  - The message cannot be decoded with the key that encoded it
  - The message can only be decoded by a matching key
- Knowledge of one key does not allow the other to be deduced

What one key does, the other key will undo
Can use either key to encrypt

One key can be published while the other is kept secret and secure

Asymmetric, Reversible (Public Key)

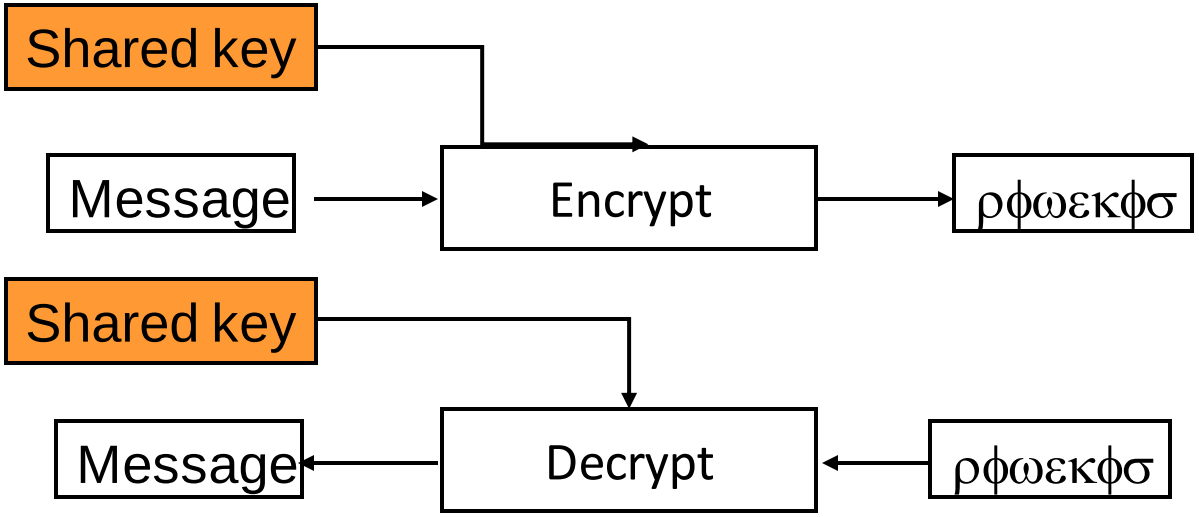
- **Private Message**
  - All devices can use a station's public key to encrypt data to send to the station.
  - The receiving station decrypts the data using its own private key.
- **Assured Message**
  - Sender uses private key to encrypt data
  - The receiving station decrypts the data using the sender's public key.
  - If the private key is kept secure, cannot deny sending the message (**non-repudiation**)

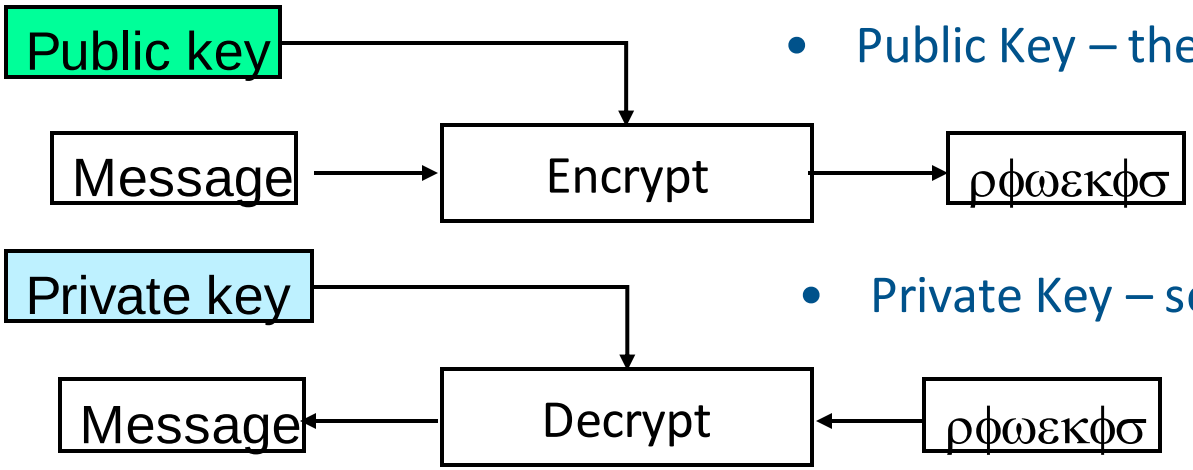
Non-repudiation

Ensuring that the party that sent a transmission or created data remains associated with the data and cannot deny sending or creating the data.

