

Introduction to IT Systems for Beginners

Maciej Gowin @ [CoderBrother](#)

All rights reserved

Cryptography

Cryptography is the practice of **securing information** by transforming it into a code, making it **unreadable to unauthorized users**.

Purpose

- Ensures **confidentiality, integrity, and authentication** of data, especially in online communications, banking, and data storage.
- It's widely used to protect data in digital communication.

Common Uses of Cryptography

- Password protection, secure messaging (e.g., WhatsApp), digital certificates for website security (HTTPS).

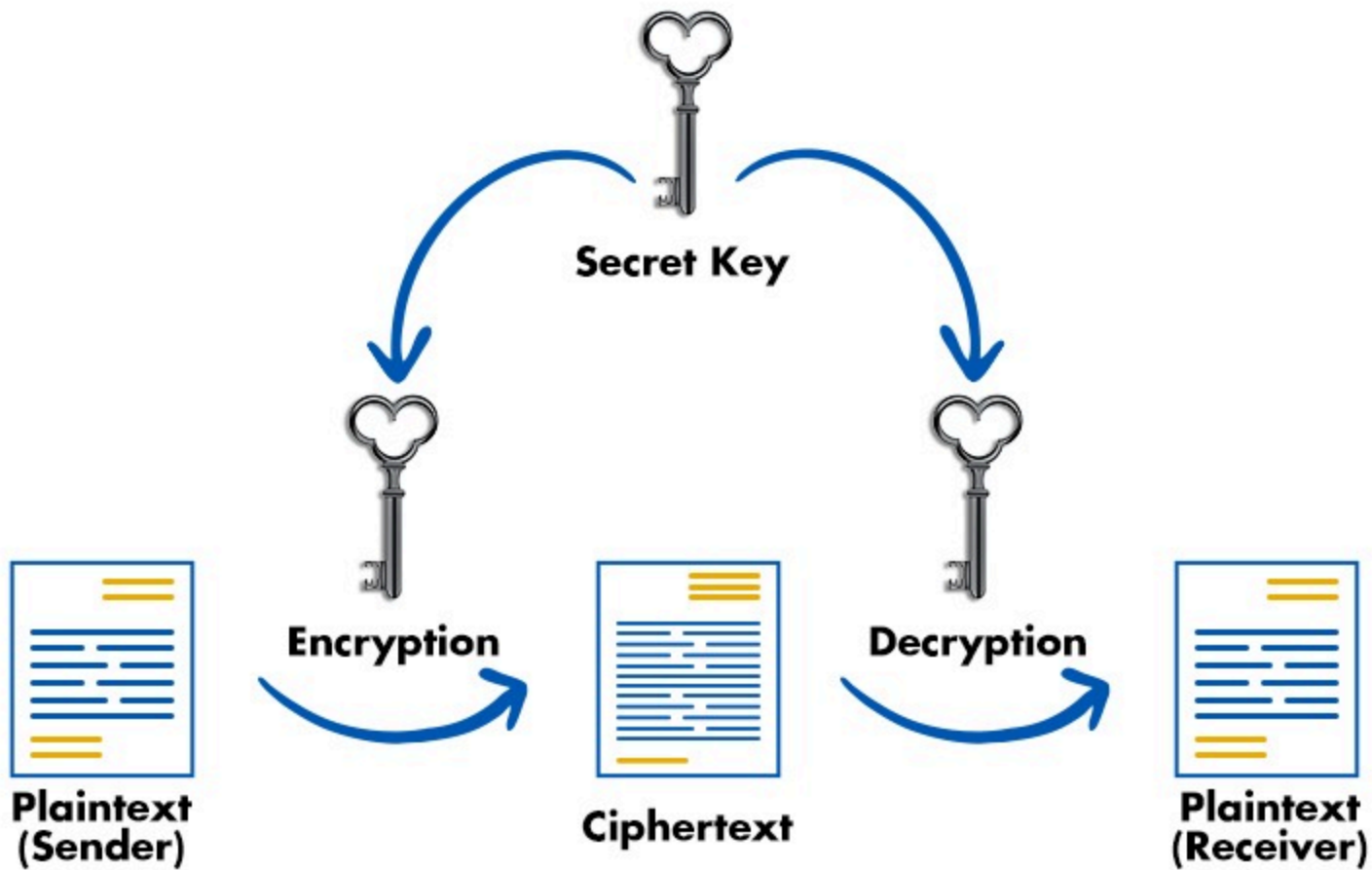
Encryption

- Encryption is the process of converting data into a coded format (ciphertext) to protect it from unauthorized access. Only those with the correct decryption key can revert it back to readable form.
- **Protects data privacy** by ensuring that only authorized parties can access sensitive information, even if intercepted.

