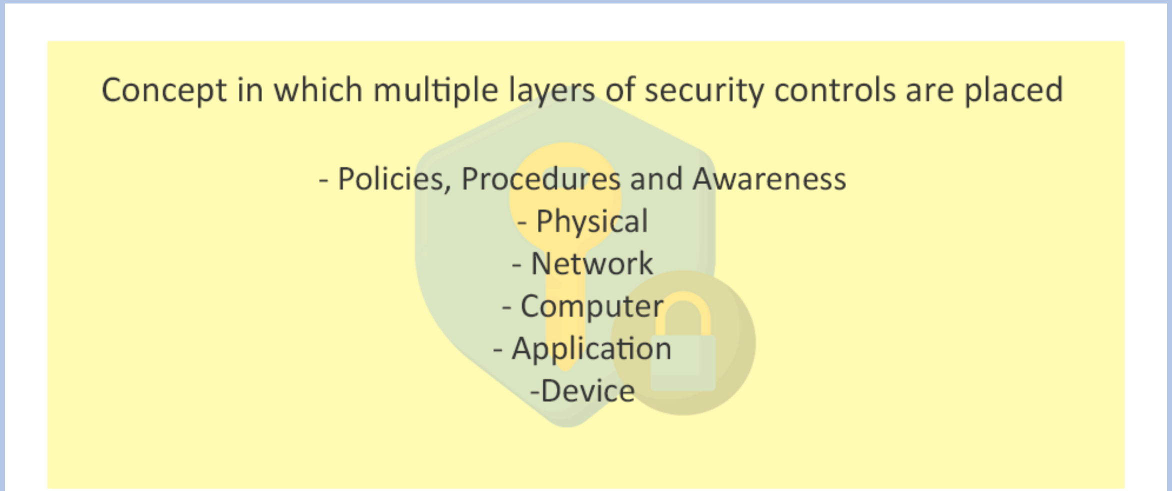
DOMAIN 03:
SECURITY ARCHITECTURE AND ENGINEERING

System Development Life Cycle

- Planning
- Requirements Identification
- Design
- Development
- Testing
- Deployment
- Evaluation
- Lesson Learned

Implementing Secure Design Principle – Defense in Depth



Concept in which multiple layers of security controls are placed

- Policies, Procedures and Awareness
- Physical
- Network
- Computer
- Application
-Device

Defense in Depth (DoD)

- To protect access
- Provide multiple layers of Security Controls
- If one layer breaks, other Security layer Controls are available to protect assets

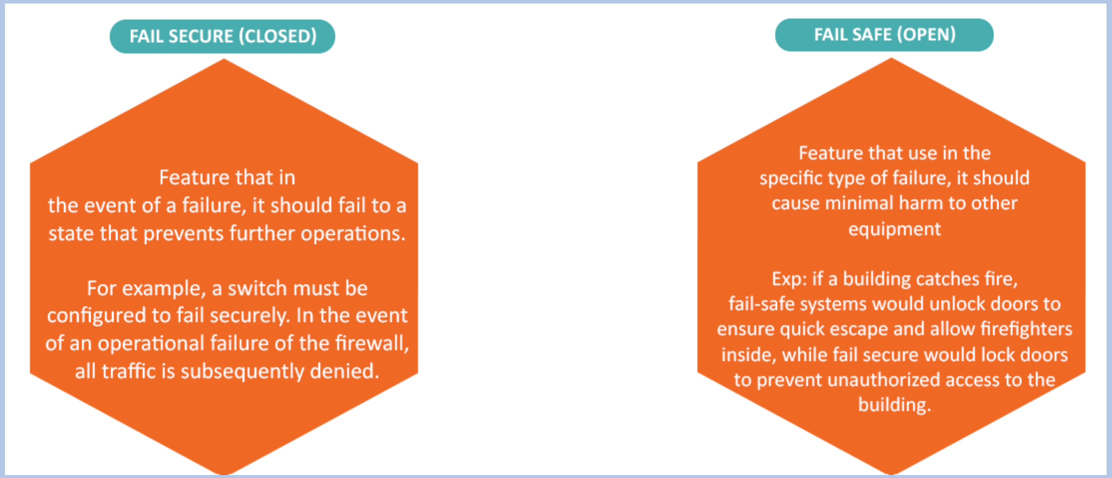
Example: Laptop (provide Controls)

- Create access policy
- Authentication
- Encryption
- End point Security
- patching

Fail Secure & Fail Safe

- **Fail Secure**
  - If Controls fail – environment is still secure
  - Security is priority

  Exp: if firewall fails - all traffic denied



FAIL SECURE (CLOSED)

Feature that in the event of a failure, it should fail to a state that prevents further operations.

For example, a switch must be configured to fail securely. In the event of an operational failure of the firewall, all traffic is subsequently denied.

FAIL SAFE (OPEN)

Feature that use in the specific type of failure, it should cause minimal harm to other equipment

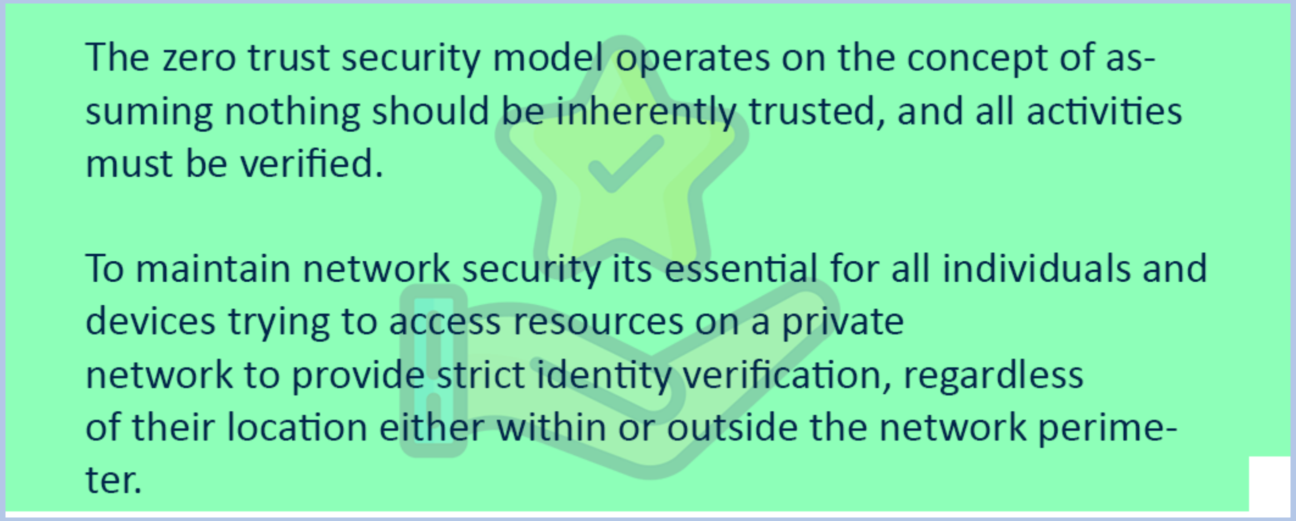
Exp: if a building catches fire, fail-safe systems would unlock doors to ensure quick escape and allow firefighters inside, while fail secure would lock doors to prevent unauthorized access to the building.

- **Fail Safe**
  - If Controls fail – environment is in-secure

  Exp: if the building has fire – fail safe system to unlock doors to escape

- Safety requirements always over write Security requirements

Zero Trust



The zero trust security model operates on the concept of assuming nothing should be inherently trusted, and all activities must be verified.

To maintain network security its essential for all individuals and devices trying to access resources on a private network to provide strict identity verification, regardless of their location either within or outside the network perimeter.

- **Zero Trust Overview**
  - Stated in 2017 by NIST
  - Trust nothing but verifying /validate Controls
  - Firewall is controlling Security in networks
  - Cloud environment is un-trusted
  - DMZ – Organization is creating separate network outside VPN, so if users access Data they cant access organization Data
  - By default communication is blocked till Security validation

Zero Trust

Never Trust always Verify

Every access request be authenticated, authorized, and encrypted prior to the access being granted

Trust but Verify

Once Controls implemented, always verify the Controls
Trust on Controls but keep monitoring

Bell La-Padula Model

Rule 1

No READ Up (due to no Security breach)

Rule 2

No WRITE down (otherwise Top Secret Security leaks the info)

Rule 3

Can access on SAME Level

Purpose:

To maintain **Confidentiality**

Maintain confidentiality

BIBA Integrity Model

- To maintain Integrity

Maintain Integrity

- Use in Government & Public organization

Rule 1

- **No READ down**

Rule 2

- **No WRITE up**

If users (high trusted) read low level Data, this model not allowed because it will be destroy the trust

Can READ Up

How Read Down can corrupt

- I have high integrity info
- If read low integrity
- My integrity has less trust because I am reading low trust info

Clark-Wilson Model

- This model not allowed partial transactions
- If 3 transactions working simultaneously, and posted only 1 transaction and other 2 are down.
- Not allowed partial transactions due to maintain Security

Well formed transaction

Separation of Duties

Open System

- Publically available
- Anyone can review

Easier to integrate with systems from different manufacturers that support the same standards

Close System

- Proprietary system for any particular user
- Not available for public

Same manufacturer, often proprietary and not normally disclosed

Cyber Security Frameworks

- ISO/IEC 27000….
- NIST - Risk Management Framework (RMF)
- Cloud Security Alliance (CSA)

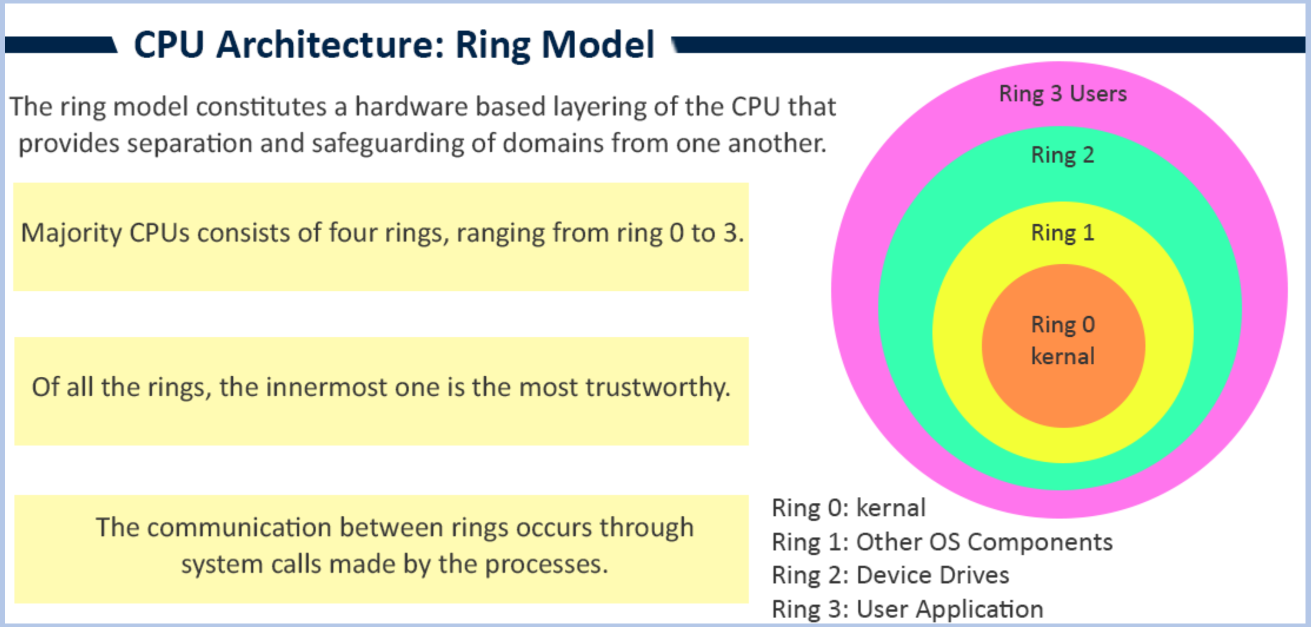
- **Risk Management Framework**
  - ISO 31000 – Can use any Risk Assessment

ISO/IEC 270xx
Risk Management Framework (RMF)
Cloud Security Alliance(CSA) Security, Trust, Assurance, and Risk Registry(STAR)

System Architecture

- **Backed-in** concept
  - When Security Controls identify during requirement gathering stage
  - Pro-active approach for software development
  - Use Security requirements in earlier phases

- **Backed-On** Concept
  - Security Controls identify when needed (not proactive approach)
  - Security Controls apply once software built



CPU Architecture: Ring Model

The ring model constitutes a hardware based layering of the CPU that provides separation and safeguarding of domains from one another.

Majority CPUs consists of four rings, ranging from ring 0 to 3.

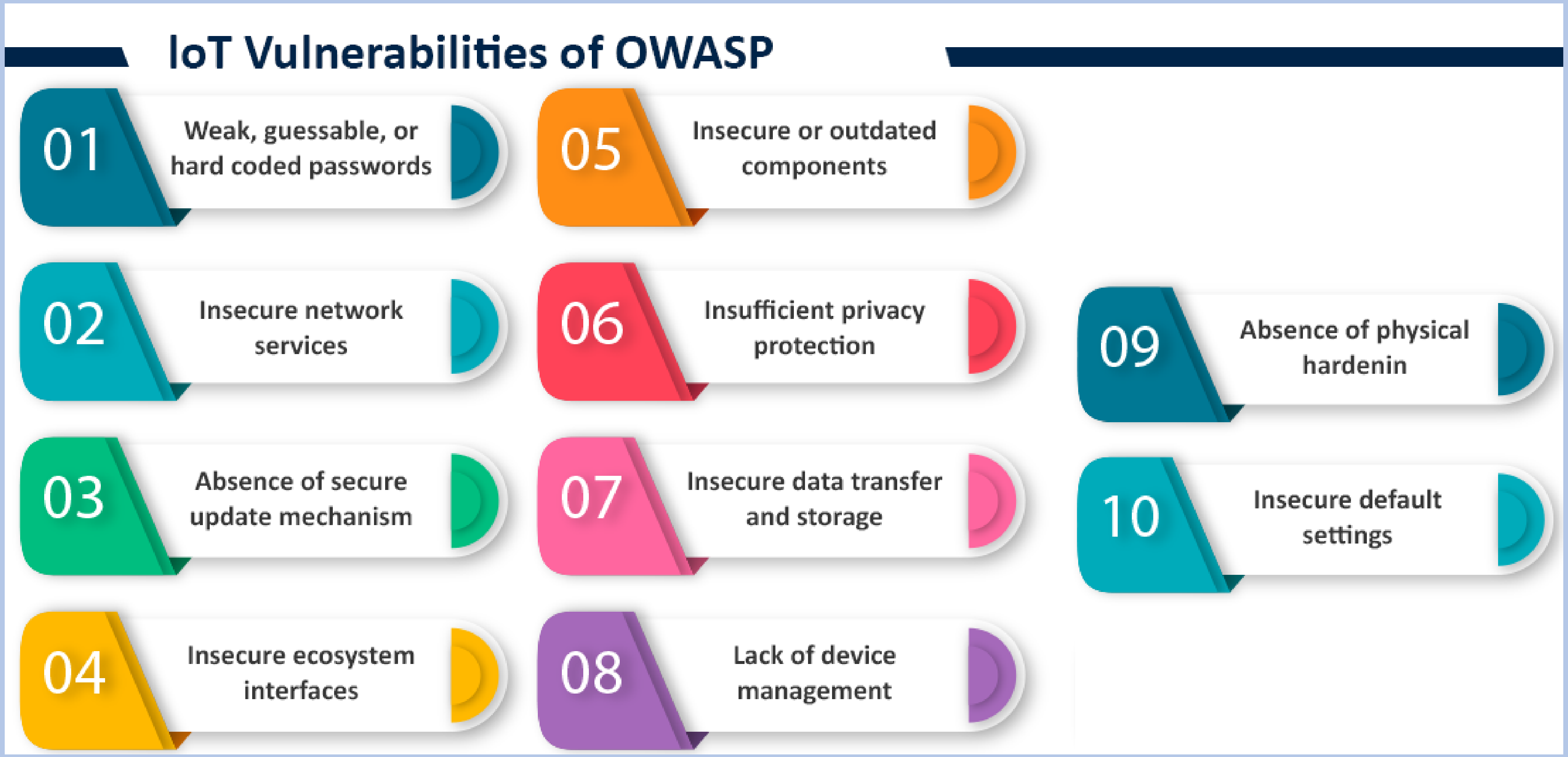
Of all the rings, the innermost one is the most trustworthy.

The communication between rings occurs through system calls made by the processes.

Ring 0: kernal
Ring 1: Other OS Components
Ring 2: Device Drives
Ring 3: User Application

- CPU Architecture – Ring Model
    - Every CPU has 4 Rings
    - Ring 0
        - Kernel
    - Ring 1
        - Other OS setup
    - Ring 2
        - OS setup
    - Ring 3
        - OS setup
        - Admin Mode
            - When CPU running through Ring 0 (Kernel) it processes all instructions (100 instructions)
            - Implementing Access control
        - Normal User Mode
            - When user logging to ring 2 process only 50 instructions
            - Ring 0 – Admin Mode (Full control)
            - Ring 3 – User Mode (limited control)

Vulnerabilities of OWASP



IoT Vulnerabilities of OWASP

01 Weak, guessable, or hard coded passwords

02 Insecure network services

03 Absence of secure update mechanism

04 Insecure ecosystem interfaces

05 Insecure or outdated components

06 Insufficient privacy protection

07 Insecure data transfer and storage

08 Lack of device management

09 Absence of physical hardenin

10 Insecure default settings

Cryptography – Keep Secret Secret

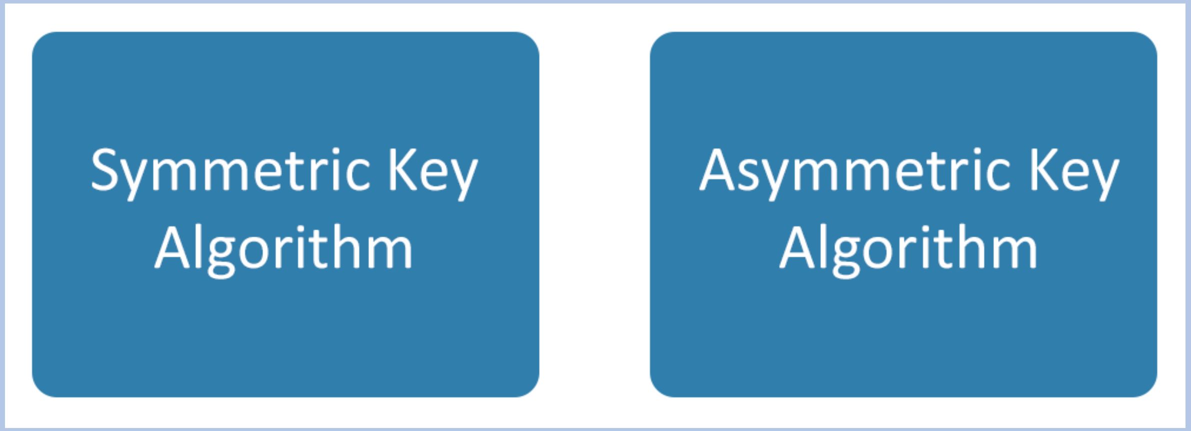
Symmetric

- Stream
- Block
- Algorithm

Symmetric cryptography may be used with temporary keys that exist only for a single session

Asymmetric

- Algorithm
- Hash/HMAC
- Digital Signature



Cryptography

- Use protocols for communication
- Protocols are not secured
- To secure protocols, we have to provide
    - Encryption
    - Authentication
    - Non-Repudiation

- To secure communication, use
    - Encryption
    - Data Integration
    - Authentication
    - Non Repudiation

- Repudiation – Denying something

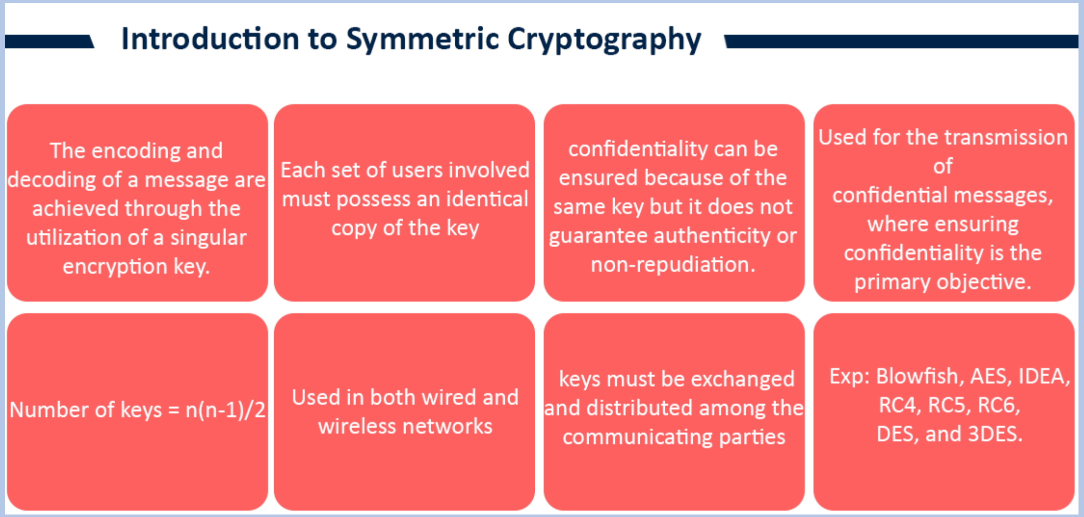
- Hash/HMAC – Data Integrity
- Encryption – Confidentiality
- Digital Signature – Authentication or Non Repudiation

Plain Text – Unencrypted message

Cipher Text – Encrypted message

Encryption – Convert plain text into Cipher

Decryption – Convert Cipher text to Plain



Introduction to Symmetric Cryptography

The encoding and decoding of a message are achieved through the utilization of a singular encryption key.

Each set of users involved must possess an identical copy of the key

confidentiality can be ensured because of the same key but it does not guarantee authenticity or non-repudiation.

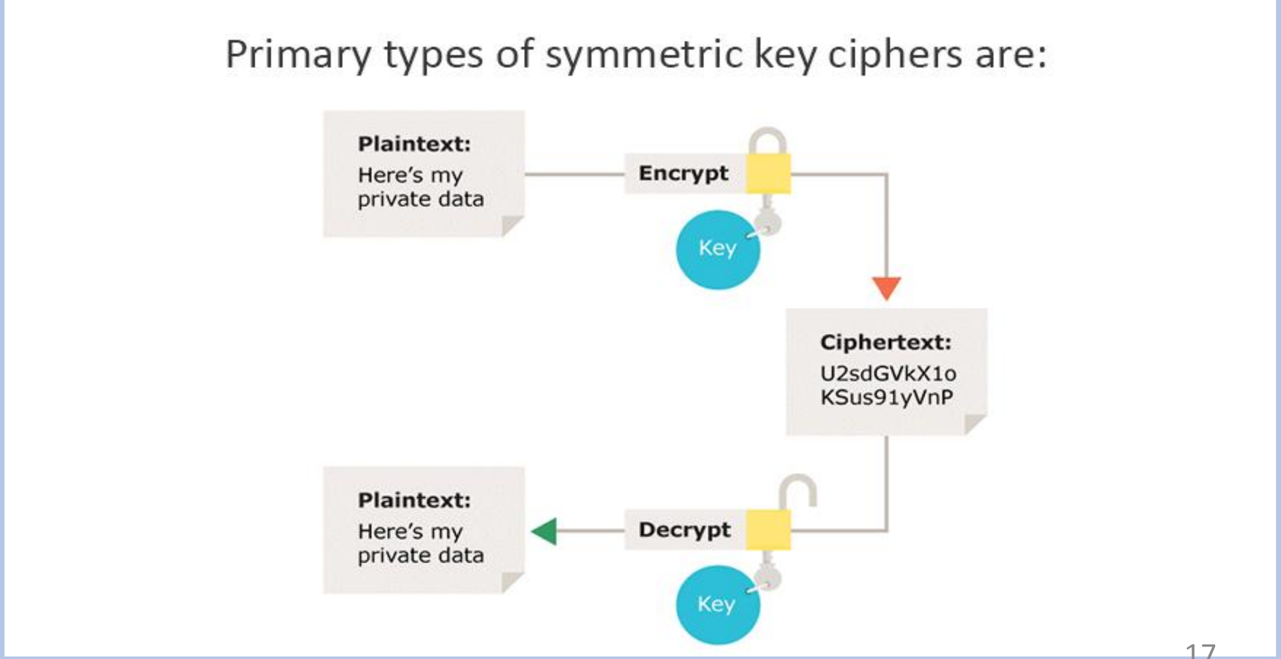
Used for the transmission of confidential messages, where ensuring confidentiality is the primary objective.

Number of keys = n(n-1)/2

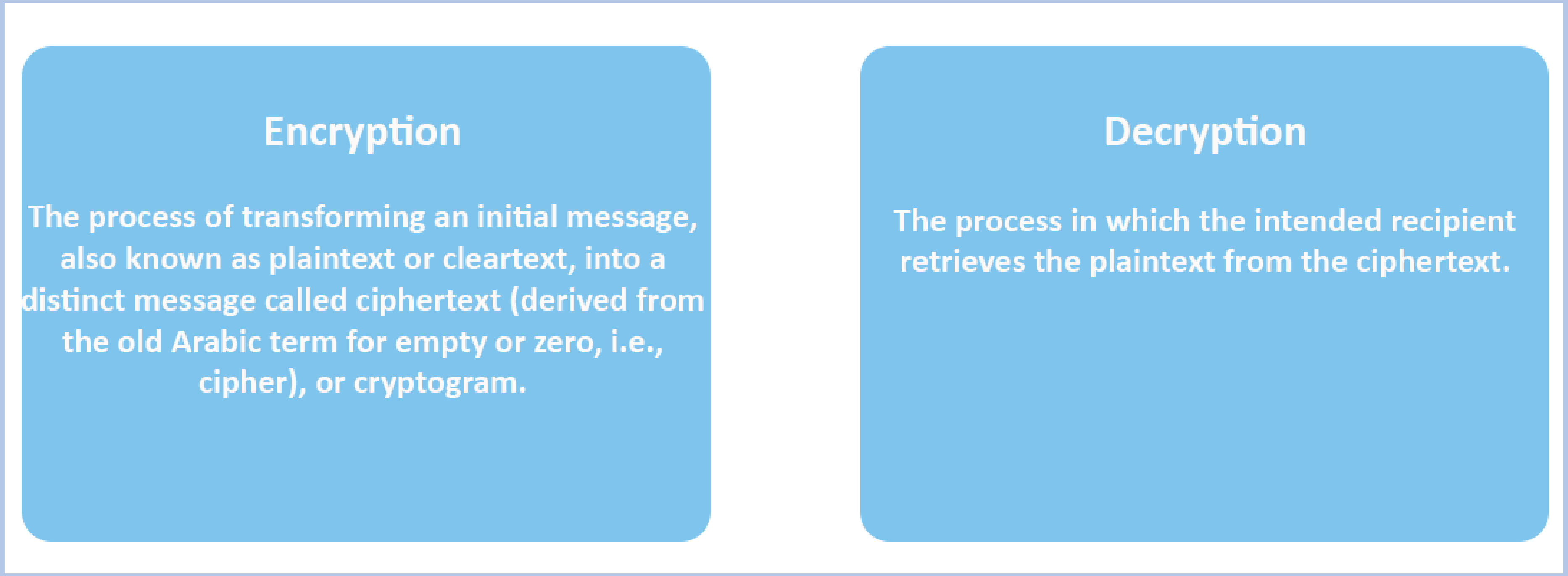
Used in both wired and wireless networks

keys must be exchanged and distributed among the communicating parties

Exp: Blowfish, AES, IDEA, RC4, RC5, RC6, DES, and 3DES.



Primary types of symmetric key ciphers are:



Encryption

The process of transforming an initial message, also known as plaintext or cleartext, into a distinct message called ciphertext (derived from the old Arabic term for empty or zero, i.e., cipher), or cryptogram.

Decryption

The process in which the intended recipient retrieves the plaintext from the ciphertext.

Asymmetric Encryption

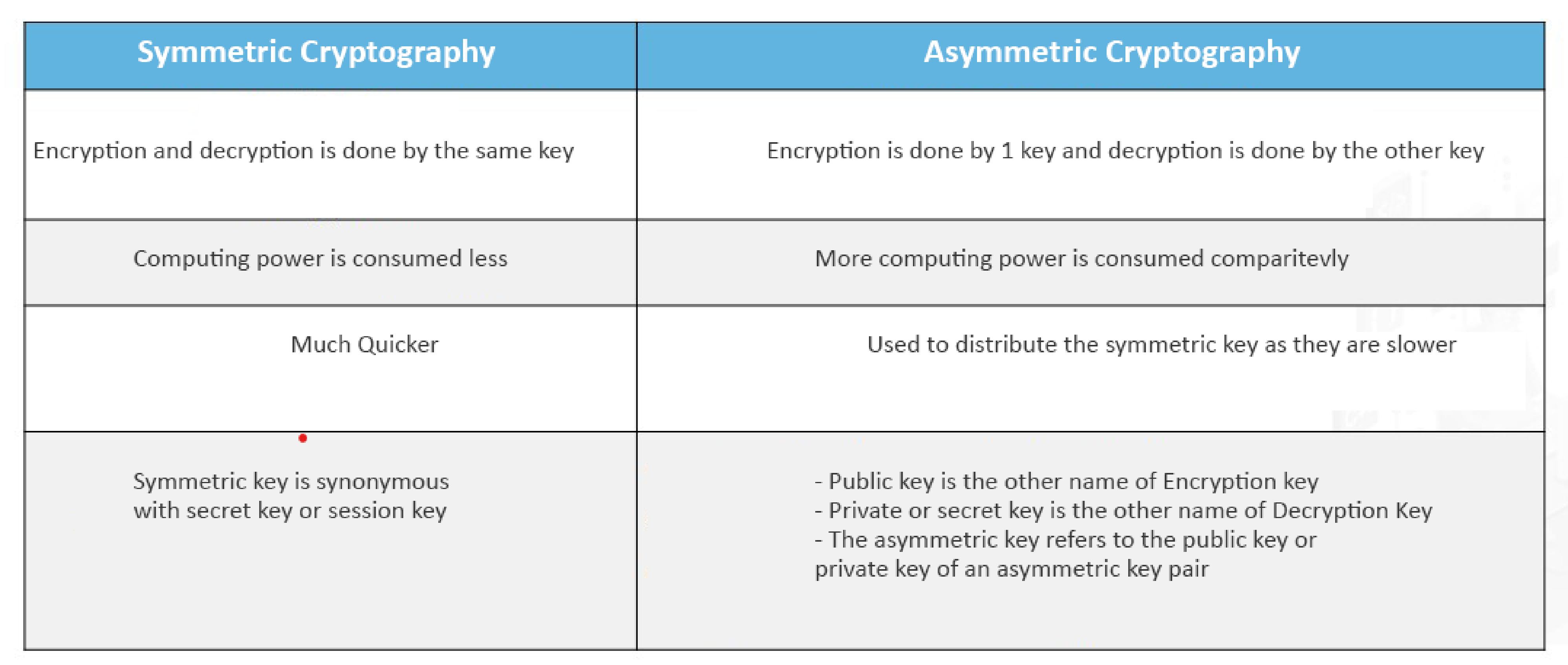
- 2 keys used (public/private)

- Encrypt (public) – Decrypt (private) → Key sharing
- Encrypt (private) – Decrypt (public) → Digital Signature (Authentication/Non Repudiation)

- DH (Diffie Hellman)
  - Use for key sharing
  - Only use for digital signature

- A ------------------------------ B
- Public key (Z)   Public key (Z)
- Private key (X * Y)
- X * Y = Z

- Public key derived from private key (X * Y)



Symmetric Cryptography	Asymmetric Cryptography
Encryption and decryption is done by the same key	Encryption is done by 1 key and decryption is done by the other key
Computing power is consumed less	More computing power is consumed comparitevly
Much Quicker	Used to distribute the symmetric key as they are slower
Symmetric key is synonymous with secret key or session key	- Public key is the other name of Encryption key - Private or secret key is the other name of Decryption Key - The asymmetric key refers to the public key or private key of an asymmetric key pair

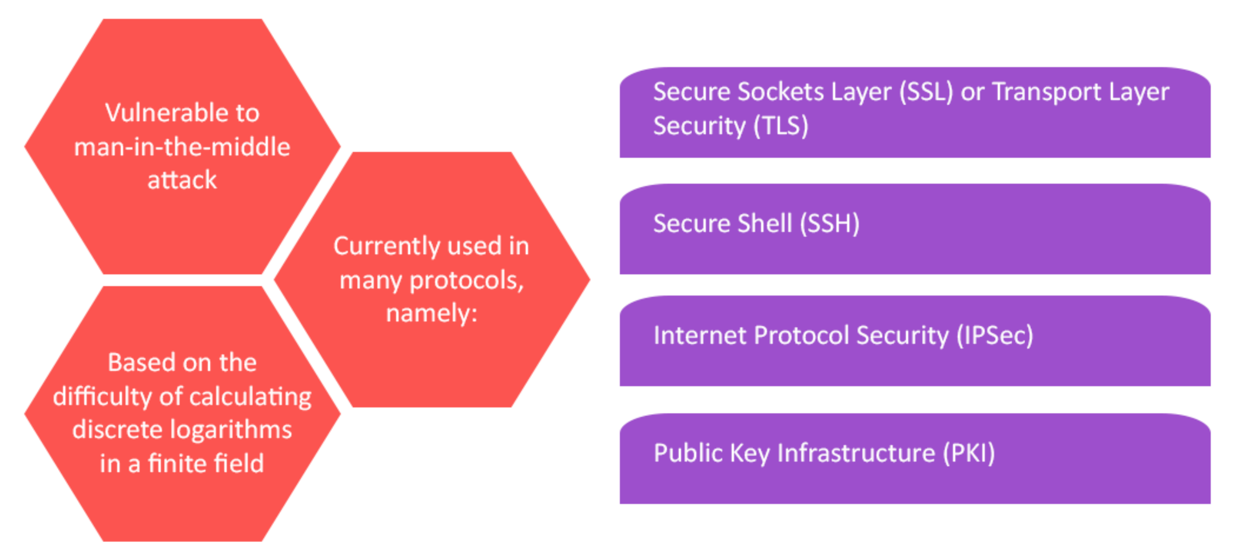
Diffie Helman Key Sharing Mechanism

- A wants to connect to B

- Public key of A sent to B
- B will sent their public key to A

- Public key of A & private key of B – can drive the master key for encryption/decryption

- Vulnerability – Man in the Middle attack



Vulnerable to man-in-the-middle attack

Currently used in many protocols, namely:

Based on the difficulty of calculating discrete logarithms in a finite field

Secure Sockets Layer (SSL) or Transport Layer Security (TLS)

Secure Shell (SSH)

Internet Protocol Security (IPSec)

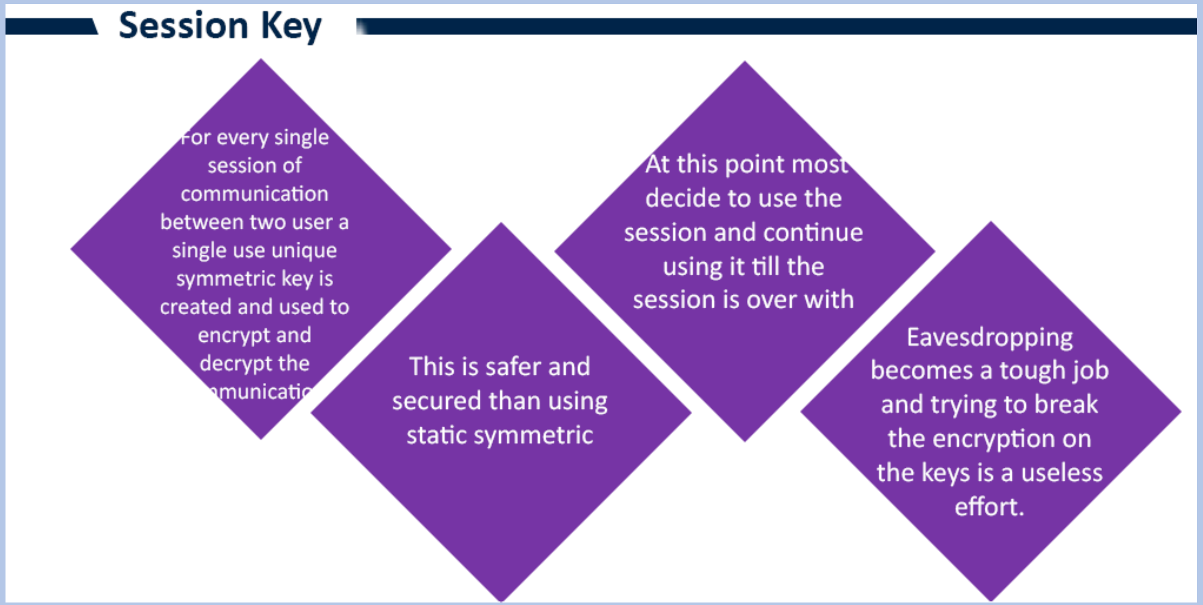
Public Key Infrastructure (PKI)

Session Key

- Symmetric key – only use for session
- Once browser open – generate session key through Master key
- When session over – Session keys destroyed
- When browser closed, session & master key destroyed

- In the browser – if second Tab open – Master key generates another session key

Symmetric cryptography may be used with temporary keys that exist only for a single session



Session Key

For every single session of communication between two user a single use unique symmetric key is created and used to encrypt and decrypt the mmunicatio

This is safer and secured than using static symmetric

At this point most decide to use the session and continue using it till the session is over with

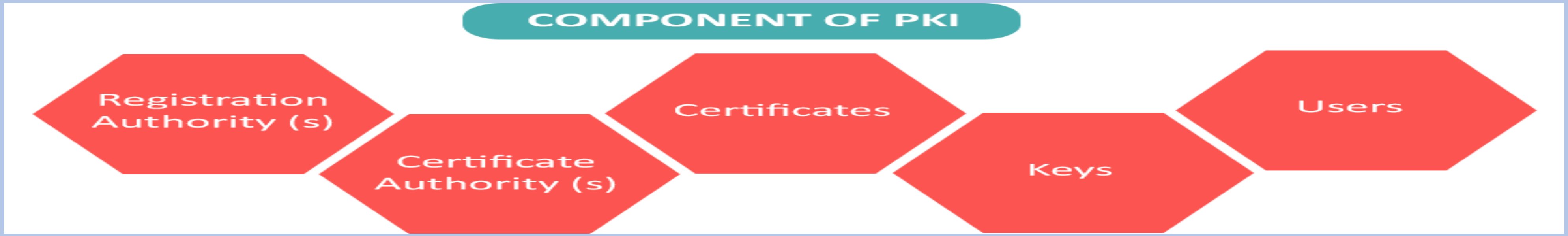
Eavesdropping becomes a tough job and trying to break the encryption on the keys is a useless effort.

Public Key Infrastructure (PKI)

- Why digital signature required
- If user connects to browser & browser send their public key to user - how user knows he is communicating to correct browser (might be some hackers sent info)

Use Symmetric, Asymmetric Encryption & Hashing to provide Digital Certificate

- Web Admin – send public key to PKI – they verify if public key is correct – then they create Digital Certificate
- Send digital certificate to web admin
- Web admin load digital certificate to browser
- PKI to verify
- Web name
- Validity/time period
- RA (Registration Authority)
  - Verify background of the webSite
- CA (Certification Authority)
  - Issue digital certificate



Certificate & Regulatory Authority

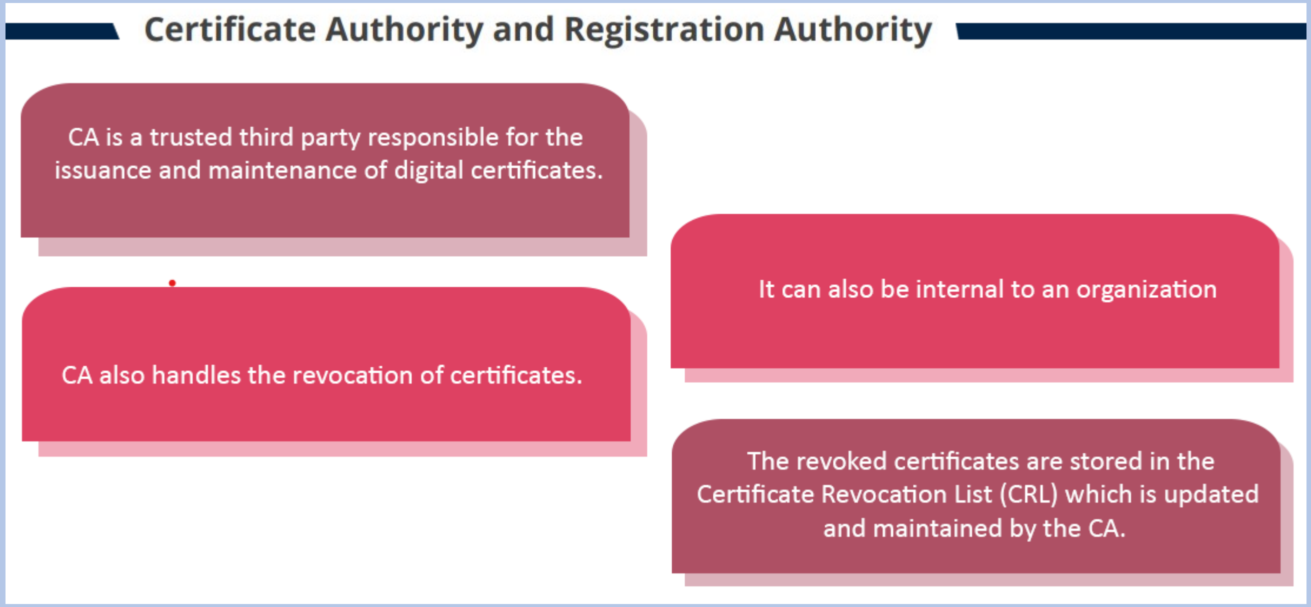
- OCSP – protocol to communicate

- Certification Revocation List (CRL) Criteria
    - Time for CA to put certificate in Revoke List if
        - When certificate time expired
        - If found error
        - If Private key compromised

        - Active Directory & IIS can both create Digital Certificates

        - When setup email account for new employee, always setup digital signature for new employee

  - Enrollment Process
    - Public key submit to CA for validation



Certificate Authority and Registration Authority

CA is a trusted third party responsible for the issuance and maintenance of digital certificates.

CA also handles the revocation of certificates.

It can also be internal to an organization

The revoked certificates are stored in the Certificate Revocation List (CRL) which is updated and maintained by the CA.

HASH Function (Integrity)

Variable length Plain Text hashed into Fixed length hash or Message Digest (MD)

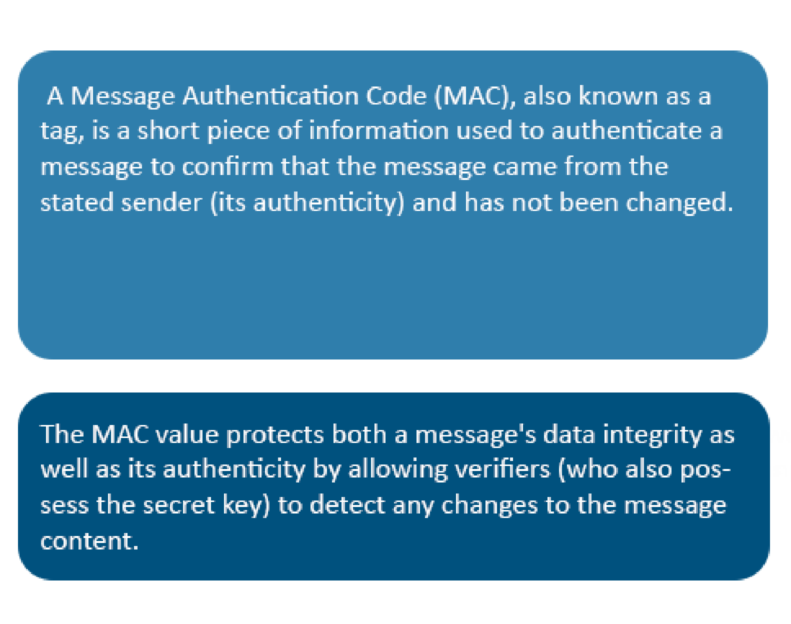
MD5 – 128 bit fixed length hash

SHA1 (secure Hash Logarithm) – 160 bit

- In Hash output always remain fixed
- Hash values never be same for different inputs
- Hash value not reversible (One way active)
- From output – Input cannot be retrieved

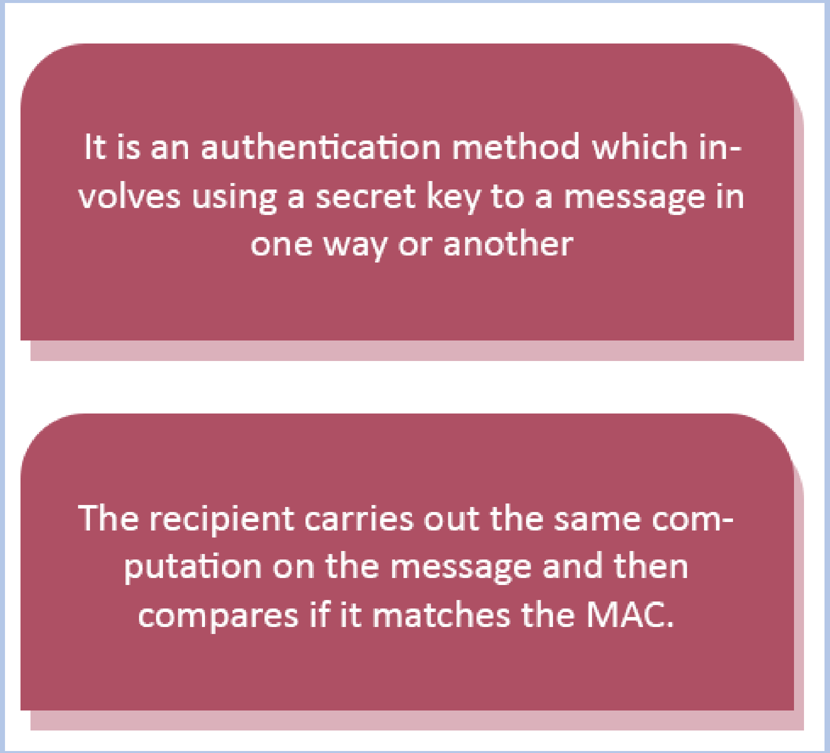
- **Collision**
    - When two inputs provide same output
    - Always avoid collision
    - More output bits → less chances of collision
    - Can reduce collision by using large output value
    - Good design

Message Authentication Code (MAC)



A Message Authentication Code (MAC), also known as a tag, is a short piece of information used to authenticate a message to confirm that the message came from the stated sender (its authenticity) and has not been changed.

The MAC value protects both a message's data integrity as well as its authenticity by allowing verifiers (who also possess the secret key) to detect any changes to the message content.



It is an authentication method which involves using a secret key to a message in one way or another

The recipient carries out the same computation on the message and then compares if it matches the MAC.

How Hash function verify Data integrity

- User A creates message
- Hash the message
- Encrypt the message called Hash Mac (HMAC)

- User A send message and HMAC to user B
- B has to verify HMAC

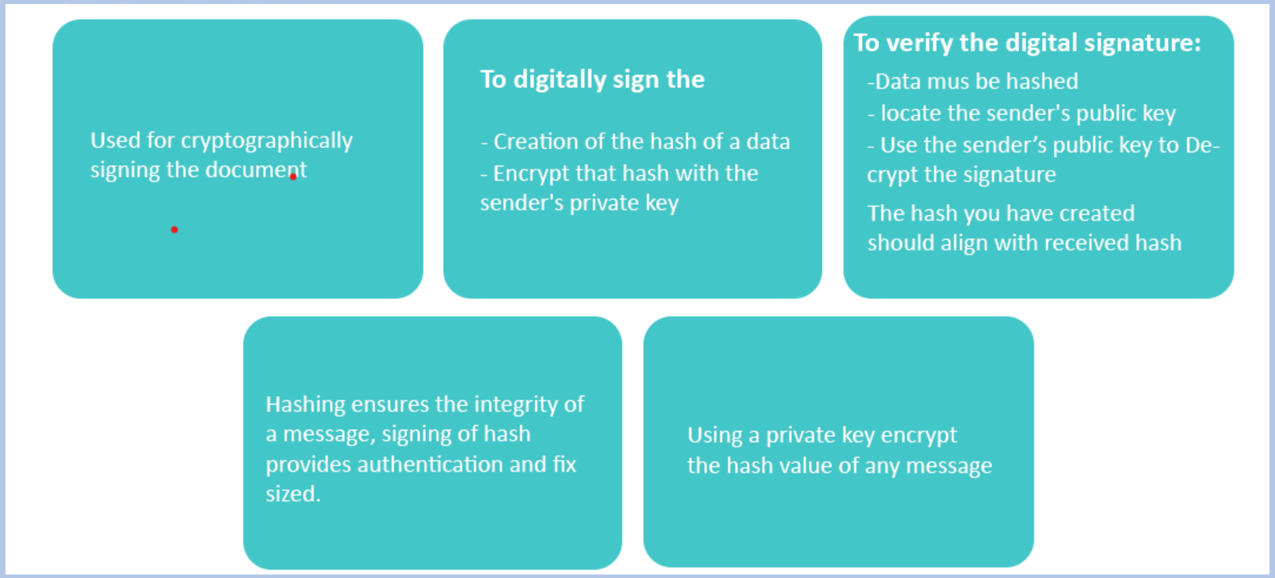
- B receives the message & HMAC
- B Hash the message
- Encrypt message by using same key called HMAC

- If both HMAC same – message is same, not modified

Digital signature (DS)

Integrity

Non-repudation



Used for cryptographically signing the document

To digitally sign the

- Creation of the hash of a data
- Encrypt that hash with the sender's private key

To verify the digital signature:

-Data mus be hashed

- locate the sender's public key

- Use the sender's public key to De-crypt the signature

The hash you have created should align with received hash

Hashing ensures the integrity of a message, signing of hash provides authentication and fix sized.

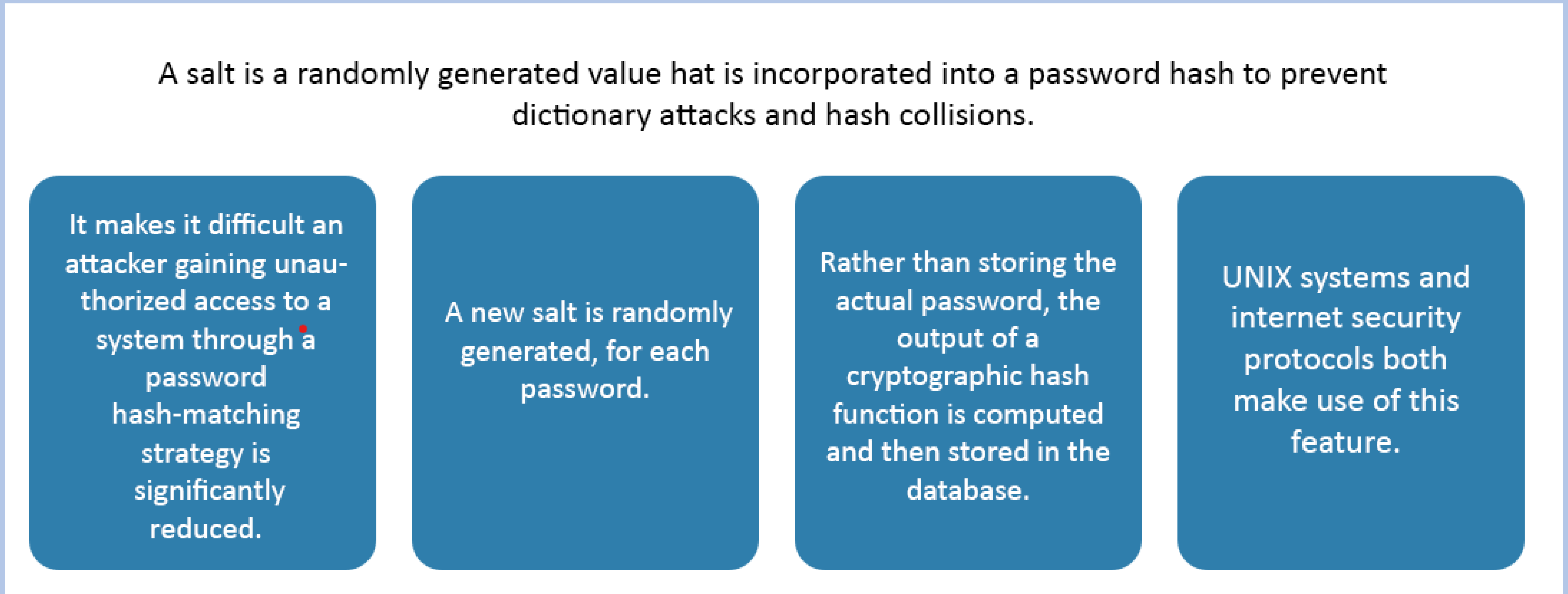
Using a private key encrypt the hash value of any message

- **Digital Certificate (DC)**
  - Contain digital signature
  - Authentication
  - Non-repudation

Salting

- Attackers to crack password
  - Brute Force
  - Birthday attack
  - Rainbow Table

To defend against dictionary attack



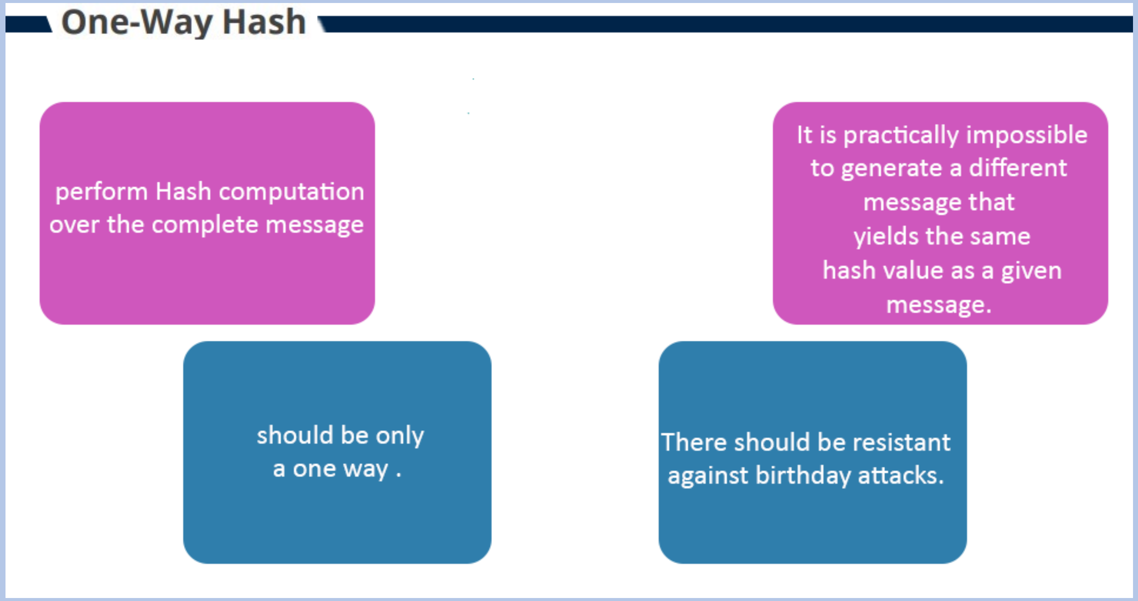
A salt is a randomly generated value hat is incorporated into a password hash to prevent dictionary attacks and hash collisions.

It makes it difficult an attacker gaining unauthorized access to a system through a password hash-matching strategy is significantly reduced.

A new salt is randomly generated, for each password.

Rather than storing the actual password, the output of a cryptographic hash function is computed and then stored in the database.

UNIX systems and internet security protocols both make use of this feature.



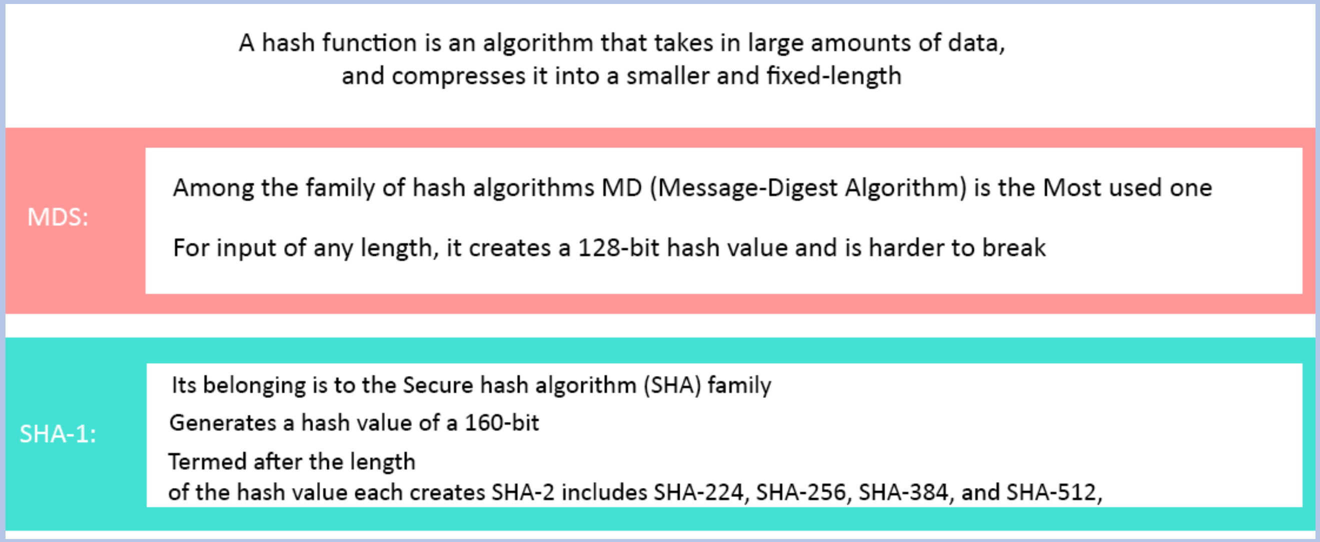
One-Way Hash

perform Hash computation over the complete message

It is practically impossible to generate a different message that yields the same hash value as a given message.

should be only a one way .

There should be resistant against birthday attacks.



A hash function is an algorithm that takes in large amounts of data, and compresses it into a smaller and fixed-length

MDS:

Among the family of hash algorithms MD (Message-Digest Algorithm) is the Most used one

For input of any length, it creates a 128-bit hash value and is harder to break

SHA-1:

Its belonging is to the Secure hash algorithm (SHA) family

Generates a hash value of a 160-bit

Termed after the length
of the hash value each creates SHA-2 includes SHA-224, SHA-256, SHA-384, and SHA-512,