Reversible Encryption

- **Symmetric**





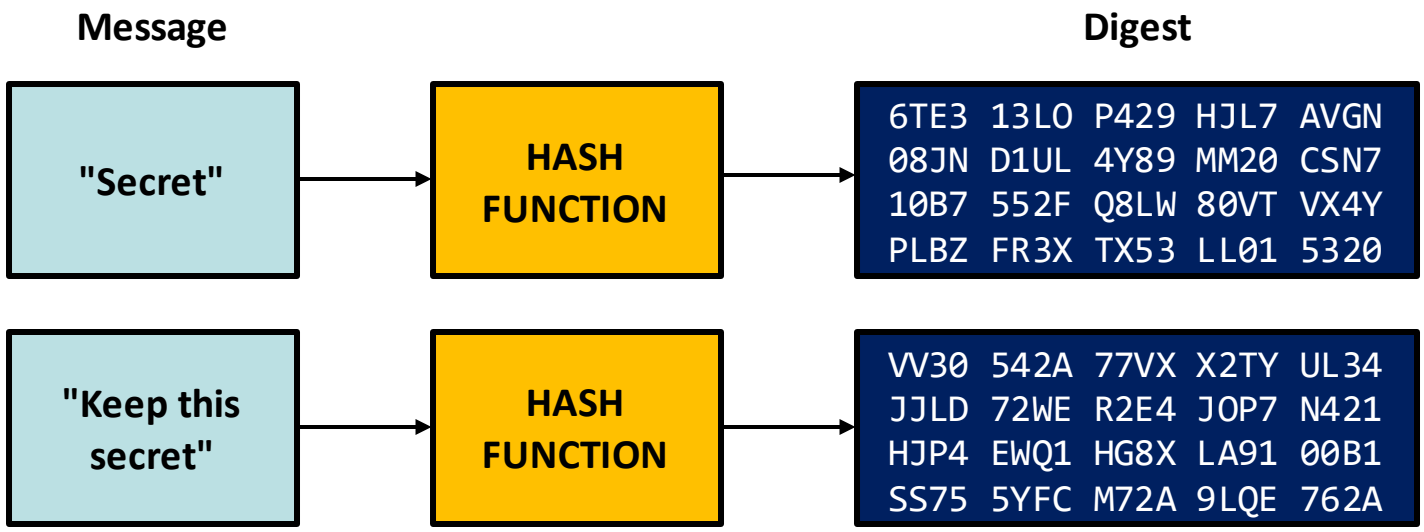
- Public Key – the key is publicly known
- **Asymmetric**
- Private Key – secret key

Hashing

Hashing: A process or function that transforms plaintext to ciphertext that cannot be directly decrypted.

Hash, hash value, or **message digest**: The value that results from hashing encryption.

- Used in standard password authentication schemes.
- Used in digital signatures.
- Used for verifying file integrity.



/etc/shadow

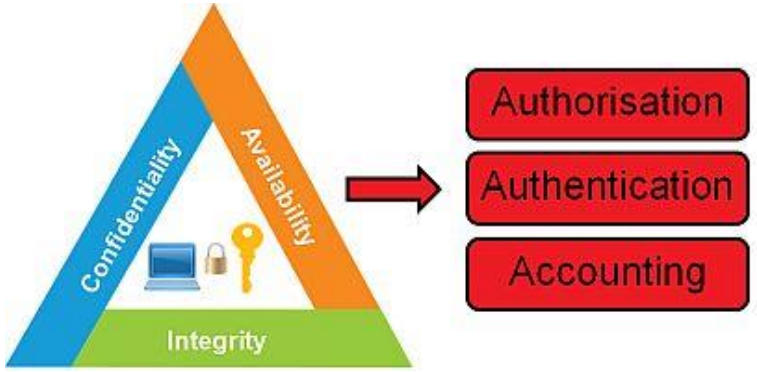
When the system administrator sets up an account, the password is typed in for the first time. The system stores the encrypted password in a secure database.