Basic Types of Encryption

1. Symmetric Encryption

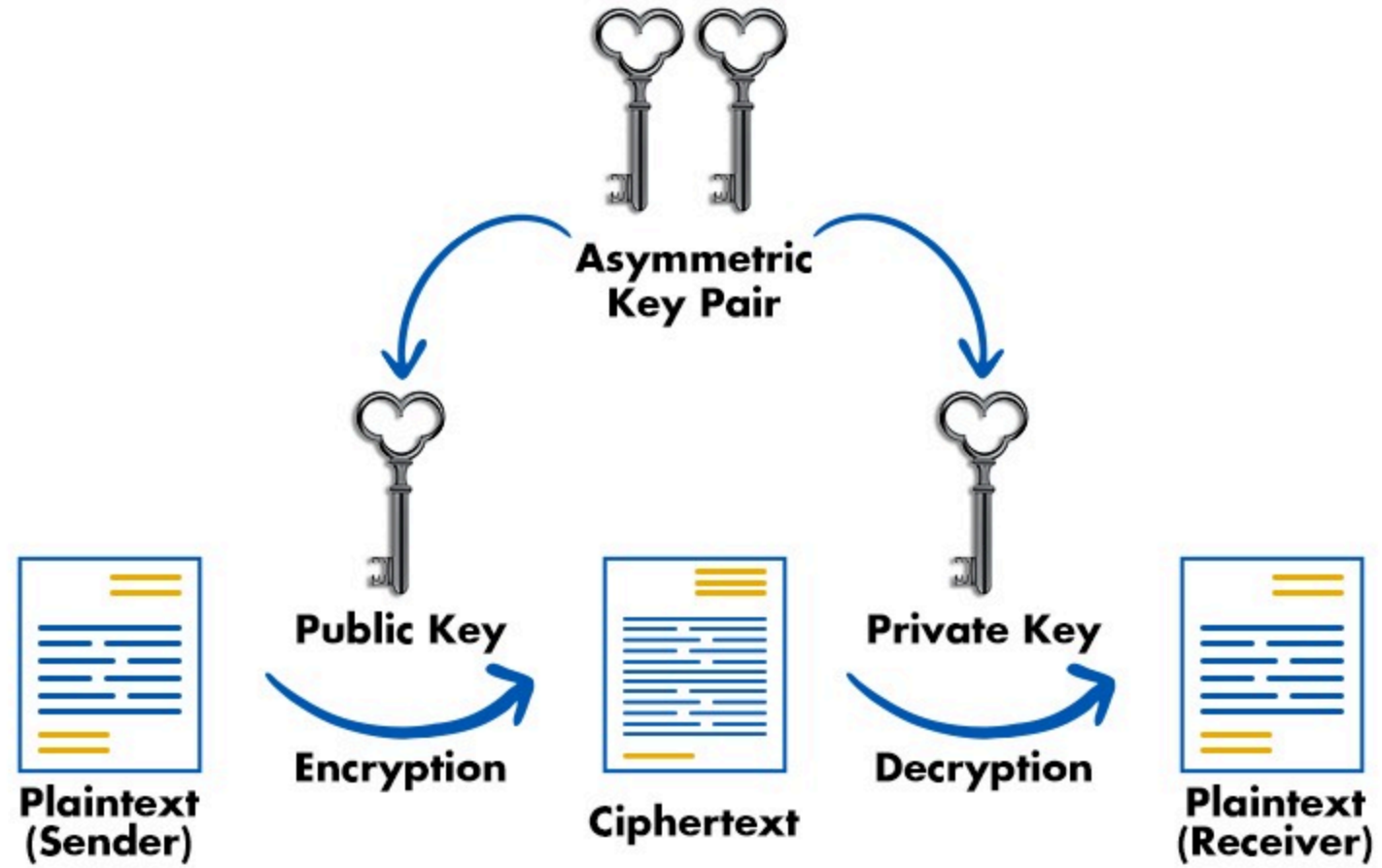
- Uses a **single key** to both encrypt and decrypt data.
- Faster, but requires both parties to share the same key.



Basic Types of Encryption

2. Asymmetric Encryption

- Uses a pair of keys: a **public key** to encrypt data and a **private key** to decrypt it.
- More secure for sharing information, as only the private key holder can decrypt messages.



Examples of Encryption Algorithms

- **RSA (Rivest-Shamir-Adleman)** - symmetric, commonly used in secure internet communications (e.g., SSL/TLS).
- **SHA-256 (Secure Hash Algorithm)** - asymmetric, often used to securely store passwords.

Hashing

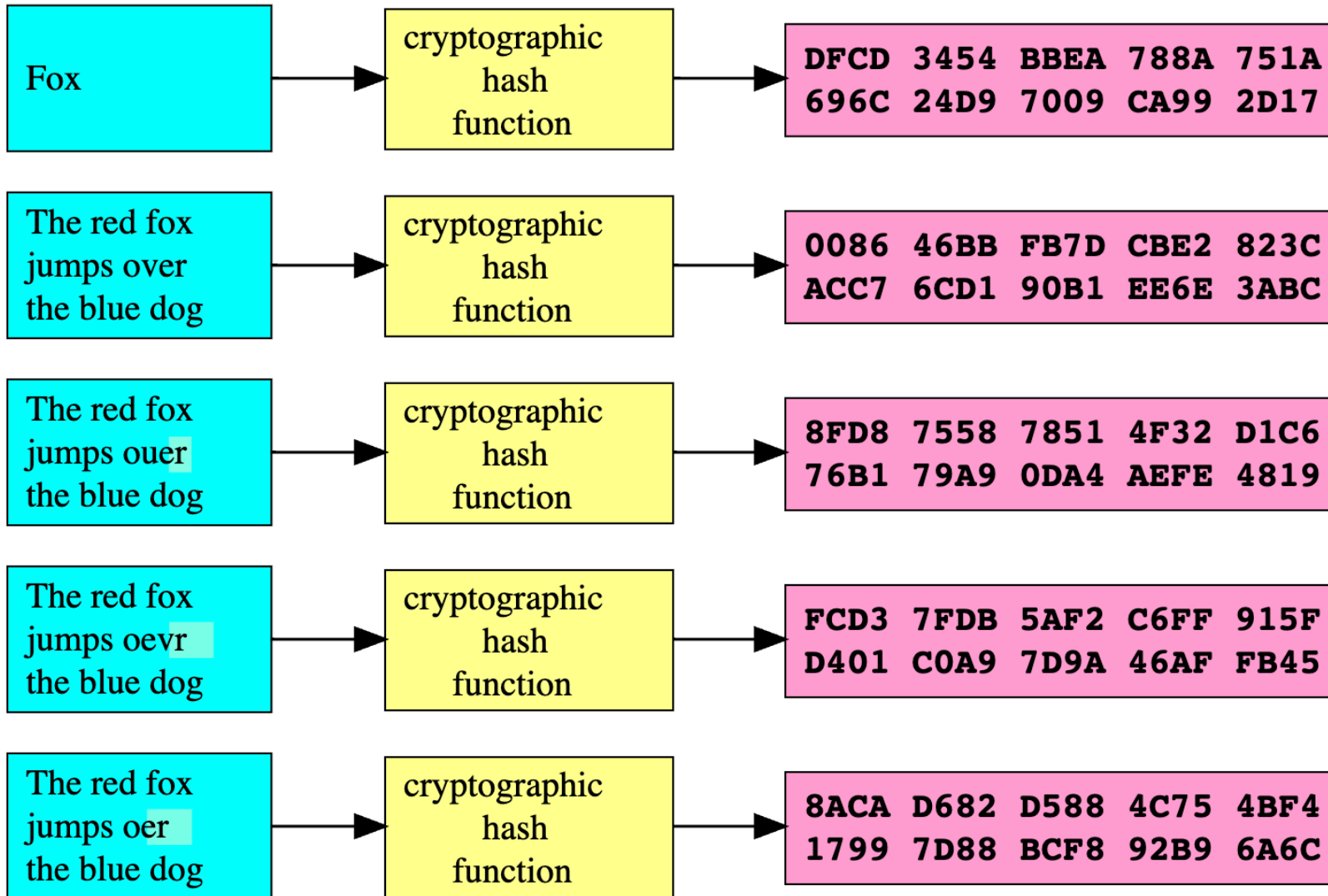
Hashing is a process that converts data (like a password or a file) into a fixed-length string of characters, usually a random-looking sequence. This string is called a hash.

Purpose

- Provides data integrity and verification without revealing the original data.
- Hashes are commonly used to securely store passwords and verify data.

Input

Digest



Key Characteristics of Hashing

- **One-Way Function:** Hashing is irreversible, meaning you can't convert the hash back to the original data.
- **Fixed Output Length:** Regardless of the input size, the output hash is always the same length.
- **Unique Output:** Even small changes in input produce a completely different hash (called the “avalanche effect”).

Common Uses of Hashing

- **Password Storage:** Passwords are hashed before being stored to protect user security.
- **Data Verification:** Hashing checks file integrity to detect changes or tampering.
- **Digital Signatures:** Ensures that documents haven't been altered.

Examples of Hashing Algorithms

- **MD5 (Message Digest Algorithm 5)** - produces a 128-bit hash, commonly used in file integrity checks.
- **SHA-256 (Part of the SHA-2 Family)** - produces a 256-bit hash and is widely used in security applications, including SSL/TLS certificates.