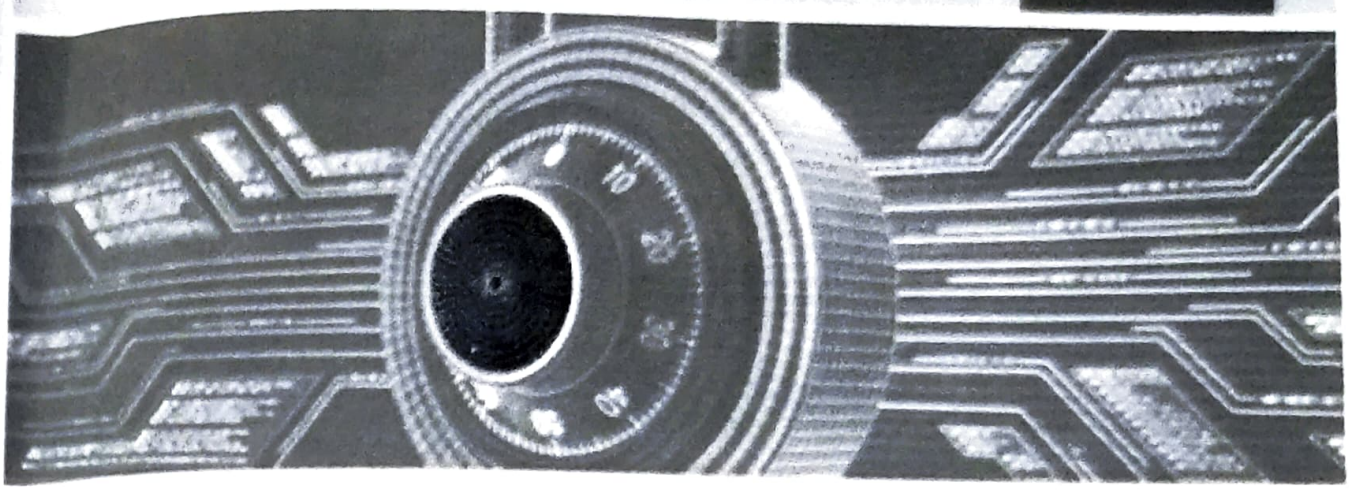
Chapter 1



CRYPTOGRAPHIC TERMINOLOGY AND PROTOCOLS

1.1 CRYPTOGRAPHY TERMINOLOGY

Cryptography is an important aspect when we deal with network security. The prefix 'Crypto' means secret or hidden and suffix 'graphy' means "writing". Cryptography is the science of secret writing with the intention of keeping the data secret.

Cryptography is like a secret code between friends. Imagine writing a message in a secret language only you and your friend understand. To keep it private, you use a special key to change the message into a code. Your friend uses the same key to decode the message back to its original form. It's like hiding your conversation from others, making sure only you and your friend can understand what's being said. Cryptography secures information by turning it into a secret code that only authorized parties can unlock.

Objectives of Cryptography

- **Confidentiality:** Protects data from unauthorized access or viewing by ensuring only intended recipients can read it.

- **Integrity:** Verifies data integrity, ensuring that information remains unchanged during transmission or storage.
- **Authentication:** Validates the identities of communicating parties to prevent impersonation or unauthorized access.
- **Non-repudiation:** Prevents denial of involvement in a communication or transaction, ensuring accountability.
- **Key Management:** Securely handles the generation, distribution, and storage of cryptographic keys to maintain security.
- **Secure Communication:** Facilitates secure transmission of information across networks, safeguarding against eavesdropping and tampering.

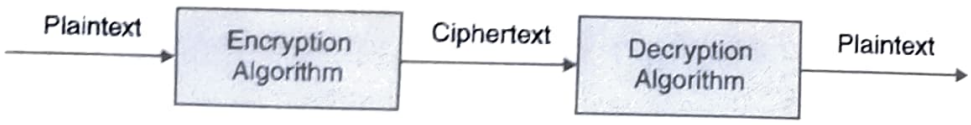
Cryptography Terminology

(a) **Plaintext:** It refers to the original message that the person wants to connect with the other. It's the message or data in its natural form, susceptible to being understood by anyone without any encryption or special codes applied.

(b) **Ciphertext:** It refers to the encrypted text. Plaintext is what you have before encryption, and ciphertext is the encrypted result. It is encrypted information, transformed from plaintext using cryptographic techniques.

(c) **Encryption:** It is a process of transforming plain text into cipher text. Cryptography needs the encryption approach to send confidential messages through an insecure channel.

(d) **Decryption:** A reverse process of encryption is known as *Decryption*. It is a procedure of transforming Cipher Text into Plain Text. Cryptography needs the decryption approach at the receiver side to acquire the original message from non-readable message (Cipher Text).



(e) **Cipher:** A cipher is a method or algorithm used in encryption and decryption to convert plaintext into ciphertext or vice versa. It's a set of rules or techniques for securing and transforming sensitive information.

(f) **Key:** In cryptography, a key is a piece of information used by an algorithm to encrypt or decrypt data. There are two main types:

- *Symmetric Key:* A single shared secret key used for both encryption and decryption. Both parties communicating must possess the same key.
- *Asymmetric Key (Public Key):* It involves a pair of keys: a *public key* used for encryption and a *private key* used for decryption. Information encrypted with the public key can only be decrypted with the corresponding private key.

(g) **Cryptography**: It is the study and practice of secure communication techniques. It involves creating codes and ciphers to protect sensitive data, ensuring confidentiality, integrity, authentication, and non-repudiation in information exchange and storage.

(h) **Cryptanalysis**: It is the study of analyzing and deciphering encrypted information without authorized access or keys. It involves breaking codes, ciphers, or encryption systems to uncover the original plaintext or vulnerabilities in cryptographic schemes.

1.2 STENOGRAPHY

It is the practice of concealing messages or information within other non-secret data (like images, audio files, or text) to avoid detection. It aims to hide the existence of the communicated information.

It's a way to send secret information without anyone knowing it's there. By changing tiny parts of the picture or song, the hidden message stays invisible to most people. It helps keep information private during communication, like a secret code within something that looks ordinary.

Example

Imagine you have a picture of a beach with a colorful umbrella. In steganography, you can hide a secret message within this image without changing its appearance much.

Let's say you and your friend agree to use a simple method: hiding the message in the colors of the umbrella. Each color in the image is made up of three primary colors: *red*, *green*, and *blue*.

Your secret message is "HI". To hide this message:

1. **Convert Message to Binary:**
   - H: 01001000
   - I: 01001001
2. **Modify Pixel Colors:** You change the last bit of each color component (red, green, and blue) in some pixels of the umbrella to represent your binary message. For example:
   - Change the last bit of some red, green, and blue values to match the binary of "HI" in selected pixels of the umbrella without significantly changing the overall colors.
3. **Send the Image:** You send the modified beach image to your friend.
4. **Retrieve the Message:** Your friend knows which pixels and colors to look at to extract the hidden message. They identify these modified bits.
5. **Convert Binary Back to Text:** By converting the retrieved binary back to text:
   - 01001000: H
   - 01001001: I

Your friend reveals the hidden message as "HI"!

In this example, the message "HI" was hidden within the colors of the umbrella without noticeably altering the overall appearance of the beach image. This demonstrates a basic way steganography can hide messages within images.

1.3 SUBSTITUTION CIPHERS

Substitution ciphers are a type of encryption technique that involves replacing plaintext characters with different ciphertext characters based on a specific rule or algorithm. In a substitution cipher, each letter or symbol in the plaintext is replaced consistently by another letter or symbol in the ciphertext.

There are several types of substitution ciphers:

1. **Caesar Cipher (Shift Cipher):** One of the simplest substitution ciphers where each letter in the plaintext is shifted a certain number of places down or up the alphabet. For instance, with a shift of 3, 'A' becomes 'D,' 'B' becomes 'E,' and so on.

   Example: With a shift of 3: "HELLO" becomes "KHOOR." Each letter is shifted three places forward. To decrypt, reverse the shift.

2. **Mono-alphabetic Cipher:** In this cipher, each letter in the plaintext is replaced by a fixed substitution throughout the entire message. The most common example is the Atbash cipher, where letters are substituted in reverse order (e.g., 'A' becomes 'Z,' 'B' becomes 'Y,' and so forth).

   Example: Encrypting "HELLO" using Atbash turns it into "SVOOL." Each letter is replaced with its counterpart in reverse order. Decryption follows the same process, switching the letters back to reveal the original message.

3. **Poly-alphabetic Cipher:** Unlike the mono-alphabetic cipher, this type uses multiple substitution alphabets. The most well-known poly-alphabetic cipher is the Vigenère cipher, which uses a keyword to determine different alphabets for the substitution. Each letter in the plaintext is shifted based on the corresponding letter in the keyword.

   Example: Encrypting "HELLO" with a keyword "KEY" involves aligning the keyword under the plaintext and shifting each letter based on the corresponding letter in the keyword. Using 'K' for 'H,' 'E' remains 'E,' 'Y' for 'L,' 'K' for 'L,' 'E' for 'O,' the encrypted message becomes "KEMNK." Decryption follows a reverse process, using the keyword to reveal the original text.

4. **Homophonic Substitution Cipher:** In this cipher, each plaintext letter can be replaced by multiple different ciphertext characters. This technique introduces ambiguity by assigning more than one cipher symbol to some plaintext letters, making it harder to break.

   Example: 'A' might represent '3' or '7,' 'B' could be '1' or '9,' and so on. Encrypting "HELLO" could yield "7 4 12 12 15," where each letter has various replacements.

5. **Playfair Cipher:** This cipher uses a 5x5 matrix of letters to encrypt digraphs (pairs of letters) in the plaintext. Each pair of letters is substituted according to certain rules based on their positions in the matrix.

   Example: encrypting "HELLO" involves using a keyword like "KEYWORD" to create the matrix. First, arrange the letters excluding duplicates in the matrix, filling in the remaining spaces with the rest of the alphabet. Encrypting "HELLO" might yield "DGIIV." Each pair is encrypted based on their positions in the matrix, following specific rules within the grid.

Substitution ciphers are relatively easy to understand and use but are generally not as secure as more complex encryption methods. However, they serve as fundamental

examples of encryption techniques and are often used in educational contexts to illustrate the basic principles of cryptography.

1.4 ONETIME PAD

The *One-Time Pad* (OTP) is a highly secure form of encryption if executed correctly. It involves using a random and secret key that's at least as long as the plaintext message. Each character in the plaintext is combined with the corresponding character in the key to produce the ciphertext. The key is used only once and must never be reused.

Example

Let's say the plaintext message is "HELLO" and the key is "MONEY".

1. Convert the plaintext and key into numbers (using, for instance, A=0, B=1, ..., Z=25):
   - Plaintext "HELLO" becomes [7, 4, 11, 11, 14].
   - Key "MONEY" becomes [12, 14, 13, 4, 24].
2. Add each number in the plaintext to the corresponding number in the key, *modulo 26*:
   - [7, 4, 11, 11, 14] + [12, 14, 13, 4, 24] = [19, 18, 24, 15, 12].
3. Convert the resulting numbers back to letters:
   - [19, 18, 24, 15, 12] corresponds to "TSYPM" in the alphabet.

The encrypted message using the one-time pad is "TSYPM".

Decryption involves subtracting the key from the ciphertext, returning to the original plaintext.

1. Convert the ciphertext to plaintext by:
   - Ciphertext "TSYPM" is [19, 18, 24, 15, 12]
   - Key "MONEY" is [12, 14, 13, 4, 24].
2. Subtract each number in the ciphertext to the corresponding number in the key,
   - [19, 18, 24, 15, 12] - [12, 14, 13, 4, 24] = [7, 4, 11, 11, -12]= [7, 4, 11, 11, 14].
3. Convert the resulting numbers back to letters:
   - [7, 4, 11, 11, 14].corresponds to "HELLO" in the alphabet.

Advantages of the one-time pad (OTP):

1. *Perfect Secrecy:* When properly implemented with a truly random key used only once, OTP offers unbreakable encryption and perfect secrecy.

2. *Mathematical Simplicity:* Conceptually simple, involving straightforward modular addition or XOR operations.
3. *No Pattern Recognition:* It lacks patterns or repetitions, making it resistant to known-plaintext attacks or statistical analysis.

Disadvantages of the one-time pad (OTP):

1. *Key Management:* Requires generating, securely distributing, and storing keys of the same length as the plaintext, which poses logistical challenges.
2. *Key Security:* Any compromise or reuse of the key compromises the entire encryption scheme.
3. *Practicality:* Impractical for large-scale use due to the necessity for large random keys, making it less feasible for everyday communication.

Applications of the one-time pad (OTP):

The one-time pad (OTP) finds applications in highly secure communications:

1. *Military Communication:* Used for top-secret military and intelligence communication due to its unparalleled security.
2. *Diplomatic Correspondence:* Utilized in diplomatic missions where utmost secrecy is essential.
3. *Highly Sensitive Data:* Applied in scenarios requiring the utmost confidentiality, such as financial transactions or critical infrastructure communication.

1.5 INTRODUCTION TO CRYPTOGRAPHY PROTOCOLS

A ***protocol*** is simply a set of rules or instructions that govern communication between devices or systems in a network or computing environment. ***Cryptography protocols*** **are sets of rules governing secure communication, ensuring confidentiality, integrity, authentication, and non-repudiation in data transmission and storage.**

Cryptography protocols are like secret codes that keep information safe when it's shared or stored. Their main job is to protect important data from unauthorized access or changes These protocols use special techniques to scramble data into a secret code (encryption) tha only the intended recipient with the right key can unlock (decryption). They ensure tha when you send a message or make a transaction online, nobody else can snoop on it or alte it. Imagine sending a secret message in a locked box only you and your friend have keys fo One key locks the box (encryption), and the other opens it (decryption). These protoco ensure your message stays private and unaltered.

Example:

Let's say Alice wants to send the message "HELLO" to Bob securely. Using a cryptography protocol like *Advanced Encryption Standard* (AES), she encrypts the message with a secret key, turning "HELLO" into something like "XJFOP." She sends "XJFOP" to Bob, who decrypts it using the same secret key and protocol, turning "XJFOP" back into "HELLO."

This process ensures that even if someone intercepts the "XJFOP" message during transmission, they cannot understand it without the secret key for decryption, maintaining the confidentiality of the communication between Alice and Bob.

Objective:

The primary objective of cryptography protocol is to protect data from unauthorized access or alteration while in transit or at rest. These protocols encompass a structured set of rules, algorithms, and procedures that employ cryptographic techniques to ensure confidentiality, integrity, authenticity, and non-repudiation of data. They establish authentication mechanisms to verify the identities of parties involved in communication, preventing impersonation or unauthorized access. For instance, when browsing securely on the internet (like buying online), HTTPS uses cryptographic protocols to encrypt data sent between your browser and the website, keeping your personal details safe from eavesdroppers. These rules help protect secrets, verify identities, and make sure information stays secure during communication.

Purpose of Cryptography Protocol

The purpose of cryptography protocols are:

- **Securing Information:** Cryptography protocols use secret codes to keep information safe from unauthorized access or changes.
- **Encryption and Decryption:** They turn readable information into a secret code (encryption) and back into readable form (decryption) using special keys.
- **Protecting Communication:** These protocols ensure that messages or data sent over the internet remain private and cannot be altered by unauthorized individuals.
- **Establishing Trust:** By encrypting data, they create secure connections between devices or systems, ensuring trust and confidentiality in online transactions or communications.

- **Preventing Hacking:** Cryptography protocols safeguard sensitive details (like credit card information) during online activities, preventing hackers from stealing or tampering with data.
- **Verifying Identities:** They confirm the identities of people or systems involved in communication, ensuring you're interacting with the right person or website.
- **Maintaining Data Integrity:** These protocols ensure that data remains unchanged during transmission, preventing unauthorized modifications or tampering.
- **Enabling Secure Transactions:** They make online transactions secure by encrypting sensitive information, allowing you to shop or bank online safely.

1.6 TYPES OF CRYPTOGRAPHY

Cryptography is a technique of secret writing especially code and cipher systems, method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. The term is most often associated with scrambling plaintext (ordinary text, sometimes referred to as *cleartext*) into *ciphertext* (a process called encryption), then back again (known as decryption).

As the Internet and other forms of electronic communication become more prevalent, electronic security is becoming increasingly important. Cryptography is used to protect e-mail messages, credit card information, and corporate data.

Cryptography encompasses various types of techniques and methods to secure data and communications.

The primary types include:

1. **Private Key Cryptography (or "Symmetric Cryptography"):** It uses a single secret key for both encryption and decryption, efficient for bulk data encryption.
2. **Public Key Cryptography (or "Asymmetric Cryptography"):** It employs a pair of keys for encryption and decryption, facilitating secure communication and key exchange without sharing secret keys.
3. **Hash Functions:** It convert data into fixed-size hash values, crucial for ensuring data integrity, digital signatures, and password storage.
4. **Digital Signatures:** It verifies the authenticity and integrity of digital messages using asymmetric cryptography, ensuring the signer's identity.

In the next section, we will discuss all these cryptography types in detail.

1.7 PRIVATE KEY CRYPTOGRAPHY (SYMMETRIC CRYPTOGRAPHY)

A private key cryptography (Symmetric-key system) is a method in which the same key is used to encrypt and decrypt the message. In private-key cryptography, sender and the recipient of the message must agree on a common key via some alternative secure channel.

Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Public-key encryption avoids this problem because the public key can be distributed in a non-secure way, and the private key is never transmitted.

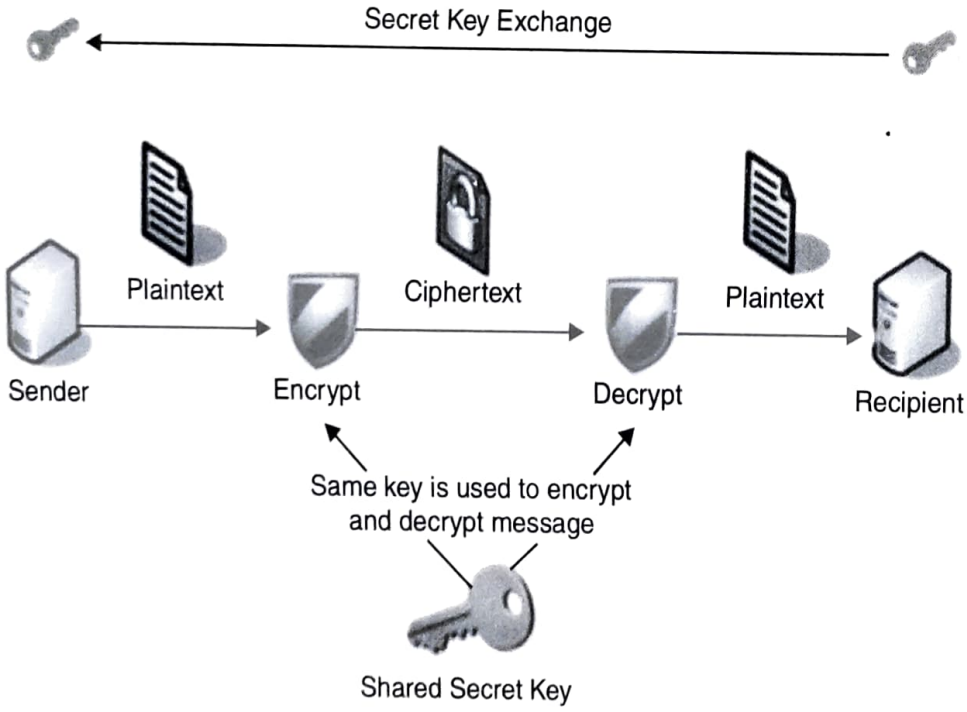


Fig 1.1 Symmetric Cryptography

Private-key encryption shown in **Fig 1.1** involves the following steps:

1. The sender creates a ciphertext message by encrypting the plaintext message with a symmetric encryption algorithm and a shared key.
2. The sender sends the ciphertext message to the recipient.
3. The recipient decrypts the ciphertext message back into plaintext with a shared key.

Implementations of symmetric-key encryption can be highly efficient, so that users do not experience any significant time delay as a result of the encryption and decryption. Symmetric-key encryption also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense.

Example:

Let's consider the encryption of the word "HELLO" using a symmetric key and a simple substitution cipher:

1. **Key Generation:** Alice and Bob agree on a secret key, let's say "KEY" for this example.
2. **Encryption:**
   - Alice wants to send "HELLO" to Bob securely.
   - She applies a simple substitution cipher using the shared key "KEY." Each letter is shifted by a certain value according to the key.
   - "H" becomes "K," "E" becomes "H," "L" becomes "O," and "O" becomes "R."
   - So, "HELLO" encrypted with the key "KEY" becomes "KHORR."
3. **Transmission:** Alice sends "KHORR" to Bob.
4. **Decryption:**
   - Bob receives "KHORR" and knows the shared key is "KEY."
   - Using the same key and decryption process, he reverses the substitution cipher.
   - "K" becomes "H," "H" becomes "E," "O" becomes "L," and "R" becomes "O."
   - So, "KHORR" decrypted with the key "KEY" becomes "HELLO."

1.7.1 Advantages of Symmetric Cryptography

The key advantages of symmetric cryptography are:

- **Speed and Efficiency:** Symmetric cryptography is notably fast and computationally efficient, ensuring swift encryption and decryption processes.
- **Lower Computational Overhead:** Algorithms involved in symmetric key operations

are less complex, demanding fewer computational resources compared to asymmetric cryptography.

- **Smaller Key Size:** Symmetric keys are shorter in length than asymmetric keys, making key management, storage, and transmission more manageable and faster.
- **Suitable for Bulk Data Encryption:** Ideal for securing large volumes of data during transmission or storage, such as encrypting files, databases, and communication channels.
- **High Performance:** Offers superior performance due to its efficiency, crucial in applications demanding rapid encryption and decryption capabilities.

1.7.2 Disadvantages of Symmetric Cryptography

The key disadvantages of symmetric cryptography are:

- **Key Distribution:** Securely sharing keys between parties poses a significant challenge, as any interception compromises security.
- **Scalability:** As the number of communicating parties increases, the number of required keys grows quadratically, complicating key management.
- **Lack of Non-Repudiation:** Symmetric encryption does not inherently provide non-repudiation, making it challenging to verify the sender's identity.
- **Key Exchange Challenge:** Establishing a secure key exchange method without interception remains a critical hurdle in symmetric cryptography.
- **Single Key Vulnerability:** The entire security of the system relies on safeguarding a single shared key, increasing susceptibility to key compromise.

1.8 PUBLIC-KEY CRYPTOGRAPHY (ASYMMETRIC CRYPTOGRAPHY)

Public-key cryptography, also known as *asymmetric cryptography* is a class of cryptographic system that uses two keys - a *public key* and *a private key*. A public key is known to everyone while *private* (or *secret key*) is known only to the recipient of the message.

The term "asymmetric" stems from the use of these two keys to perform these opposite functions. The public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. The public key encrypt plaintext or to verify a digital signature; whereas the private key decrypt ciphertext or to create a digital signature.

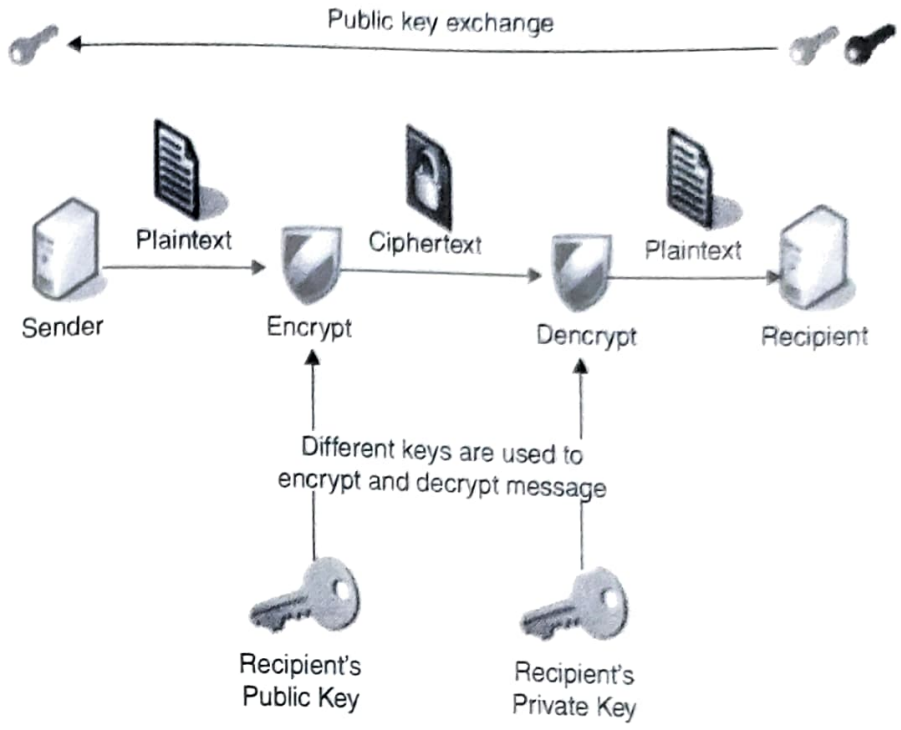


Fig 1.2 Public Key Encryption

As illustrated in **Figure 1.2**, asymmetric encryption involves the following steps:

1. The sender creates a ciphertext message by encrypting the plaintext message with an asymmetric encryption algorithm and the recipient's public key.
2. The sender sends the ciphertext message to recipient.
3. The recipient decrypts the ciphertext message back to plaintext using the private key that corresponds to the public key that was used to encrypt the message.

With public key encryption, the sender converts the plaintext message into ciphertext by encrypting it with the public key in the message recipient's X.509 certificate. The message recipient converts the ciphertext back into the plaintext message by decrypting it with the corresponding private key.

The basic motivation of using Public Key Cryptography is to send messages in such a way that only the person who receives them can understand them. By using public key encryption, a message sender has assurance that only the recipient will be able to read the message.

Example:

Here's an example to illustrate public key cryptography:

Imagine Alice wants to send a secure message to Bob using public key cryptography:

1. **Key Generation:**
   - Bob generates a pair of keys: a public key and a private key.
   - The public key is shared openly, while the private key is kept secret.
2. **Encryption:**
   - Alice encrypts her message using Bob's publicly available encryption (public) key.
   - Only Bob's private key, which only Bob possesses, can decrypt the message encrypted with his public key.
3. **Transmission:**
   - Alice sends the encrypted message to Bob.
4. **Decryption:**
   - Bob receives the encrypted message and uses his private key to decrypt it.
   - Bob is the only one who can decrypt the message since only his private key can undo the encryption performed with his public key.

1.8.1 Advantages of Public Key Cryptography

The key advantages of public key cryptography are:

- **Security:** Provides a robust security framework by using a pair of keys, ensuring confidentiality and integrity without requiring a shared secret between parties.
- **Key Distribution:** Eliminates the need for secure key exchange as only public keys are distributed openly while keeping private keys confidential.
- **Authentication:** Enables verification of sender authenticity through digital signatures, ensuring the message's origin and integrity.
- **Non-Repudiation:** Offers proof of message origin due to the unique association between private keys and digital signatures, preventing senders from denying sending a message.
- **Scalability:** Allows secure communication among multiple parties without increasing key management complexities.
- **Versatility:** Forms the basis for secure protocols (like SSL/TLS), secure email, digital signatures, secure transactions, and authentication mechanisms across various digital applications.

- **Efficiency:** Facilitates secure key exchange for symmetric encryption, improving efficiency in encrypted communication once initial secure connections are established.

1.8.2 Disadvantages of Public Key Cryptography

The key disadvantages of public key cryptography are:

- **Computational Intensity:** Public key operations (encryption, decryption) are slower and more computationally demanding than symmetric key cryptography.
- **Key Size Overhead:** Public keys are larger than symmetric keys, increasing storage and transmission requirements.
- **Key Management Complexity:** Managing and safeguarding private keys requires robust security measures to prevent unauthorized access or loss.

1.9 ONE-WAY HASH FUNCTIONS

One fundamental cryptographic tool is a one-way hash function, which is a mathematical algorithm that takes an input (or message) and produces a fixed-size string of characters, called a hash value or digest. The input to the hash function is of arbitrary length but output is always of fixed length.

The following picture illustrated hash function:

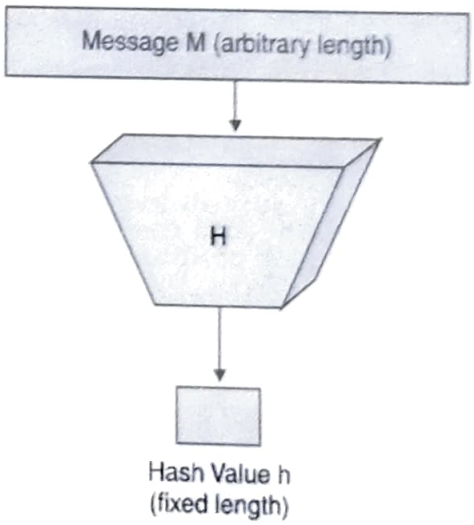


Fig. 1.3 Hash Function

One-way hash functions have three main properties:

- **Deterministic:** For a given input, the hash function always produces the same output.

- **Fixed Output Size:** Regardless of the input size, the hash function generates a fixed-size output.
- **Non-Reversible:** It is computationally infeasible to reverse the hash function to obtain the original input from the hash value.

Cryptographic hash functions work by generating the checksum value of a data object. Checksum is a value derived from data to ensure integrity and detect errors accurately. If the data is intentionally or unintentionally modified, the checksum value is changed. Thus, a data object's integrity may be evaluated by comparing and verifying previous and current checksums.

Example:

An example of a popular one-way hash function is the *Secure Hash Algorithm* 256 (SHA-256). Let's say we have a simple message, "Hello, World!" that we want to hash using SHA-256. The SHA-256 algorithm will generate a unique 256-bit (32-byte) hash value for the given input. The hash value for "Hello, World!" using SHA-256 would be:

"2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824."

Now, here's the important part: you can't reverse this process. Once "hello" turns into that code, you can't get "hello" back from the code. It's a one-way street!

1.9.1 Properties of One Way Hash Functions

The ideal hash function has three main properties:

1. Pre-Image Resistance

This property means that it is very difficult to get the original message back from the digest. For example, if a hash function h produced a hash value m, then it should be a difficult process to find any input value m that hashes to z. This property protects against an attacker who only has a hash value and is trying to find the input.

2. Second Pre-Image Resistance

This property means given an input and its hash, it should be hard to find a different input with the same hash. For example, if a hash function h for an input m produces hash value h(m), then it should be difficult to find any other input value y such that h(y) = h(m). This property of hash function protects against an attacker who has an input value and its hash, and wants to substitute different value as legitimate value in place of original input value.

3. Collision Resistance

This property means it is infeasible to find two messages with the same hash. For example, for a hash function h, it is hard to find any two different inputs x and y such that $h(x) = h(y)$. Since, hash function is compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find. This property makes it very difficult for an attacker to find two input values with the same hash.

1.9.2 Popular One Way Hash Functions

The most popular hash functions are:

1. Message Digest (MD)

The MD5 algorithm is a widely used hash function producing a 128-bit hash value. MD5 is used in many situations where a potentially long message needs to be processed and/or compared quickly. The most common application is the creation and verification of digital signatures. The MD5 hash function was originally designed for use as a secure cryptographic hash algorithm for authenticating digital signatures. MD5 has been deprecated for uses other than as a non-cryptographic checksum to verify data integrity and detect unintentional data corruption.

2. Secure Hashing Algorithms

Secure Hashing Algorithms, also known as SHA, is another popular cryptographic hash functions designed to keep data secured. The SHA algorithm consists of bitwise operations, modular additions, and compression functions. These algorithms are designed to be one-way functions, meaning that once they're transformed into their respective hash values, it's virtually impossible to transform them back into the original data. A few algorithms of interest are SHA-1, SHA-2, and SHA-5, each of which was successively designed with increasingly stronger encryption in response to hacker attacks.

1.9.3 Benefits of One Way Hash Functions

The benefits of One Way Hash Functions are:

- **Data Integrity Assurance:** One-way hash functions ensure data remains unchanged by generating unique fixed-size codes for any input, detecting alterations.
- **Password Security:** Safely stores passwords by hashing them, preventing direct access to the original password, enhancing security against unauthorized access.

- **Efficient Verification:** Enables quick comparison of hashed values for authentication, ensuring data authenticity without revealing the original content.
- **Digital Signatures:** Facilitates the creation of secure digital signatures, verifying document integrity and sender authenticity through hashed values and encryption.
- **Reduced Vulnerability:** Protects sensitive information by making it computationally infeasible to reverse-engineer the original data from its hash, enhancing overall security.

1.9.4 Message Authentication from Hash Function

Message authentication provides a way to ensure message integrity. It ensures that the message has been sent by a genuine identity and not by an imposter. Message authentication can be provided using the cryptographic techniques that use secret keys.

Message authentication is typically achieved by using *Message Authentication Codes* (MACs). MAC is realized with cryptographic hash functions which includes the symmetric key between the sender and receiver when creating the digest.

A MAC requires two inputs: a message and a secret key known only to the originator of the message and its intended recipient(s). This allows the recipient of the message to verify the integrity of the message and authenticate that the message's sender has the shared secret key. If a sender doesn't know the secret key, the hash value would then be different, which would tell the recipient that the message was not from the original sender.

The process of using MAC for authentication is depicted in the following illustration:

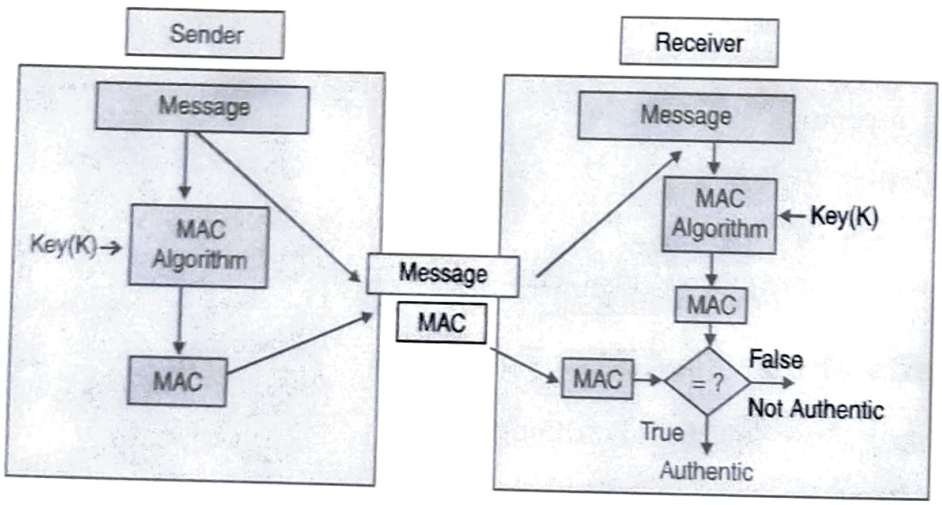


Fig 1.4 MAC Process

Let us now try to understand the entire process in detail:

- The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.

- Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and MAC is that MAC uses secret key during the compression.
- The sender forwards the message along with the MAC.
- On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
- The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures that the message has been sent by the intended sender.
- If the computed MAC does not match the MAC sent by the sender, the receiver safely assumes that the message is not the genuine.

1.10 DIGITAL SIGNATURE

A digital signature authenticates electronic documents in a similar manner like a handwritten signature authenticates printed documents. This signature cannot be faked and it claims that a named person wrote or otherwise agreed to the document to which the signature is attached. The recipient of a digitally signed message can verify that the message originated from the person whose signature is attached to the document and that the message has not been altered either intentionally or accidentally since it was signed. Also the signer of a document cannot later disagree with it by claiming that the signature was forged. In other words digital signatures enable the authentication of digital messages assuring the recipient of a digital message of both the identity of the sender and the integrity of the message.

In short, it is used:

- To ensure message content integrity.
- To verify the authenticity of the message sender.

Example:

Imagine you're sending a secret letter to your friend, but you want to make sure it doesn't get tampered with and your friend knows it's really from you.

1. **Creating the Digital Signature:** First, you write your letter. Then, you create a unique code that represents your letter. This code is like a secret mark only you can create.

2. **Using Your Secret Code:** You use your secret code to "lock" your letter. This is your digital signature. Your letter is now locked, and no one can change it without breaking the lock.
3. **Sending the Letter:** You send the locked letter to your friend.
4. **Verification by Your Friend:** Your friend receives the locked letter and sees your digital signature. They use a special key (your public key) to unlock the signature. If it unlocks successfully, they get the code that represents your original letter.
5. **Checking Integrity:** Your friend checks the code they got by unlocking the signature. If it matches the code they create from the letter, they know the letter hasn't been changed and is genuinely from you.

In this way, a digital signature works like a lock and key. You create a unique lock (the signature) only you can make, and your friend uses a key (your public key) to unlock it and check if the letter is still the same as when you sent it. This ensures that your message remains secure, unaltered, and authentic during transmission.

1.10.1 Digital Certificate

A digital certificate is an electronic certificate attached to electronic message for security purpose. It is a digital form of identification, much like a passport or driver's license in the physical world. It serves to validate the authenticity of digital entities such as individuals, organizations, or websites in the online realm. It allows a person, computer or organization to exchange information securely on the web. A digital certificate ensures that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

A Digital Certificate typically contains:

- Owner's public key
- Owner's name
- Expiration date of the public
- Name of the issuer (the CA that issued the Digital Certificate)
- Serial number of the Digital Certificate
- Digital signature of the issuer

A digital certificate is issued by a *Certification Authority* (CA). When a user wants to send an encrypted message, he applies for a digital certificate from a Certificate Authority *(CA)*.

The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available through print publicity or perhaps on the Internet.

The recipient of an encrypted message uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA and then obtains the sender's public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply. Digital certificates are used with self-signatures and message encryption.

An X.509 certificate is a digital certificate that uses the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.

1.11 DIGITAL SIGNATURES WITH ENCRYPTION

Digital signatures are based on public key cryptography, also known as asymmetric cryptography. It is a class of cryptographic system that uses two keys - a ***public key*** and a ***private key***. The private key is used to create a signature, and the corresponding public key is used to verify the signature. A public key is known to everyone while ***private*** (or ***secret key***) is known only to the recipient of the message.

The term "asymmetric" stems from the use of these two keys to perform these opposite functions. The public and private keys are related in such a way that the private key is used to create a digital signature; whereas the public key is used to verify a digital signature.

Encryption:

- A Hashing algorithm is used to prepare a hash data.
- The hashed data is encrypted using the sender's private key.
- The encrypted hashed data is the digital signature.
- The digital signature and the sender's X.509 Digital Certificate are appended to the end of the message.
- Public keys are maintained in X.509 certificates, which are digital documents that bind a subject's identity claims to a public key from a public/private asymmetric key pair.

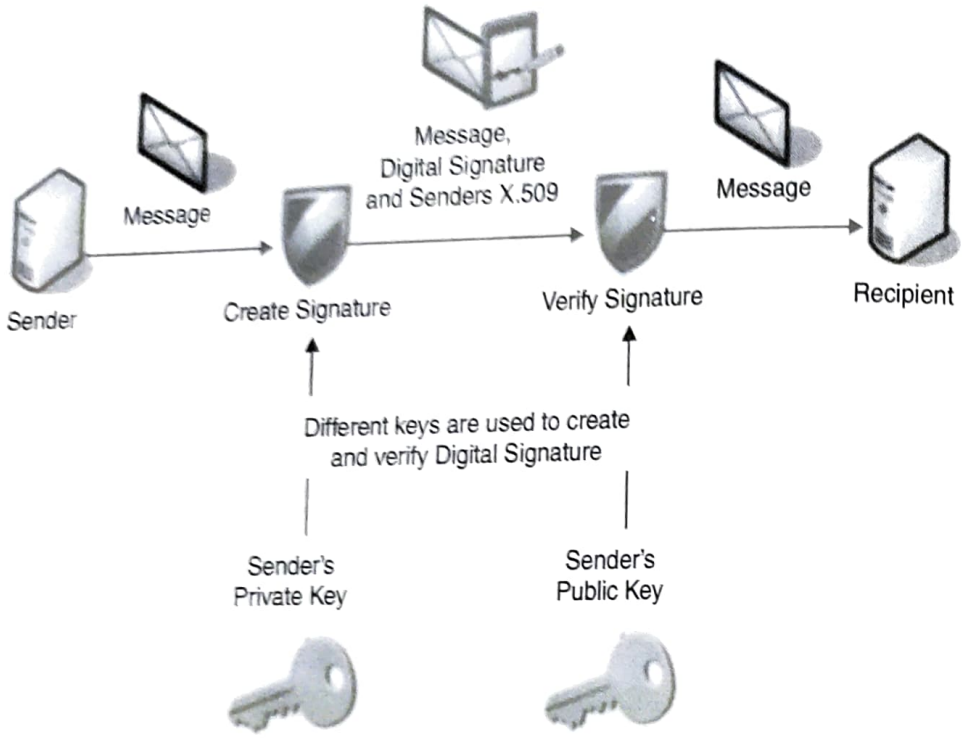


Fig 1.5 Digital Signature Process

Decryption:

- The receiver receives the message along with digital signature.
- The receiver separately calculates a hashed data for the received message.
- The receiver use the signer's public key to decrypt the hash.
- If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn't changed since it was signed.
- If the two hashes do not match, the data may have been tampered with or the data may not be authentic or may not have been intended for the receiver.

1.11.1 Advantages of Digital Signature

The advantages of digital signature are:

- **Authentication:** It ensures the identity of the sender, verifying the authenticity of the signer, and preventing impersonation or unauthorized access.
- **Integrity:** It guarantees that the signed content has not been altered or tampered with during transmission, maintaining its originality.
- **Non-Repudiation:** It provides evidence that the sender cannot deny sending the signed message or document, preventing disputes or false claims.

- **Security:** It protects sensitive information by using cryptographic techniques, safeguarding against unauthorized access and ensuring confidentiality.
- **Efficiency:** It streamlines processes by enabling secure electronic transactions, reducing paperwork, and eliminating the need for physical signatures.
- **Global Acceptance:** It is internationally recognized and legally binding in many jurisdictions, fostering trust in digital communications and transactions.
- **Reduced Costs:** It saves expenses associated with paper-based signing, postage, and manual processing, leading to cost-efficiency in business operations.
- **Compliance:** It helps organizations meet regulatory requirements and standards by providing a secure and traceable method for digital transactions.

1.11.2 Applications of Digital Signature

The various applications of digital signatures are:

- **E-Signatures:** Used in digital contracts, agreements, and approvals, enabling legally binding electronic signatures.
- **Secure Transactions:** Ensures the integrity and authenticity of online financia transactions and banking activities.
- **Email Security:** Verifies the origin of emails and ensures message integrit protecting against phishing and spoofing.
- **Document Verification:** Authenticates digital documents, certificates, licenses, a records, ensuring their validity and preventing forgery.
- **Software Integrity:** Certifies the authenticity and integrity of software by sig code, preventing unauthorized modifications or malware insertion.
- **Government Services:** Used in digital governance, e-governance initiatives, official document signing for regulatory compliance and secure govern operations.

1.12 RANDOM AND PSEUDO-RANDOM SEQUENCE GENERAT

Random and pseudo-random sequence generation is method used in gen sequences of numbers or values. They differ in their ability to produce truly unpre sequences

Random Sequence Generation

Random sequence generation involves generating values that are

unpredictable and independent of previous values. Imagine flipping a fair coin. Each time you flip it, you either get "heads" or "tails". If you flip it without any patterns or biases, the results are random. This is an example of true randomness. Similarly, rolling a fair six-sided die and getting any number from 1 to 6 is another example of a random outcome.

It is essential in cryptography, simulations, gambling, and statistical sampling where unpredictability is crucial. Cryptographic keys need to be unpredictable to resist attacks. Random sequence generation ensures the creation of strong and secure cryptographic keys used in symmetric and asymmetric encryption schemes.

Pseudo-Random Sequence Generation (PRSG)

Pseudo-randomness involves generating sequences that appear random but are actually generated using algorithms and a seed value. Pseudo-random number generators (PRNGs) produce sequences that mimic randomness but are deterministic—given the same initial state (seed), they'll produce the same sequence.

Think of a simple game where you start at a number, say 1, and then add 3 to it repeatedly: 1, 4, 7, 10, 13, and so on. This sequence appears random, but it's actually generated by following a fixed rule (adding 3 each time). If you know the starting point (seed), you can predict the entire sequence.

PRNGs are widely used in computer algorithms, simulations, gaming, and statistical modeling due to their speed and repeatability. Pseudo-random number generators (PRNGs) are used to create session keys or temporary values. Although deterministic, their unpredictability in generating sequences makes them suitable for various cryptographic operations.