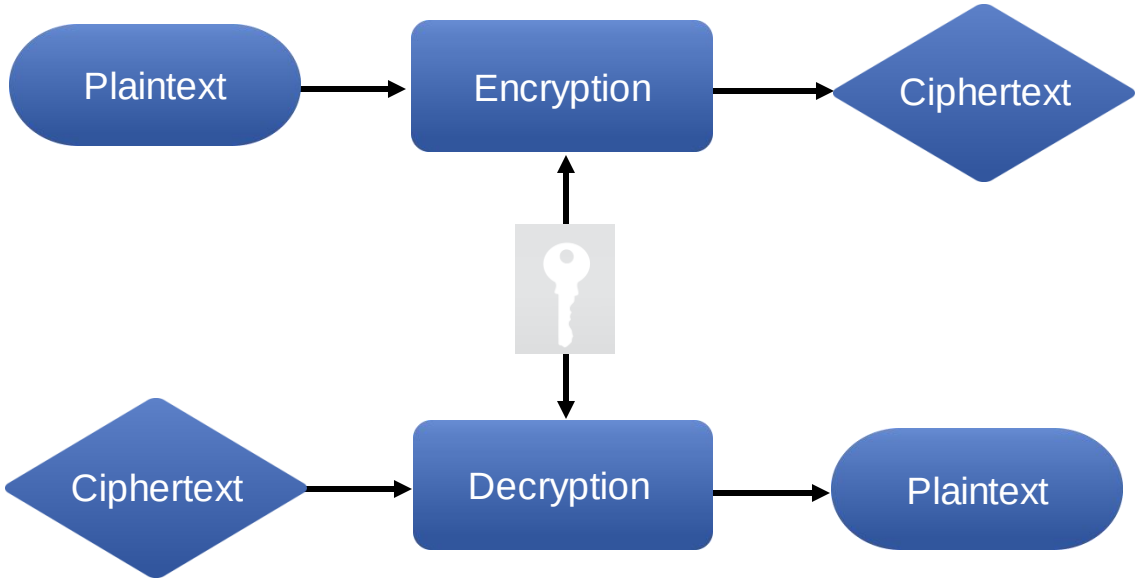
```
          User : Password : UID : GID : GCOS : Home : Shell
root:KnR8gQb3hDYg6:0:0:root:/root:/bin/bash
httpd:zYFuFaKw6ZovI:15:60:Apache_1.2.4:/home/httpd:/bin/bash
tomas:k6x5IqJcB1C.Q:500:500:TommyGun:/home/tomas:/bin/bash
guest:qfRc424I3/QGI:502:502:Guest User:/home/guest:/bin/pdmsh
jacob:dbBOzTSdCX5HU:503:503:Linux User:/home/jacob:/bin/bash
peterlai:T5RlEJUib7wOk:504:504:Linux User:/home/peterlai:/bin/bash
toshiba:DPz8cCRkIQc1Q:505:502:SLIP User:/home/rest:/usr/sbin/dip -i
victor:wiSml8WiVwJt2:506:506:Linux User:/home/victor:/bin/bash
dolphin:5JaZ7KmR7hEzM:508:507:Mail User:/home/dolphin:/usr/bin/pine
```

Each time the user logs in after that, the system encrypts the password that has been typed in and compares it to the cyphertext stored in the database.

Encryption and Security Goals

- **Confidentiality**
- **Integrity**
- **Access control**
- **Authentication**
- Non-repudiation





"The CIA Triad"

Principle	Description
Confidentiality	Keeping information and communications private and protected from unauthorized access. • trade and military secrets; personnel, health, and tax records. • Controlled via encryption, access control, and steganography.
Integrity	Keeping organizational information accurate, free of errors, and free from unauthorized modification. • Includes modification information stored on network servers. • Controlled via hashing, digital signatures, certificates, and change control.
Availability	Ensuring that computer systems operate continuously and that authorized persons can access the data they need. • Includes ensuring that vital data are both captured and distributed • Controlled via redundancy, fault tolerance, and patching.

- The important feature of most cryptographic operations is that they are easy to perform if you have the right information and infeasible to perform if you don't have that information.
- Trying to keep the design of a security system secret as its only method of security is called *security through obscurity*.
- **The best kind of security exists when the attacker would know everything about the way the system works but still would not be able to gain access to any of the data.**

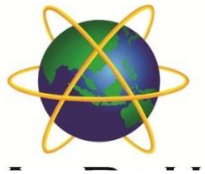
- Application developers need not become experts in cryptography to be able to use cryptography in their applications.
  - They simply use an application programming interface (API) to a cryptography module (library).
- Programmers don't have to worry what the implementation of the cryptography, they only have to understand how to provide plaintext or cyphertext to the API and get the answer back from the API.

