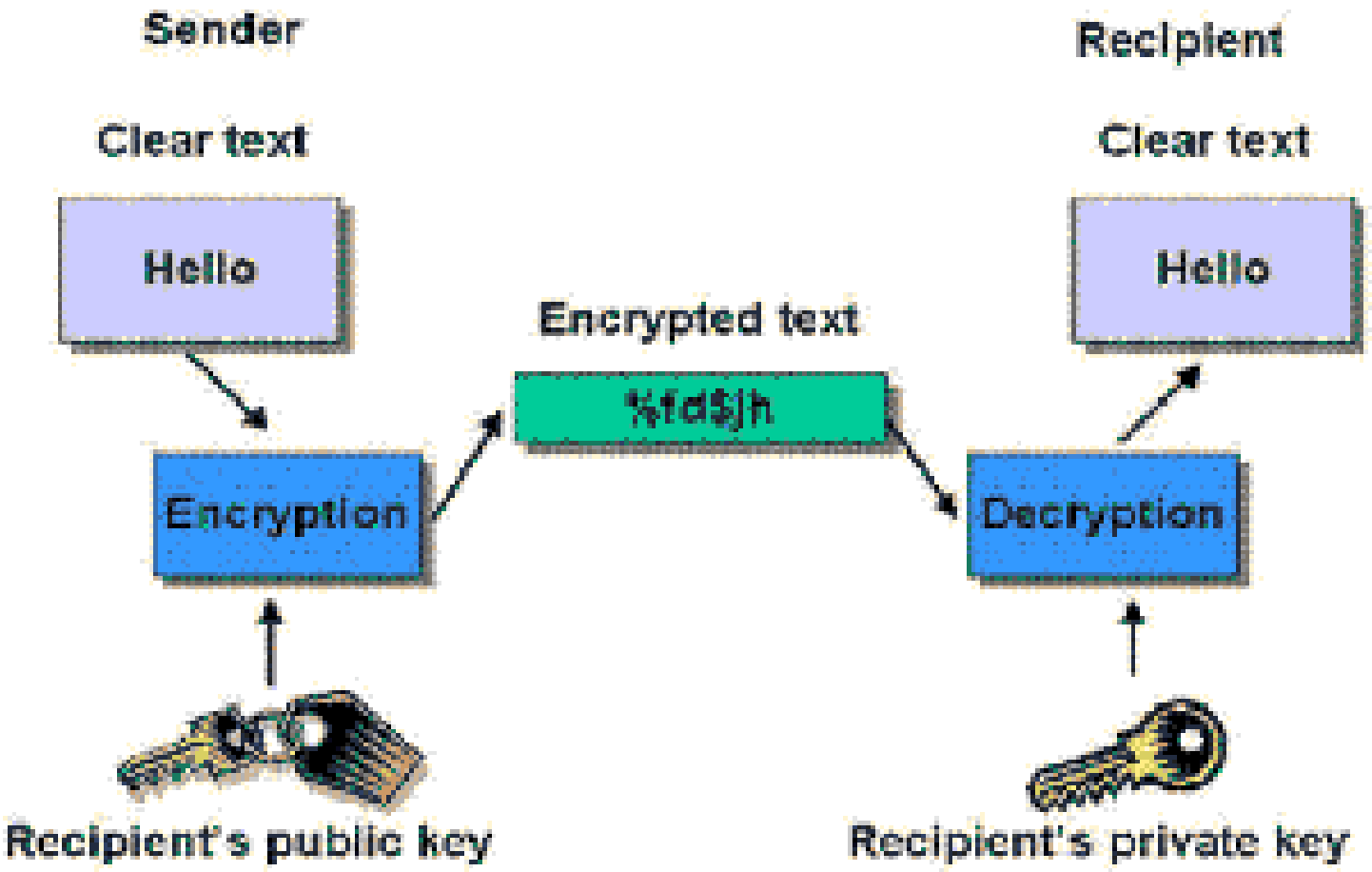
Public-key cryptography

Public-key cryptography

- Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses two mathematically related, but not identical, keys - a public key and a private key. Unlike symmetric key algorithms that rely on one key to both encrypt and decrypt, each key performs a unique function.
- The public key is used to encrypt and the private key is used to decrypt.
- It is computationally infeasible to compute the private key based on the public key. Because of this, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures, and private keys can be kept secret, ensuring only the owners of the private keys can decrypt content.


Sender
Clear text
Hello
Encryption
Recipient's public key
Encrypted text
%fd$jh
Recipient
Clear text
Hello
Decryption
Recipient's private key

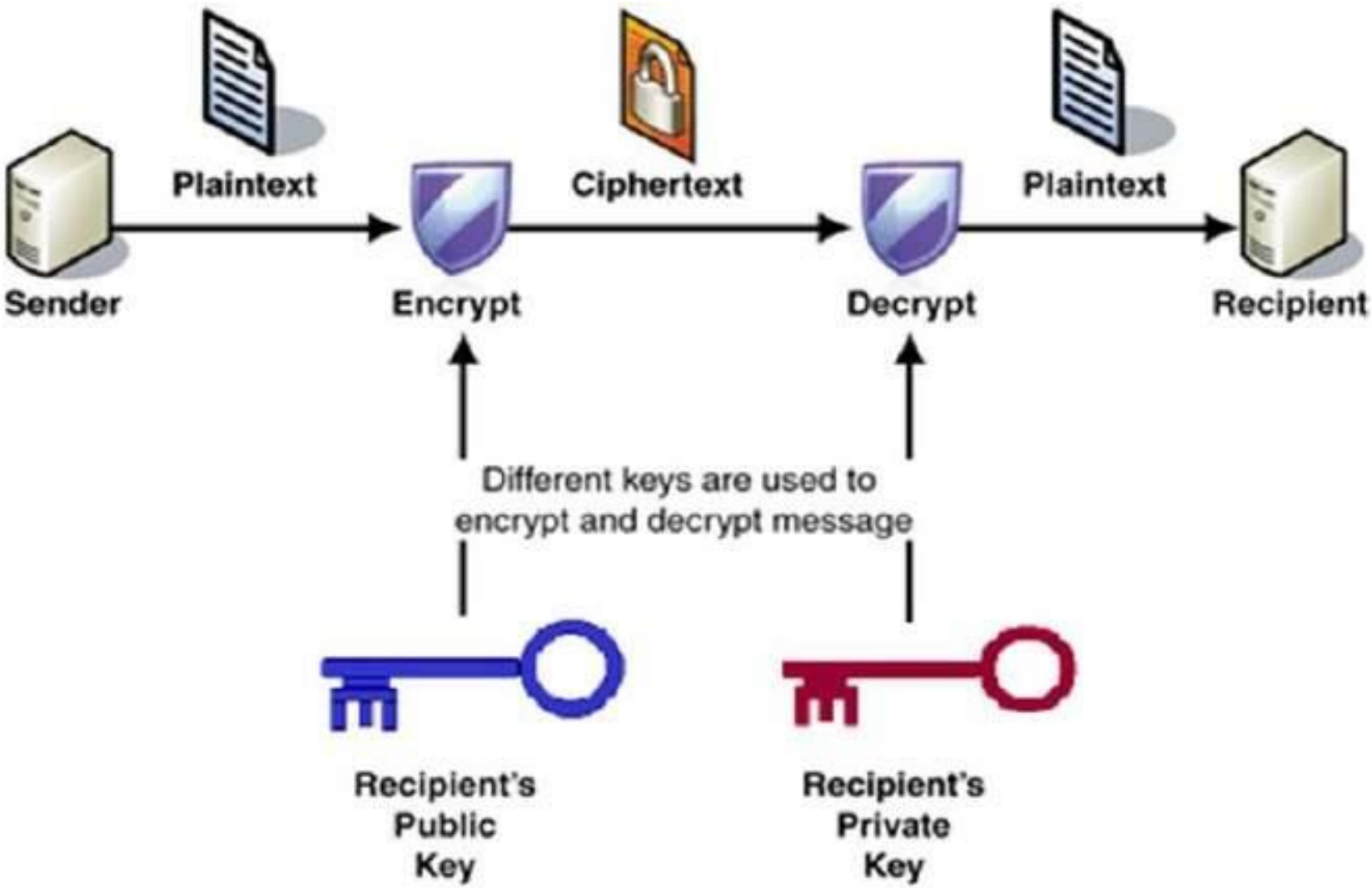
The most important properties of public key encryption scheme are −

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.

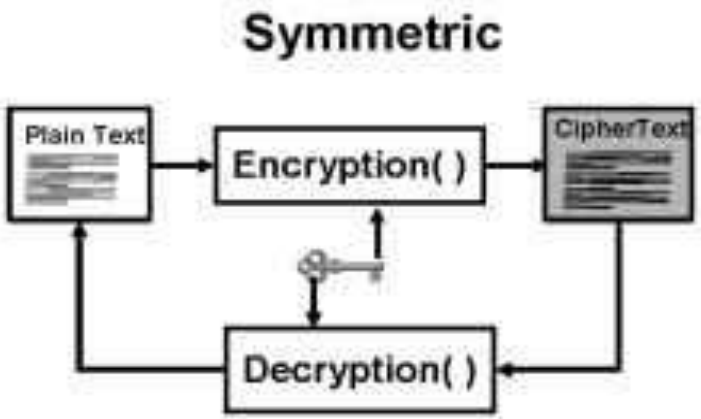
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key.

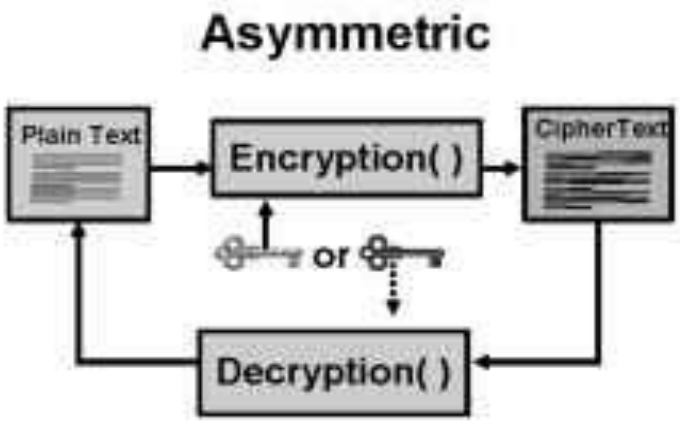
Public Key Encryption schemes.

- RSA Cryptosystem
- ElGamal Cryptosystem


Plaintext
Ciphertext
Plaintext
Sender
Encrypt
Decrypt
Recipient
Different keys are used to encrypt and decrypt message
Recipient's Public Key
Recipient's Private Key

Symmetric vs. Asymmetric Encryption Algorithms



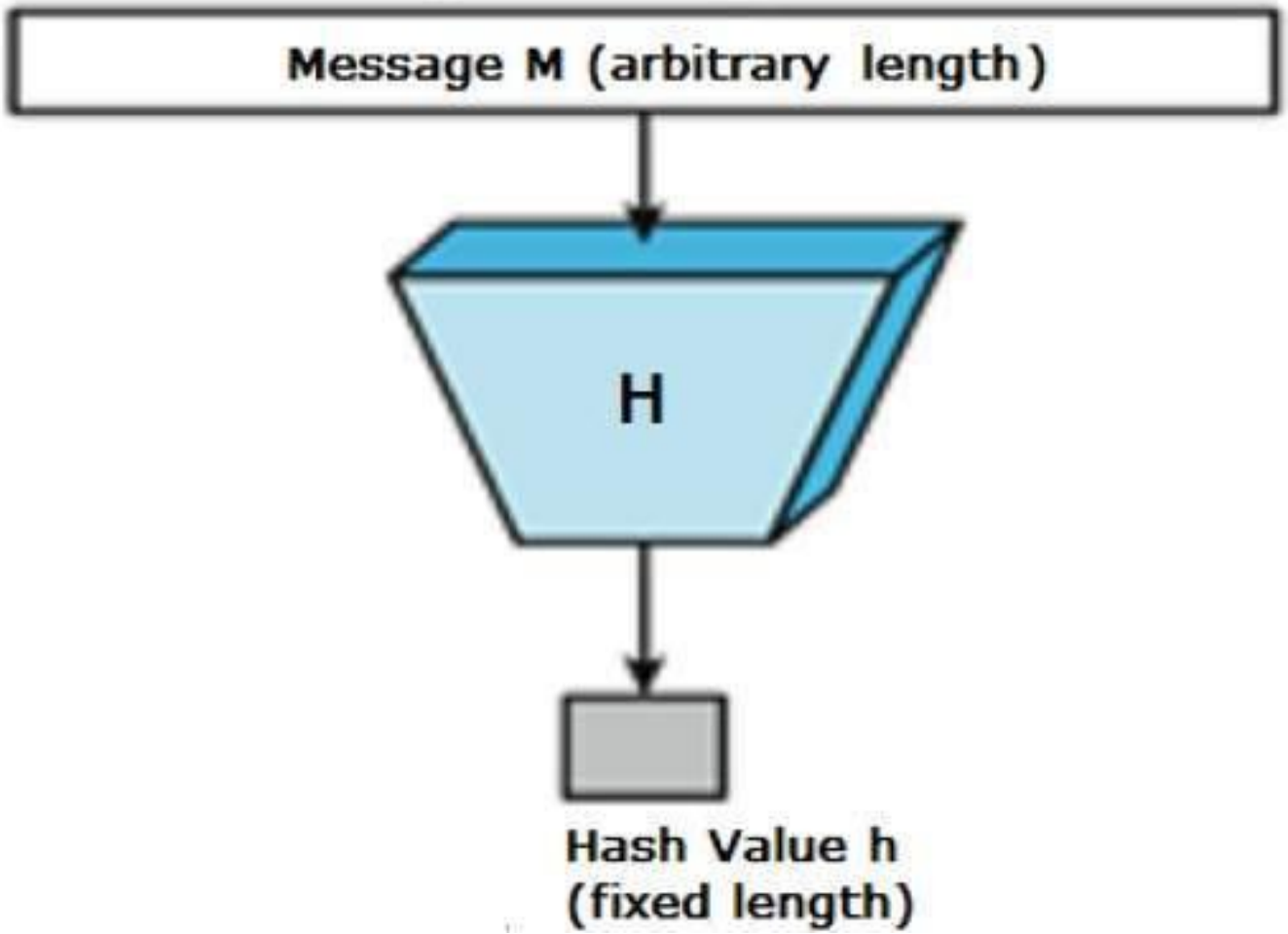


- Secret key cryptography
- Encryption and decryption use the same key
- Typically used to encrypt the content of a message
- Examples: DES, 3DES, AES

- Public key cryptography
- Encryption and decryption use different keys
- Typically used in digital certification and key management
- Example: RSA

Hashing

- Hash functions are extremely useful and appear in almost all information security applications.
- A hash function is a mathematical function that converts a numerical input value into another compressed numerical value.
- The input to the hash function is of arbitrary length but output is always of fixed length.
- Values returned by a hash function are called **message digest** or simply **hash values**. The following picture illustrated hash function


Message M (arbitrary length)
H
Hash Value h
(fixed length)

Features of Hash Functions

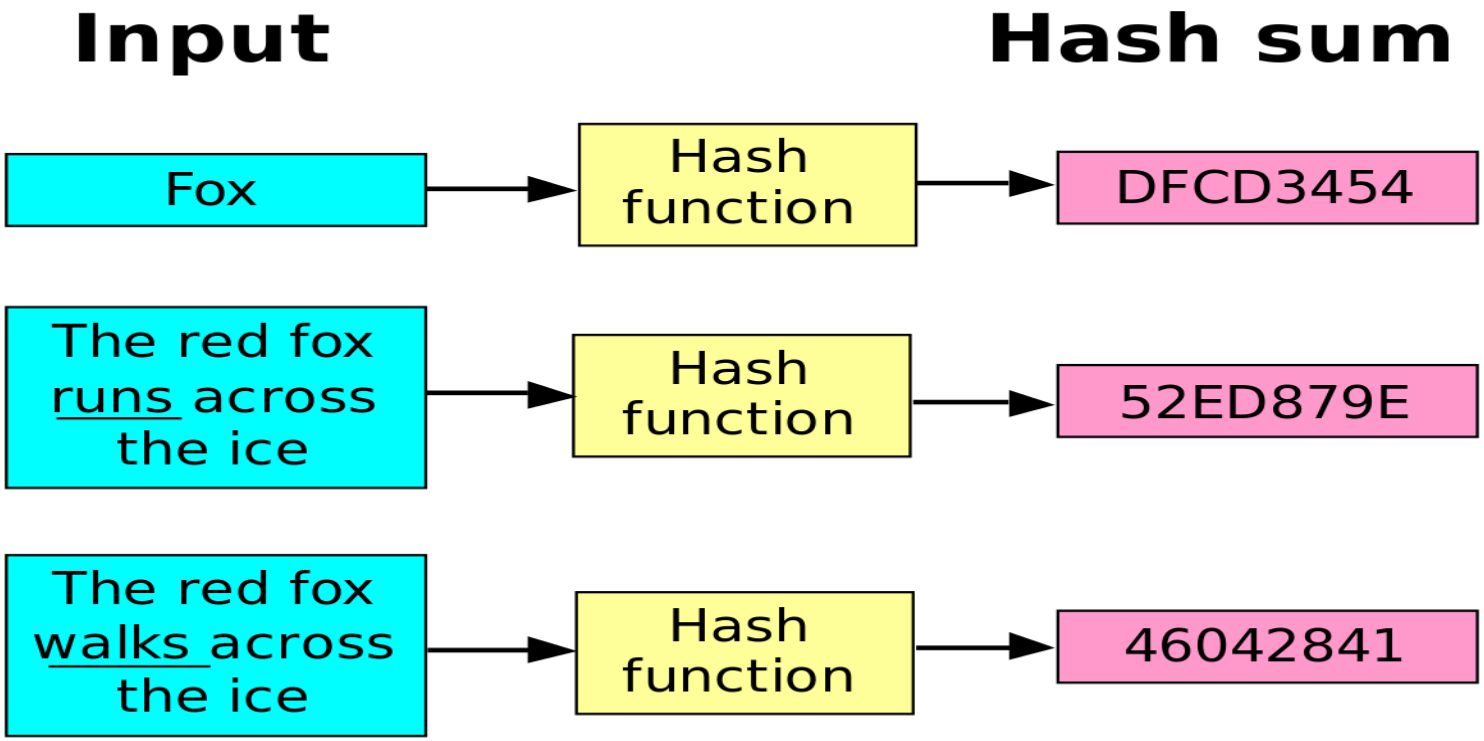
- **Fixed Length Output (Hash Value)**
  - Hash function coverts data of arbitrary length to a fixed length. This process is often referred to as **hashing the data**.
  - In general, the hash is much smaller than the input data, hence hash functions are sometimes called **compression functions**.
  - Since a hash is a smaller representation of a larger data, it is also referred to as a **digest**.
  - Hash function with n bit output is referred to as an **n-bit hash function**. Popular hash functions generate values between 160 and 512 bits.
- **Efficiency of Operation**
  - Generally for any hash function h with input x, computation of h(x) is a fast operation.
  - Computationally hash functions are much faster than a symmetric encryption.

Popular Hash Functions

- Message Digest (MD)
- Secure Hash Function (SHA)
- Whirlpool
- RIPEMD(RACE Integrity Primitives Evaluation Message Digest.)

Cryptographic hash functions have many information-security applications

- Digital signatures,
- Message Authentication Codes (MACs), and other forms of authentication.
- They can also be used as ordinary hash functions, to index data in hash tables, for fingerprinting, to detect duplicate data.

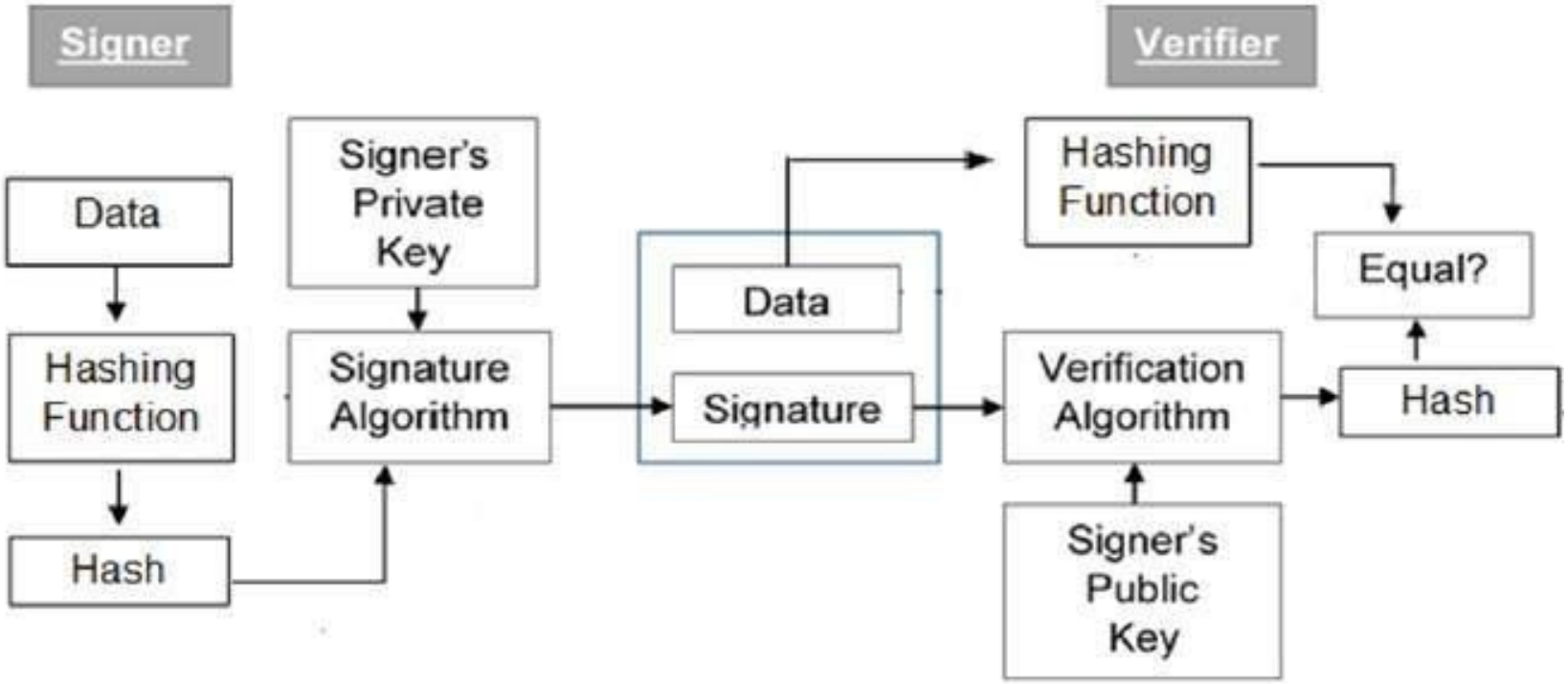

Input
Hash sum
Fox
Hash function
DFCD3454
The red fox runs across the ice
Hash function
52ED879E
The red fox walks across the ice
Hash function
46042841

Digital signature

- A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document.
- In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.
- Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.
- Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

Model of Digital Signature

- The digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration –



The following points explain the entire process in detail –

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.

- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by ‘private’ key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

How this is achieved by the digital signature –

- **Message authentication** – When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.
- **Data Integrity** – In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.
- **Non-repudiation** – it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

Public Key Infrastructure (PKI)

- Public Key Infrastructure (PKI) is that it uses a pair of keys to achieve the underlying security service. The key pair comprises of private key and public key.
- Since the public keys are in open domain, they are likely to be abused. It is, thus, necessary to establish and maintain some kind of trusted infrastructure to manage these keys.

Key Management

- It refers to management of cryptographic keys in a cryptosystem. This includes dealing with the generation, exchange, storage, use and replacement of keys.
- Successful key management is critical to the security of a cryptosystem.

Management steps

- Once keys are inventoried, key management typically consists of three steps: exchange, storage and use.

Key exchange

- In some instances this may require exchanging identical keys (in the case of a symmetric key system).

  In others it may require possessing the other party's public key. While public keys can be openly exchanged (their corresponding private key is kept secret), symmetric keys must be exchanged over a secure communication channel.

Public key Infrastructure (PKI)

↓

asymetric Encryption.

→ 3 different formats of msg can be used in public key crypto sys.

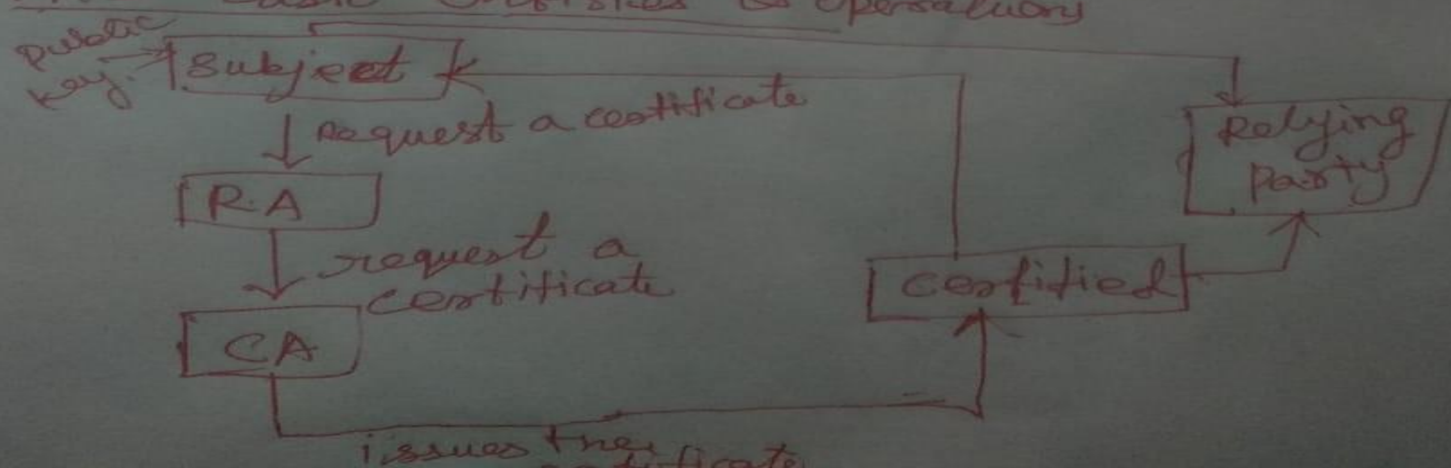
1. Encrypted msg
2. Signed msg.
3. signed & Encrypted msg.

⇒ **PKI Entities.**

1. CA (Certificate Authority)
2. RA (Registration Authority)
3. subscriber
4. Relying party.
5. Repository.

CA → not

PKI basic entities & operations



Key storage

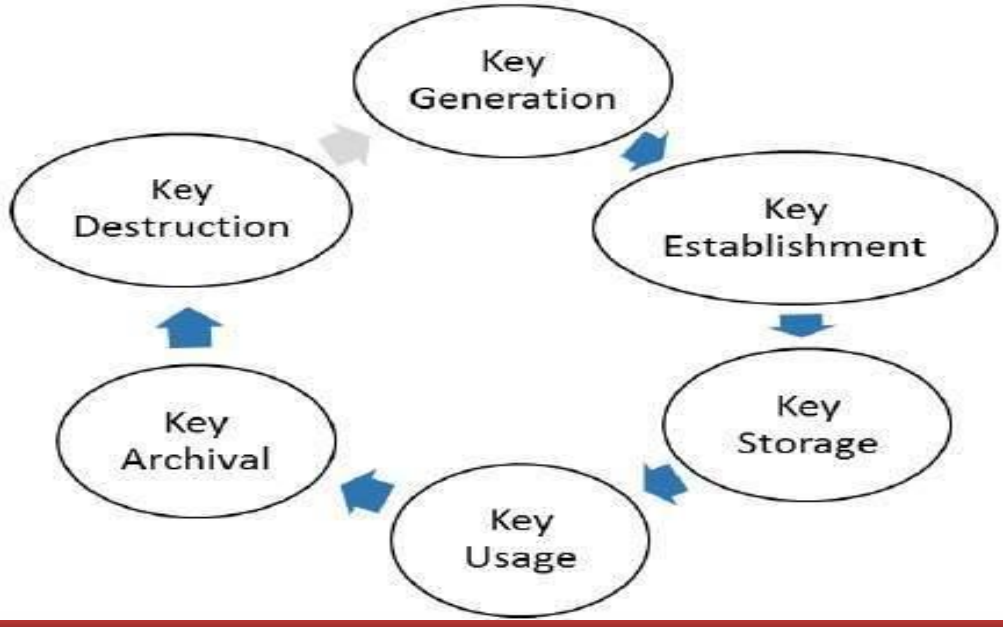
- However distributed, keys must be stored securely to maintain communications security.
- keys may be stored in a Hardware Security Module (HSM) or protected using technologies such as Trusted Execution Environment.

Key use

- The major issue is length of time a key is to be used, and therefore frequency of replacement. Because it increases any attacker's required effort, keys should be frequently changed. This also limits loss of information

There are some important aspects of key management which are as follows −

- Cryptographic keys are nothing but special pieces of data. Key management refers to the secure administration of cryptographic keys.
- Key management deals with entire key lifecycle as depicted in the following illustration −



There are two specific requirements of key management for public key cryptography.

- **Secrecy of private keys.** Throughout the key lifecycle, secret keys must remain secret from all parties except those who are owner and are authorized to use them.
- **Assurance of public keys.** In public key cryptography, the public keys are in open domain and seen as public pieces of data.
- By default there are no assurances of whether a public key is correct, with whom it can be associated, or what it can be used for. Thus key management of public keys needs to focus much more explicitly on assurance of purpose of public keys.

Public Key Infrastructure (PKI)

- PKI provides assurance of public key. It provides the identification of public keys and their distribution. An anatomy of PKI comprises of the following components.
  - Public Key Certificate, commonly referred to as 'digital certificate'.
  - Private Key tokens.
  - Certification Authority.
  - Registration Authority.
  - Certificate Management System.

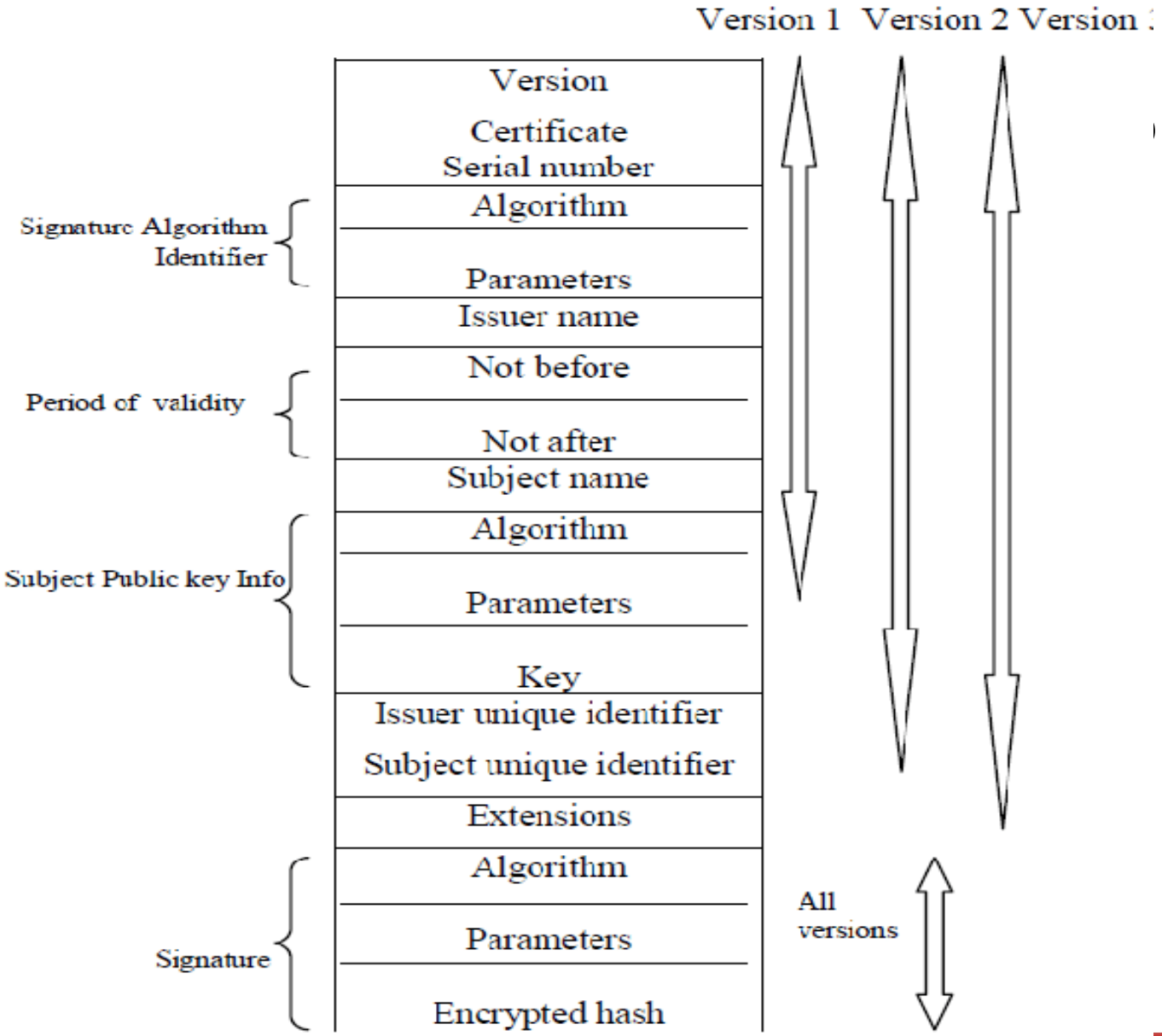
X.509 certificates

Digital Certificate

- A certificate can be considered as the ID card issued to the person. People use ID cards such as a driver's license, passport to prove their identity. A digital certificate does the same basic thing in the electronic world, but with one difference.
- Digital Certificates are not only issued to people but they can be issued to computers, software packages or anything else that need to prove the identity in the electronic world.
  - Digital certificates are based on the ITU standard X.509 which defines a standard certificate format for public key certificates and certification validation. Hence digital certificates are sometimes also referred to as X.509 certificates.

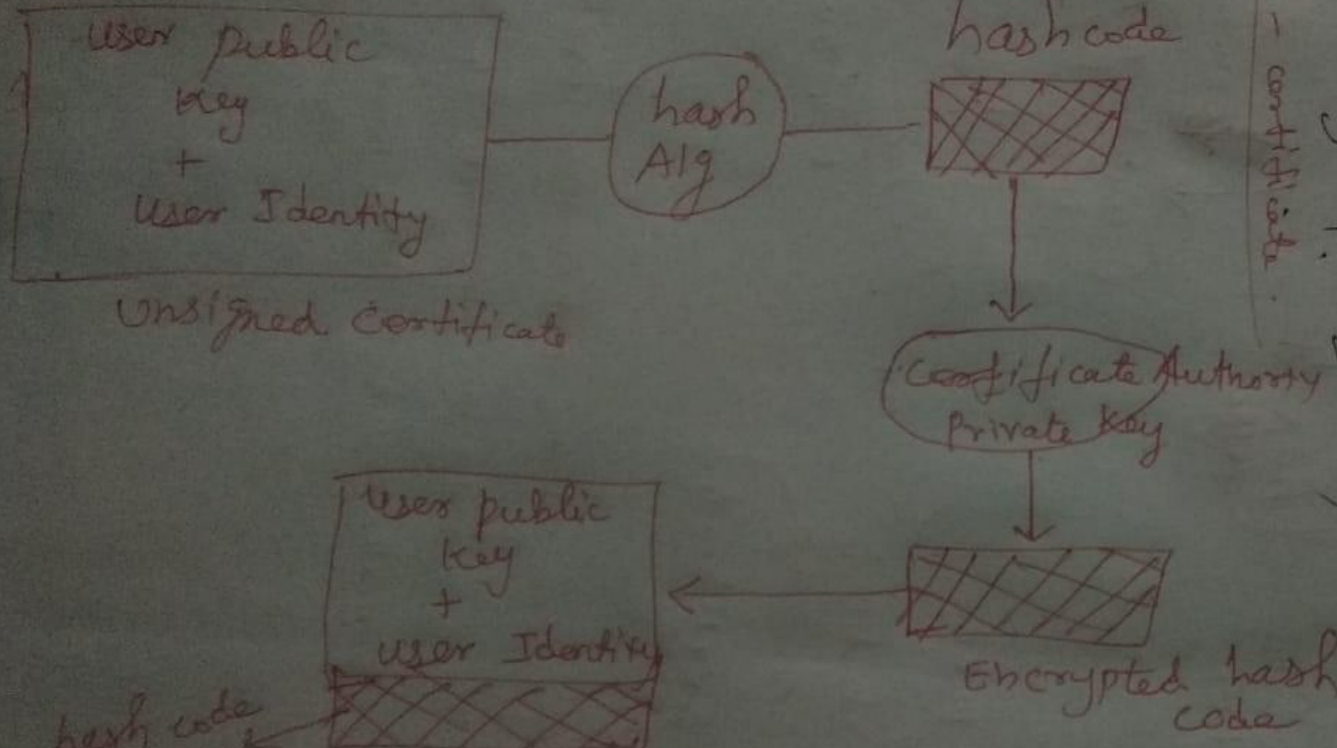
- In cryptography, **X.509** is a standard defining the format of public key certificates.
- X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS.
- An X.509 certificate contains a public key and an identity (a hostname, or an organization, or an individual), and is either signed by a certificate authority or self-signed.
- X.509 is defined by the International Telecommunications Union's Standardization sector (ITU-T), and is based on ASN.1, another ITU-T standard.

The prc
perso
n.

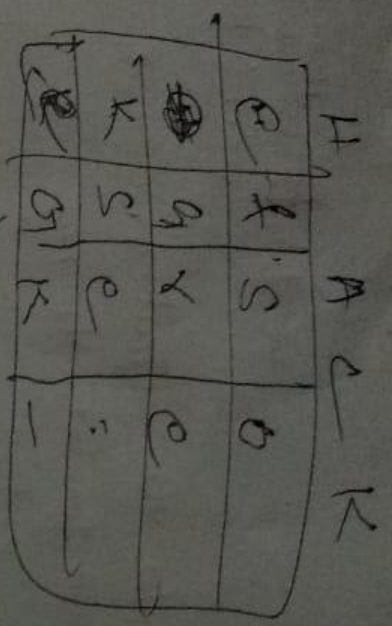


As shown in the illustration, the CA accepts the application from a client to certify his public key. The CA, after duly verifying identity of client, issues a digital certificate to that client.

CA → not responsible to generate key
→ But it provides signature



X.509 - certificate.



Note:
* X.509 - one of the certification directory.
* X.509 is responsible to provide certificates in single place

OpenSSL

- **OpenSSL** is a software library for applications that secure communications over computer networks against eavesdropping or need to identify the party at the other end. It is widely used by Internet servers, including the majority of HTTPS websites.
- OpenSSL contains an open-source implementation of the SSL and TLS protocols.
- The core library, written in the C programming language, implements basic cryptographic functions and provides various utility functions. Wrappers allowing the use of the OpenSSL library in a variety of computer languages are available.
- The OpenSSL Software Foundation (OSF) represents the OpenSSL project in most legal capacities including contributor license agreements, managing donations, and so on
- OpenSSL Software Services (OSS) also represents the OpenSSL project, for Support Contracts.

OpenSSL supports a number of different cryptographic algorithms:

- Ciphers
  - AES, Blowfish, Camellia, Chacha20, Poly1305, SEED, CAST-128, DES, IDEA, RC2, RC4, RC5, Triple DES
- Cryptographic hash functions
  - MD5, MD4, MD2, SHA-1, SHA-2, SHA-3, RIPEMD-160, MDC-2, GOST R 34.11-94, BLAKE2, Whirlpool
- Public-key cryptography
  - RSA, DSA, Diffie–Hellman key exchange, Elliptic curve, X25519, Ed25519, X448, Ed448


Thank You