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Cryptographic Attacks

A software attack that exploits weaknesses in cryptographic system elements, such as code, ciphers, protocols, and key-management systems.

Types of Cryptographic Attacks



Cryptographic Attack Type	Description
Known plaintext attack (KPA)	• Attacker has a plaintext message and its corresponding ciphertext. • Attacker tries to derive the correlation between them to determine the encryption key.
Chosen plaintext attack	• Attacker encrypts a selected plaintext message. • Attacker analyzes the resulting ciphertext to crack the cipher. • Attacker uses attack results to iteratively repeat the attack for an adaptive chosen plaintext attack.
Ciphertext-only attack	• Attacker has access to ciphertext. • Attacker tries to use frequency analysis or other methods to break the cipher.
Chosen ciphertext attack	• Attacker analyzes a selected ciphertext message and tries to find the matching plaintext. • Attacker uses the attack results to iteratively repeat the attack for an adaptive chosen ciphertext attack.

Types of Cryptographic Attacks (Cont.)

Cryptographic Attack Type	Description
Downgrade attack	• Attacker exploits the need for backward compatibility. • Attacker forces a computer to abandon the use of encrypted messages in favor of plaintext messages.
Replay attack	• Attacker intercepts session keys or authentication traffic. • Attacker uses them later to authenticate and gain access.
Weak implementation attacks	• Focus on how the cryptographic system is implemented. • (Other cryptographic attacks focus on the algorithm used to encrypt the targeted data.)

Randomness

- Random numbers are used frequently in cryptography. One of the challenges with creating random numbers with a machine is that they are "pseudo-random", not truly random.
- This creates a problem: Confusion means there shouldn't be any patterns, and there should be no way to recognize any part of the plaintext by simply looking at the ciphertext.
- Truly random numbers would have no pattern, but machine-generated "pseudo-random" numbers often do have predictable patterns.

Randomness

- Randomness means unguessable input. If the input is guessable, then the output can be easily calculated.
  - Suppose we play several rounds of a game with the same deck of cards.
  - The shuffling of the deck between rounds is the primary source of randomness.
  - If we didn't shuffle properly, you could beat the game by predicting cards.

Multiple Mechanisms

Say you have an algorithm whose security properties are not very good if a lot of fairly predictable data is encrypted with the same key. You can fix this by adding randomness to the process.

Encrypt like this

1. Generate a random key.
2. Encrypt the data with that random key.
3. Encrypt the random key with the shared key.
4. Send the encrypted data from step 2 along with the encrypted key from step 3.

Decrypt like this

1. Decrypt the encrypted random key with the shared key.
2. Decrypt the encrypted data with the random key that you just decrypted.

The shared key is re-used, but only to encrypt random data.

David Schwartz -http://crypto.stackexchange.com/questions/2686/how-can-encryption-involve-randomness

Repetitive v. Random

- Let's say the daily message is

  Nothing to report - 10:43PM January 7

- Without randomness, the first bytes of the encrypted message match the next message if it is

  Nothing to report - 10:43PM January 8

- Without randomness, an attacker can tell if two encrypted outputs correspond to the same plaintext just by comparing them.

- Even with some randomness, enough of these messages could be collected for cryptanalysis.

Multiple Mechanisms

- But if we encrypt the two predictable messages

  Nothing to report - 10:43PM January 7

  Nothing to report - 10:43PM January 8

- with different random keys that are never reused, the attacker cannot tell if two encrypted outputs correspond to the same plaintext just by comparing them.

- Even an attacker who gets to choose what data you encrypt has no control over what data is encrypted with the persistent key.

This is a simple example, but it shows two things.

1. It shows how an encryption process can be perfectly reversible but still involve randomness.

2. It shows how using randomness in an encryption process can improve the security properties.

Security Protocols

- You can't safely assume that an eavesdropper doesn't have complete details of any/all encryption and decryption algorithms
- A security protocol can be considered secure only if someone who knows all of the details of these algorithms is unable to recover a message without trying every possible key.
- Ultimately, security rests on the infeasibility of trying all possible decryption-key values.

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION