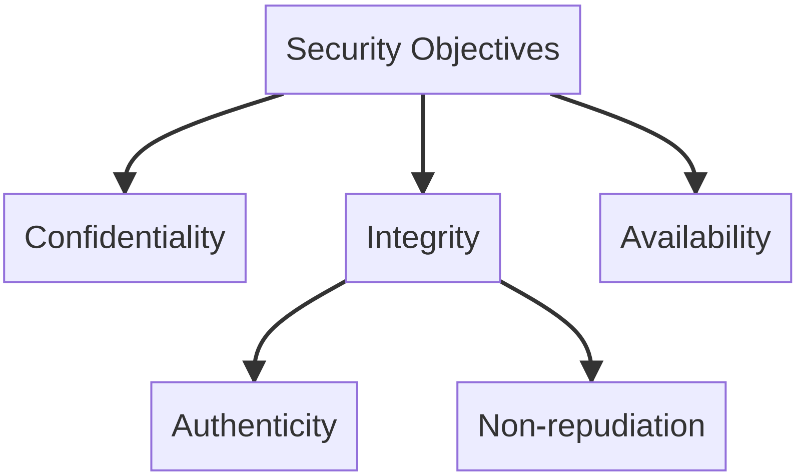
Study Notes: IT-Security

1. Basics

1.1 Security Objectives



Confidentiality:

- Confidentiality means protecting information from access by unauthorized persons
- **Measures:** Access Control, Physical Protection, Encryption

Integrity:

- Integrity means that data is complete and correct when an authorized persons created, transfered and stored. Intentional or unintentional changes can cause a violation of integrity.
- **Measures:** Hashing, Digital Signatures, Store copies to compare them later, Access rules

Availability:

- Availability of an IT-System means that the system is available and accessible to authorized person
- **Measures:** Backups, Redudant design

2. Cryptology

2.1 Kerckhoffs's principles

- If a system is not provably secure, it should be practically secure
- The design of a system should not require secrecy and should not be a problem if it falls in the hands of the enemy
- A cryptosystem must be easy to use

2.2 Perfect Security

- Crypto method is perfectly secure if it's secure against attackers with umlimited resources (time, computing power)
- **Example:** One-Time Pad

2.3 Practical Secure

- Crypto method is practically secure if it's resistant against attacker with limited resources (limited time and computing power)

2.4 Security Level

- Crypto method has a security level of n bits if an attacker needs 2^n attempts to break the method
- Today should have security level of $\geq$ 100 bits $\rightarrow$ 2^{100} attempts
- Security Level ≥ 100 100 bits means at least 2^{100} keys (2^{101} assuming one key is the correct one)

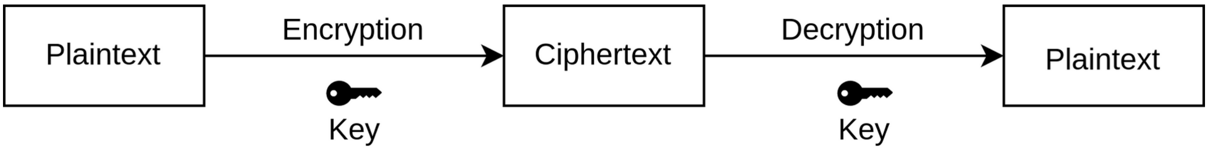
2.5 Types of Cryptography

- Symmetric
- Hash Functions
- Asymmetric

2.6 Symmetric Cryptography

2.6.1 Basics

- Sender and receiver use same key for encryption and decryption



- $enc(k, m) = c$
- $dec(k, c) = m$
- $dec(k, enc(k, m)) = m$

2.6.2 Pros

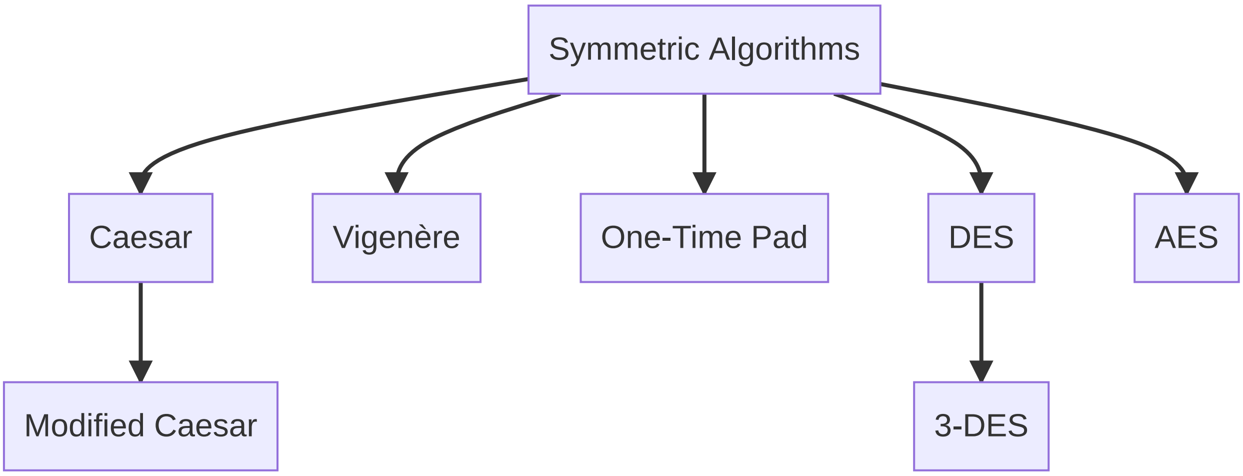
- Faster than asymmetric cryptosystems

- Can be implemted by hardware

2.6.3 Cons

- The sender and recipient must know the (same) key
- The number of keys to be managed quickly becomes very large

2.6.4 List of Symmetric Algorithms



- Modern algorithms: AES (Advanced Encryption Standard), 3-DES (Triple Data Encryption Standard)

2.6.5 One-Time Pad

- Message and key Converted to bits
- Message and key have same length
- Encryption by XOR message and key
  - $\mathrm{XOR}(k, m) = c$
  - $\mathrm{XOR}(k, c) = m$

```
 Plaintext    C        r        y        p        t        o
      Bits 01000011 01110010 01111001 01110000 01110100 01101111
                               XOR
       Key 00110000 00110000 00110000 00110000 00110000 00110000
                                =
Ciphertext 01110011 01000010 01001001 01000000 01000100 01011111
                               XOR
       Key 00110000 00110000 00110000 00110000 00110000 00110000
                                =
      Bits 01000011 01110010 01111001 01110000 01110100 01101111
 Plaintext    C        r        y        p        t        o
```

- For attacker ciphertext is absolutely random if...
  - A random key is used

    - Key is indpendent from plain text
- Perfectly secure as ciphertext does not provide any infrmation about plain text or key

2.6.6 Functioning of Symmetric Algorithms

- Symmetric Algorithms are either work as *stream ciphers* or *block cipher*

Stream Ciphers:

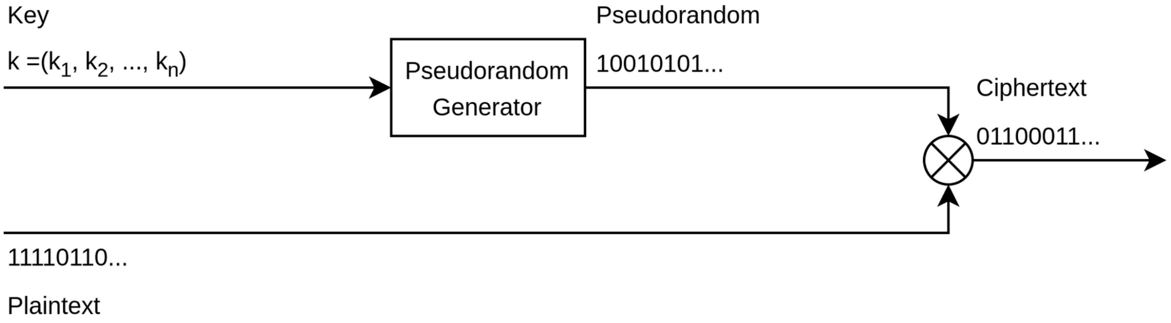
- Converts plaintext into ciphertext bit-by-bit
- **Algorithms working as Block Ciper:** One-Time Pad, A5/1

Block Ciphers:

- Convert plaintext into ciphertext in fixed-size blocks (Usually 64-bit or 128-bit)
- **Algorithms working as Block Ciper:** DES, 3-DES, AES, Blowfish

2.6.7 Stream Ciphers

- Reproduction of the One-Time Pad
- Pseudorandom key is generated from the key k $\in$ {0,1}^{n}, n $\geq$ 100 (keystream)
- Keystream linked bit-by-bit with plaintext via XOR



2.6.8 Advantages of Stream Ciphers

- Pseudorandom Generator via LFSR (Linear Feedback Shift Register) can be implemented efficiently in hardware (only needs XOR and shift)
- Encryption and Decryption are very efficient

2.6.9 Block Ciphers

- Maps blocks of fixed length bitstrings to blocks of fixed length bitstrings
- Blocks usually 64-bits or 128-bits
- Keys must have bit length of $\geq$ 100 to be practically secure
- Mostly 128-bit or 256-bit keys are used

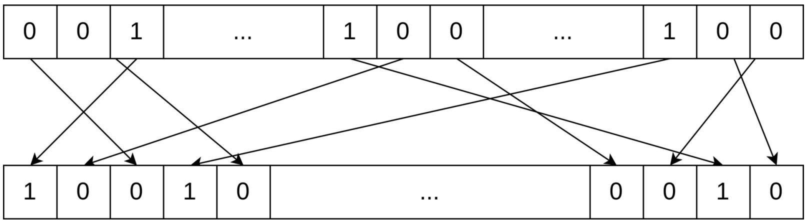
2.6.10 Blocks Cipher Construction

- Block Ciphers can be constructed on 3 ways, using *Diffusion*, *Confusion*, *Round based*

Diffusion:

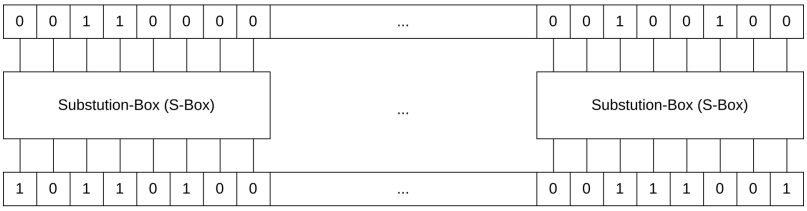
- The order of the bits is changed

- Realized by *permutation*
- Effects only arise after several rounds



Confusion:

- Each bit of the ciphertext depends on as many bits as possible of the key.
- Realized by *substitution*
- mapped $\{0,1\}^n \rightarrow \{0,1\}^n$
- Bit string usually 8- to 16-bit length



Round based:

- Repeated use of diffusion, confusion and key (derivation).

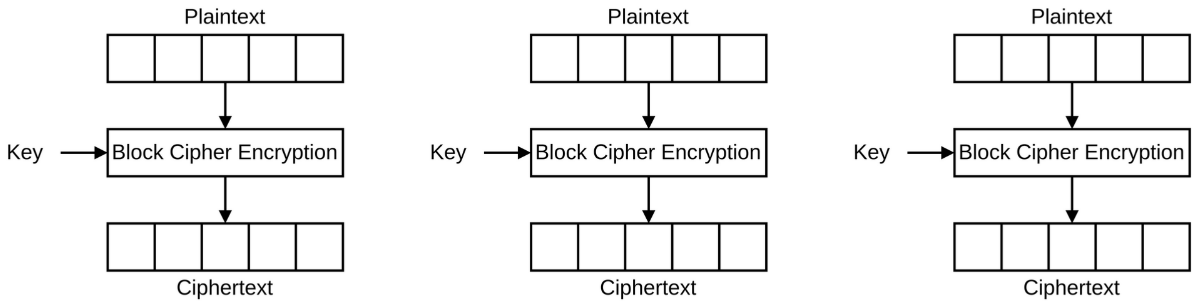
2.6.11 Block Cipher Operating Modes

- To encrypt data which is longer than one block, several operating modes exist
- Describe how to apply the encryption of a single block to the other blocks
- For encryption of longer plaintexts it's broken down into blocks and last block is filled with a PADDING
- The operating modes ECB, CBC und CTR existieren

```
001010....... 101001....... ............. 101111PADDING
\___________/ \___________/               \___________/
   Block 1       Block 2                     Block 8
   128 bit       128 bit                     128 bit
```

Electronic Code Book (ECB):

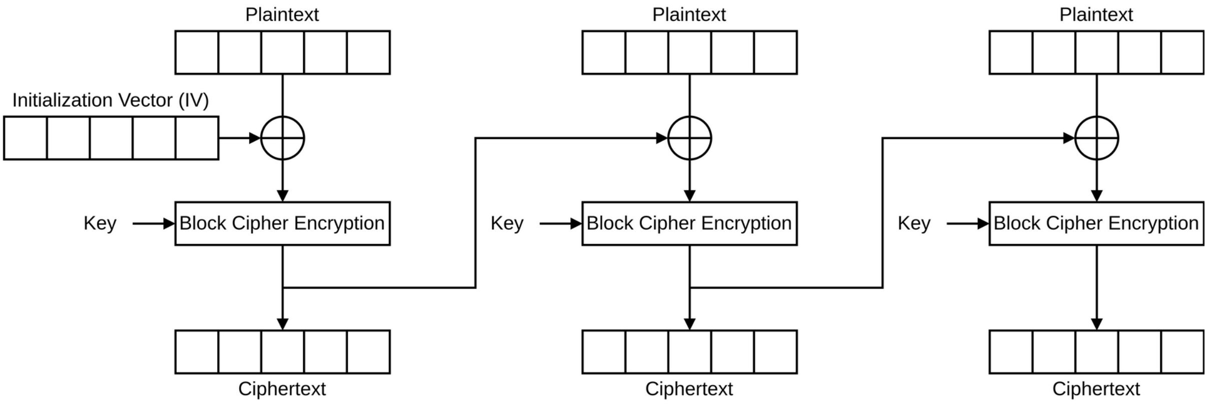
- Plaintext is broken down into blocks an then encrypted block by block
- Problem: Same plaintext block is mapped to the same ciphertext block which can be used to recognize a pattern in the ciphertext. → ciphertext should not depend on key and plaintext but also on other parameter



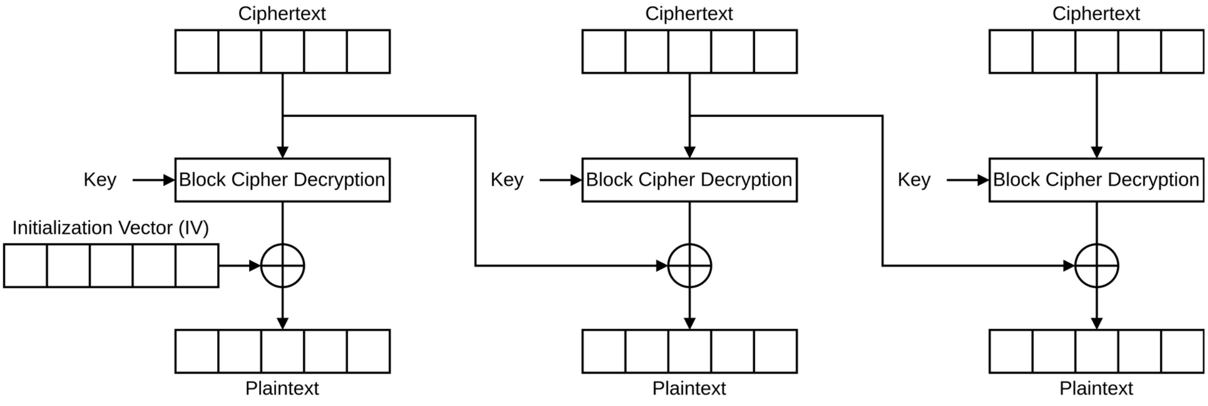
Cipher Block Chaining (CBC):

- An random initilization vector is linked with the first block of plaintext and encrypted. The generated ciphertext is then linked via XOR with the next ciphertext and encrypted and so on

For Encryption:



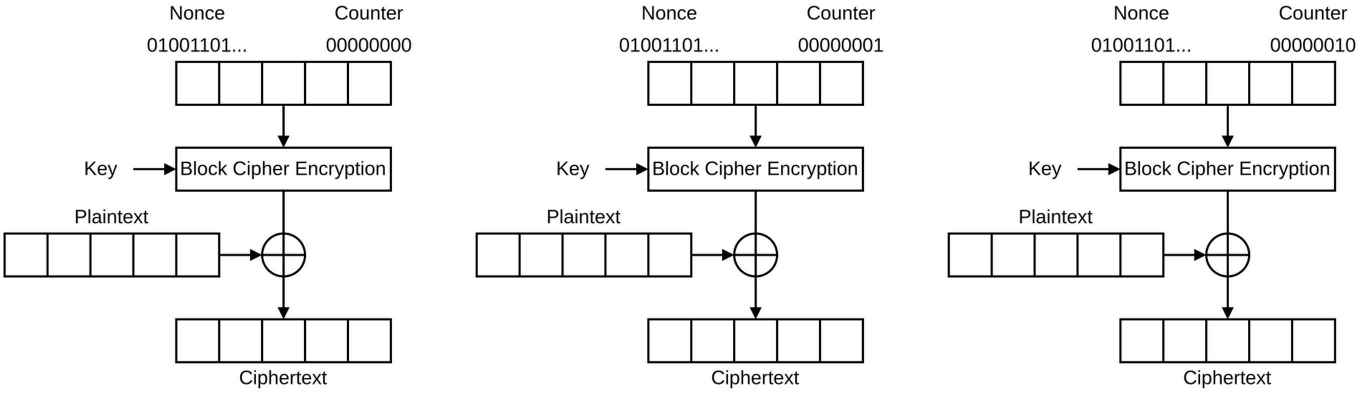
For Decryption:



Counter Mode (CTR):

- Initialization vector consists of *Nonce* and *Counter*. For Each block the Counter increments of 1.

For Encryption:



2.7 Hash Functions

2.7.1 Basics

- Map infinite length of bit string to fixed length of bit string
- $H : \{0,1\}^* \to \{0,1\}^n$

2.7.2 Properties of Hash Functions

- Input size of message possibly inifinite (possible)
- Hash value always same fixed output size
- Colission resistant
- Preimage resitant
- Efficently calculated

2.7.3 Usage

- Digital signatures (and Certificate)
- **Integrity protection:** Check wheter a message has been changed. Storage of message and hash on different systems

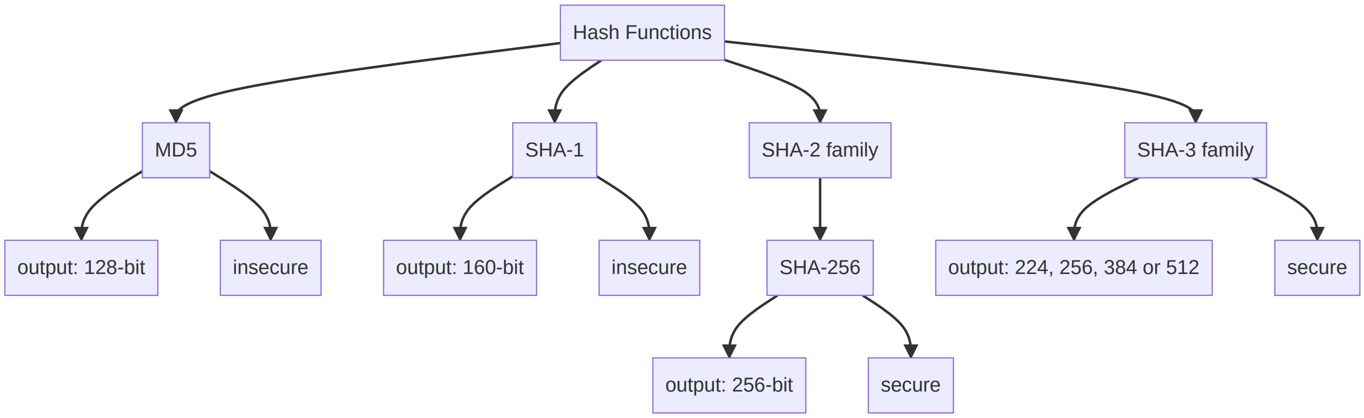
2.7.4 Colission Resistance

- Hash functions should be colission resistant
- Should not be two different messages that generate the same hash value
- In practice collisions cannot be avoided, as for every hash function we map an infinite set of possible messages to a finite set of possible hash values
- Should be practically impossible

2.7.5 Preimage Resistance

- Hash functions should be preimage resistant
- Should be no clues, such as a pattern in the hash value, that can be used to recreate the original message

2.7.6 Cryptographic Hash Functions



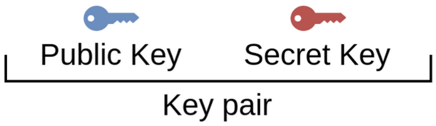
2.7.7 Salts

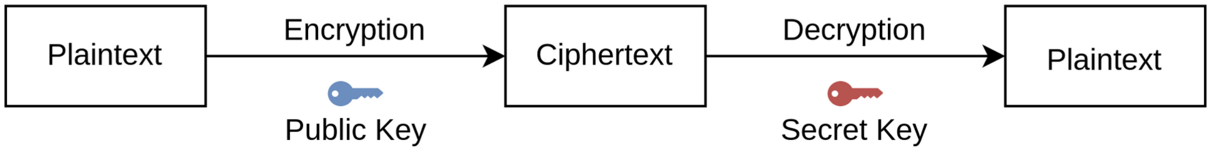
- Problem: The same input always results in the same hash value
- To avoid this problem a different, random *salt* is added/concatenated to the input before hashing, resulting in different hashes
- Salt is stored together with the password

2.8 Asymmetric Cryptosystems

2.8.1 Basics

- Both communication paterns have a keypair generated
  - A has a key pair (pk_A, sk_A)
  - A has a key pair (pk_B, sk_B)
- Keypair consists of a public key (pk) and a private key/secret key (sk)
- Instead of using the same key for encrypting we use public key for encryption and secret key for decryption





Public Key:

- Used for encrypting
- Key can be shared with everyone

- Anyone can encrypt a message with your public key

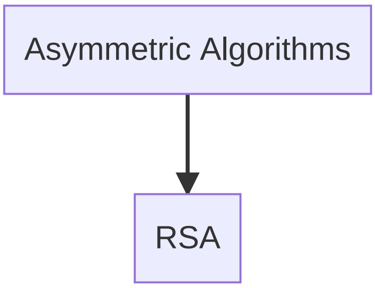
Secret Key:

- Used for decrypting
- Should not be shared with anyone
- You use the key for decrypting message encrypted with your public key by others

2.8.2 Pros

- Solves key exchange problem

2.8.3 List of Asymmetric Algorithms



2.9 Symmetric Cryptography vs Asymmetric Cryptography

Symmetric:

> ✅ **Pro**
>
> Low complexity, higher efficiency (HW implementation)

> ⚠️ **Con**
>
> Complex key distribution

> ⚠️ **Con**
>
> No meaningful and effective realization of digital signatures

Asymmetric:

> ⚠️ **Con**
>
> Higher complexity, lower efficiency (despite HW implementations)

✅ **Pro**

Simple key distribution

✅ **Pro**

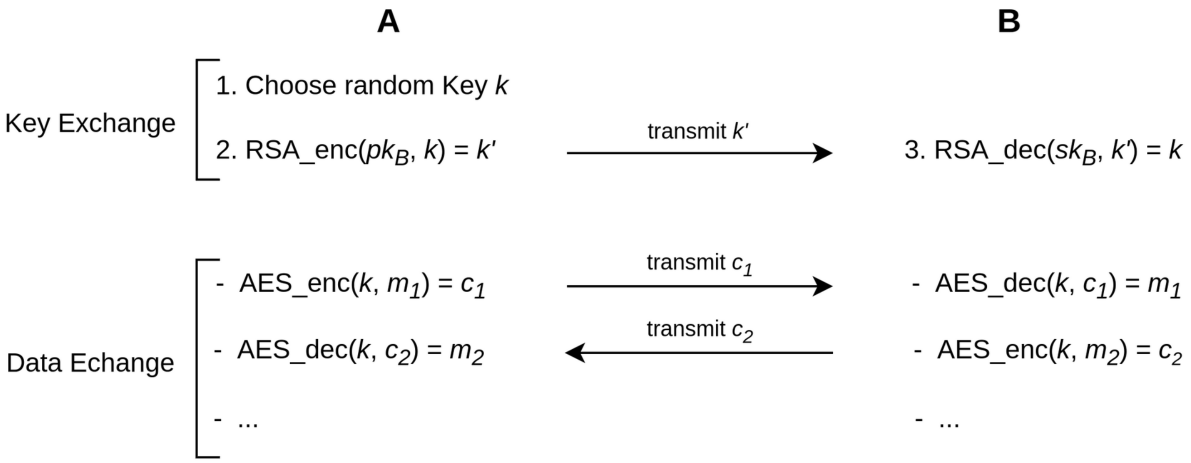
Digital signatures easy to implement

Solution:

- Hybrid Approach
  - Asymmetric cryptography for key exchange
  - Symmetric cryptography for the actual encryption

2.10 Hybrid Cryptosystems

- Combination of Asymmetric cryptography for key exchange and symmetric cryptography for actual encryption



A
B
Key Exchange
1. Choose random Key k
2. RSA_enc(pk_B, k) = k'
transmit k'
3. RSA_dec(sk_B, k') = k
Data Echange
- AES_enc(k, m_1) = c_1
transmit c_1
- AES_dec(k, c_1) = m_1
- AES_dec(k, c_2) = m_2
transmit c_2
- AES_enc(k, m_2) = c_2
- ...
- ...

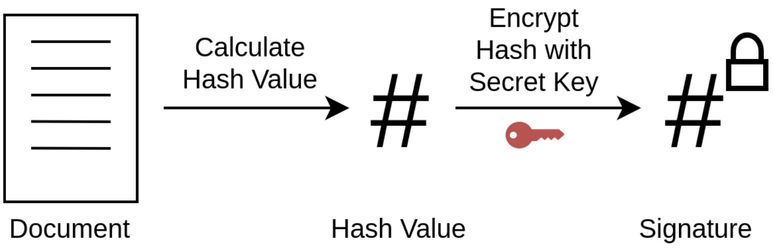
2.11 Digital Signatures

2.11.1 Requirements of Digital Signatures

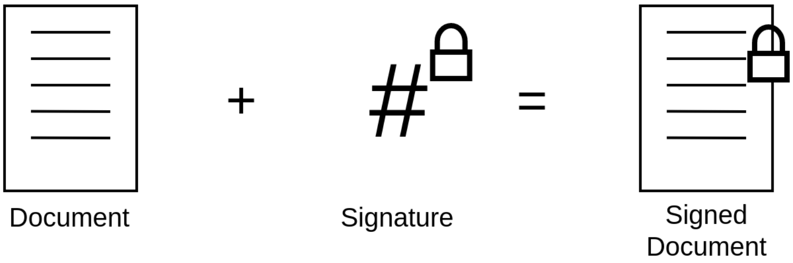
- Authenticity
- Integrity
- Non-repudiation

2.11.2 Create Digital Signatures

1. Hash Document
2. Sign Document using your own secret key

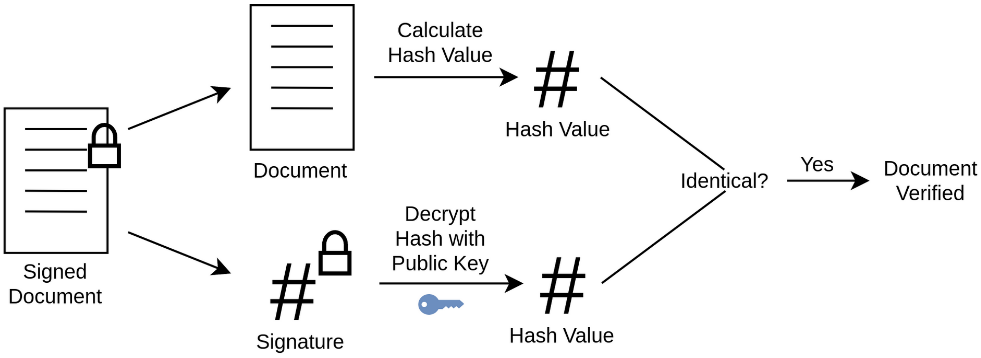


- The document together with the signatures makes a digital signed document



2.11.3 Validate Signature

- Signed document used to read out signature and normal document
- Document is hashed to create hash
- Signature is decrypted using public key to create original hash
- If both hashes are identical the communicate partner could validate the message is from you



3. Malware

- Malware is a combination of the words *malicious software*
- Describes software that affects or changes the functionality of a computer in a malicious way

3.1 Malware Classification

Malware can either be classified based on . . .

- . . . means and mechanism of distribution. So how the malware distributes, e.g via email or self-replicaitng.
- . . . the malicious behavior of the malware.

3.2 Basic Terms

Backdoors:

- Backdoor can be used to gain control over a system
- Manipualtes various parts of an operating system to ensure it's not discovered and to monitor the system

Bots/Zombies:

- An infected system with Internet connection that can be used for all kinds of malicious activities

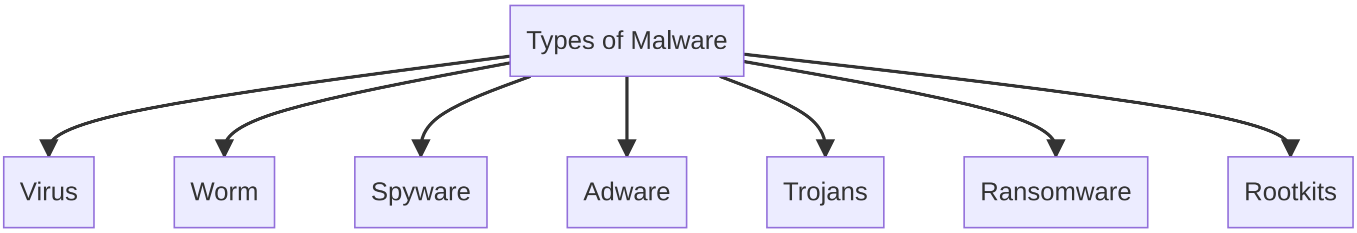
Botmaster:

- Entity that is in control of the bots

Botnets:

- Networks of compromised PCs that are remotly controlled for some functionality

3.3 Types of Malware



Virus:

- Malware which spreads uncontrollably on a system by infecting a host (e.g files, programs)
- Multiple types of viruses exist, e.g *boot viruses*, *file viruses* and *macro viruses*
- **Boot Virus:** A boot virus infects the boot sector of a system. A boot sector is a physical sector on a hard disk that contains information about booting an operating system. Hard disks and other storage devices have boot sectors. When the computer boots, the BIOS looks in the first sector of a storage device for the information needed to boot the operating system. The first sector/boot sector

is often either the Master Boot Record (MBR) or the Volume Boot Record (VBR). The boot sector is a potential attack point for malware. The advantage is that the boot sector always starts automatically and sometimes without any protection (modern hardware usually has protection mechanisms). So when the infected boot sector code runs, the malicious code is executed.

- **File Viruses:** A file virus insert code into executable files. When the file is opened or used may overwrite the file and cause damgae to the content of the file.
- **Macro Viruses:** Macro viruses are written in a macro langauge. (programming language which is embedded inside a software application). When the macro is executed, e.g when opening a program, the malicous code is executed.

Worm:

- Standalone program, i.e. it does not require a host program
- Replicates itself to spread via computer networks or removable devices
- Can distribute via email, p2p-networks, instant messaging, etc.

Spyware:

- Software that is installed with other programs
- Used to analyzes the behavior of users

Adware:

- Software that is installed with other programs
- Used to displays unwanted advertisements

Trojans:

- Malware that looks like normal, non-malicious software, but contains hidden features that negatively affect the system
- Needs cooperation of user to be installed

Ransomware:

- Program that blocks access to the users's data unless a ransom is paid

Rootkits:

- Set of tools that runs on target computer when you have gained access to it with `root` privileges
- Purpose to turn temporary access into an always-open door.

3.4 Damage that Malware can cause

- The attacker could encrypt the data and blackmail the affected person, e.g. either the person pays or the data is destroyed.
- The attacker could gain access to the computer to include it in a bot-network. The computer is then used for attacks like DDos, spam mails or other things without the owner's knowledge. The malware thus also harms third parties and not only the infected computer.

- The attacker might not have malicious intentions, instead it could be a pentester trying to discover security vulnerabilities in a system or of a software.
- The attacker could use the malware to obtain information. The purpose could depend on the party, e.g., a central authority, a hacktivist, or a blackhat.

3.5 Protective Measures of Malware

Scriptblocker/Adblocker:

- Deactives certain script which may change the behaviour of the site. E.g an adblocker blocks scripts so no advertsing is displayed.

Firewalls:

- Filters the traffic of a network and blocks possible attackers from unauthorized access.

Pentesting:

- Describes the analysis and testing of a system for weaknesses in order to close them.

Honeypots:

- Systems which are suppose to be infected by an attacker in order to study the attackers pattern und observe him. Honeypots are used to distract the attacker from the actual target.

Virus Scanners:

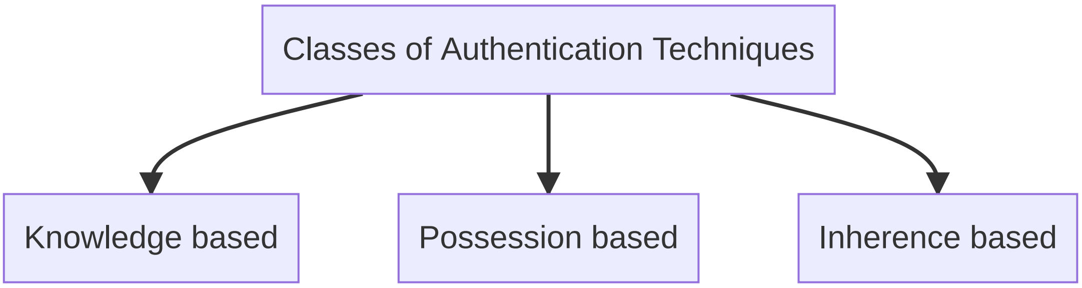
- Detection, deactivation and possibly deletion of malware
- **Manuel monitoring:** Scan for patterns in the code, based on a blacklist of viruses/virus patterns.
- **Realtime monitoring:** Check read write access

4. Authentication and Access Control

4.1 Objective of Authentication

- Clear identification and proof of identity
- Prevention of identity theft

4.2 Classes of Authentication Techniques



Knowledge-based:

- Knowledge-based authentication aims at proving a user's authenticity based on knowledge of private information about the them
- Used to prove that the person providing the identity information is the owner of the identity. (e.g Passwords, PINs, Cryptographic Keys)

Possession-based:

- Possession-based authentication aims to prove a person's identity using items that the user carries, usually a hardware device such as a security token or a phone. (e.g smart card, USB token, Sim card)

Inherence-based:

- Inherence-based authentication aims to verify the identity of a person based on the unique biological characteristics of a person. (e.g fingerprint, iris, face, voice, dna tests)

4.3 Unilateral Authentication vs. Mutual Authentication

Unilateral Authentication:

- One party identifies itself with another party (one-way)

Mutual Authentication:

- two parties authenticate themselves to the other one

5. Network Security

5.1 IPSec (Internet Protocol Security)

5.1.1 Basics

- IPSec extension of Internet Protocol (IP)
- IPSec is a set of protocols and algorithms (not a single protocol)
- Secure communication between hosts
- **Security for Layer:** Network Layer (Layer 3)

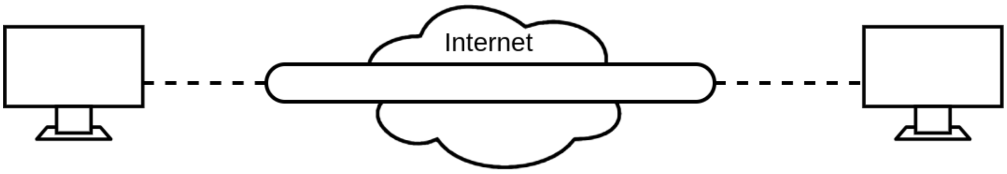
5.1.2 Protection Objectives

- Authenticity (Authenticity of data origin)
- Confidentiality (Confidential transmission of the user data)
- Integrity (Integrity and protection against replay attacks)

5.1.3 Operating Modes

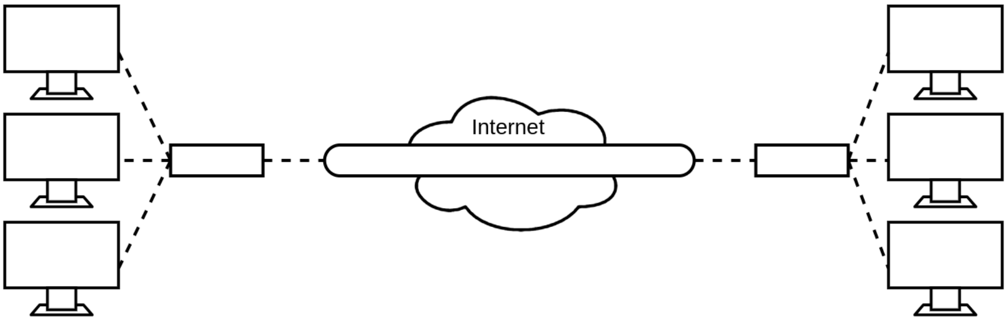
Transport mode:

- Direct secure connection between two hosts
- Hosts are the endpoints of communication
- Securing the payload



Tunnel mode:

- Tunnel between two gateways
- Different hosts use the tunnel
- Secures the connection between the gateways



- **Advantage:** Many devices but don't need to speak IPSec (endpoint don't need to change, Packet is changed on Gateways)
- **Disadvantage:** No End-To-End Connection

5.1.4 Protocol Components

- IPSec uses three protocols perform various functions
- These three protcols define how IP packets are modified
- Protocol constists of three components, *Authentication Header (AH)*, *Encapsulating Security Payload (ESP)*, *Internet Key Exchange (IKE)*
- AH and ESP can be combined

Authentication Header (AH):

- Formal: Adds header field to the sent packet that contains cryptographic hash of packet contents, which can be used to ensure payload was not modified during transmission.
- Provides data integrity, data origin authentication, and optional anti-replay services to transmitted IP-packets

Encapsulating Security Payload (ESP):

- Formal: Encrypts payload data. Additionally, a sequence number is added to the packet header so receiver can be sure it is not receiving duplicate packets or in wrong order.
-

Internet Key Exchange (IKE):

- Formal: To exchange the cryptographic keys
- Both host using AH or ESP require secrets keys for communication
- Key Management includes *key generation* and *key exchange*
- Used for mutual authentication
- Encryption method is choosen

5.1.5 Security Association (SA)

- Contains all information about IPSec connection
- Created when connection established for first time (usually when using IKE)
- SA contains *SPI*, *Receipient IP*, *Security Protocol (AH or ESP)*, *AH or ESP information*, *operating mode*, *lifetime of SA*, *sequence number*

SPI:

- The SPI Value is a 32-bit value which is used to uniqly identify an IPSec Conneciton. (Stored by SA, can get by SAD)

Sequence Number:

- The sequence number is a 32-bit value, which is increased by one for each packet. Used to check in which order the packets need to arrive.

5.1.6 Security Association Database (SAD)

- Every System uses a database to store it's SAs

5.1.7 Security Policy Database (SPD)

- Defines rules for IP-Packets
- One SPD for every IPSec computer
- SPD defines rules for the direction of IP-Packets (in or out)
  - `INBOUND`: Incoming packets
  - `OUTBOUND`: Outgoing packets
- SPD defines rules for, what happens with IP-Packetst:
  - `BYPASS`: Direct forwarding of the packet
  - `PROTECT`: IPsec must be used, reference to SA
  - `DISCARD`: Package is rejected

5.2 TLS (Transport Layer Security)

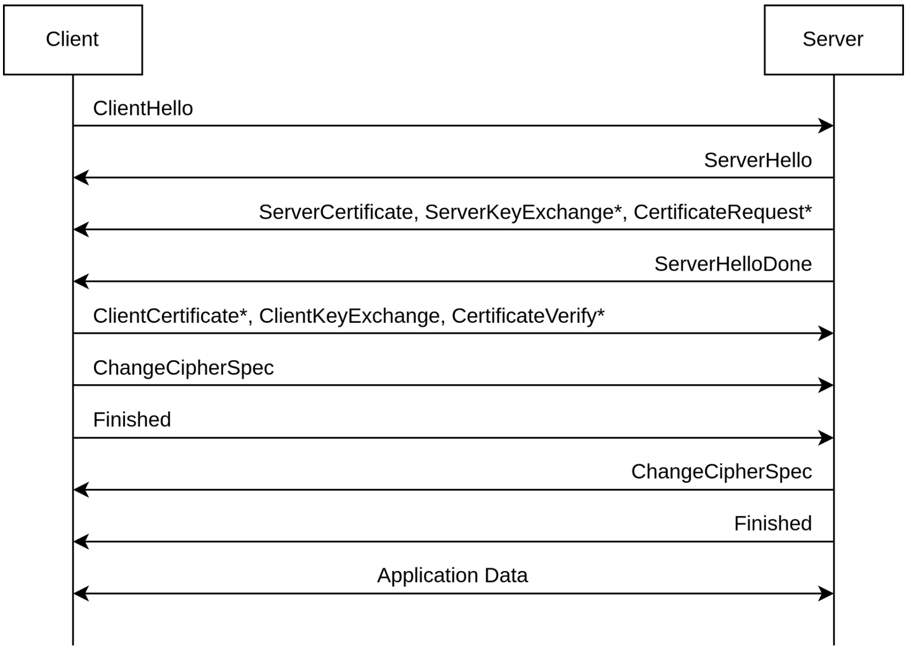
5.2.1 Basics

- Protocol providing end-to-end security
- Ensures the secure delivery of data over the Internet, avoiding possible eavesdropping and/or alteration of the content
- Implemented on top of TCP to encrypt Application Layer protocols like HTTP, FTP, SMTP, . . .
- The actual transmission of the data is done by the Application Layer protocols

5.2.2 Working of TLS

- Uses combination of asymmetric and symmetric cryptography
- Asymmetric cryptography used for securly sharing the cryptographic symmetric key (session keys)
- Symmetric cryptography used for data encryption and decryption using the transmitted session key

5.2.3 TLS Handshake


Client
Server
ClientHello
ServerHello
ServerCertificate, ServerKeyExchange*, CertificateRequest*
ServerHelloDone
ClientCertificate*, ClientKeyExchange, CertificateVerify*
ChangeCipherSpec
Finished
ChangeCipherSpec
Finished
Application Data

ClientHello:

- Supported protocol version

- Supported cipher suites (prioritized)
- Supported compression methods (prioritized)
- Random number R_c for detection of replay attacks

ServerHello:

- Selected protocol version
- Selected cipher suite and compression method

ServerCertificate:

- The server sends its Certificate message
- Certificates of the certification chain (including the public key of the server)

ServerKeyExchange*:

- Only when using DHE (Diffie–Hellman key exchange) used to exchange Key

CertificateRequest*:

- Only in case of mutual authentication
- Query of the client certificate

ServerHelloDone:

- ServerHello is over

ClientCertificate*:

- The client sends its Certificate message
- Client certificate (in case of mutual authentication)

ClientKeyExchange:

- For RSA: pre-master secret encrypted with public key of server (from certificate)
- For DHE: public DH key of client

CertificateVerify*:

- Signed hash of all messages exchanged so far

ChangeCipherSpecClient:

- Client and server calculate from shared secret (using keys and random numbers R_c and R_s)
- Subsequently all messages are encrypted

Finished:

- TLS Handshake is over

ChangeCipherSpecServer:

- All messages from now on are authenticated and encrypted

Finished:

- Analogous to client Finished message

5.2.4 What is end-to-end (E2E) security?

- Continosly security between two endpoints (sender and receipient)
- General seucity of sender and receipient, e.g secure transmission of data → confidentiality when sending from sender to receiver

Where in hybrid reference model can end-to-end be implemented?

- **Between Applications:** Application Layer (Layer 5) (security within the application)
- **Between Devices:** Transport Layer (secutity via TLS) or Network Layer (seucity via IPSec)

In which layer of hybrid reference model does TLS provides services?

- Implemented in Transport Layer and provides service to Application Layer

Protocol that can be secured by TLS?

- `HTTP`, `SMTP`, `FTP`, . . .

5.3 Firewall

5.3.1 Basics

- Monitoring and control of communication
- Rule-based filtering of (possibly malicious) data packets
- Two types of packet filters exist, *stateless* and *stateful*
    - Stateless means that no relationship is established between the packets
    - Stateful means that the state/relationship between the packets is detected (e.g. whether a connection already exists)

5.3.2 Stateless Packet Filters

- Played on Network and Transport Layer
- Filtering of data packets using TCP/IP and UDP/IP header
- Filtered information includes
    - Sender/receiver IP address
    - Sender/receiver port (service)
    - Protocol ID, ICMP type and code
    - Packet size (including some DoS attacks recognizable)
- The filtering is done using an *access list*
- Access list describes the rules for filtering

Access List

- Usually seperated list for incoming and outgoing
- Filter rules applied to each individual packet (not considering relationships between packets)

- Packet filtering based on first-match principle (List is read from top to bottom and first rule applying filters packet)
- Works using Network Address Translation (NAT)
- Differnt actions about what to do with packet exist
  - `PERMIT`: allow
  - `DENY`: discard
  - `REJECT`: discard with error message

Direction	Source IP	Destination IP	Protocol	Source Port	Destination Port	Flag	Action
...	...	...	...	...	...	...	...

Example Access List

- **Directions:** `IN`, `OUT`, `EITHER`
- **Source and Destination IP:** IP address, `EXTERNAL`, `ANY`
- **Protocol:** `TCP`, `UDP`
- **Source and Destination Port:** `25` (SMTP), `80` (HTTP), `443` (HTTPS), `>1032`
- **Flag:** `ANY`, `ACK`
- **Action:** `PERMIT`, `DENY`, `REJECT`

5.3.3 Stateful Packet Filters

- Can consider the communication context
  - Packets belong to the beginning of a new connection
  - Packets are part of an established connection
  - Packets are not part of a connection
- Packet filtering based on static rules (policies) and the previously observed packet traffic of the connections
- States of connection stored in a *state table*

State Table

Direction	Source IP	Destination IP	Protocol	Source Port	Destination Port	State	Action
...	...	...	...	...	...	...	...

Example State Table

- **Directions:** `IN`, `OUT`, `EITHER`
- **Source and Destination IP:** IP address, `EXTERNAL`, `ANY`
- **Protocol:** `TCP`, `UDP`
- **Source and Destination Port:** `25` (SMTP), `80` (HTTP), `443` (HTTPS), `>1032`
- **State:** `NEW`, `ESTABLISHED`, `ANY`
- **Action:** `PERMIT`, `DENY`, `REJECT`

5.3.4 Linux Kernel Firewal

- `Netfilter` is a Linux firewall
- Support stateless and stateful
- Work using first-match principle
- `Netfilter` uses three concepts *(ip)tables*, *chains*, and *rules* to provide certain functionality

Rules:

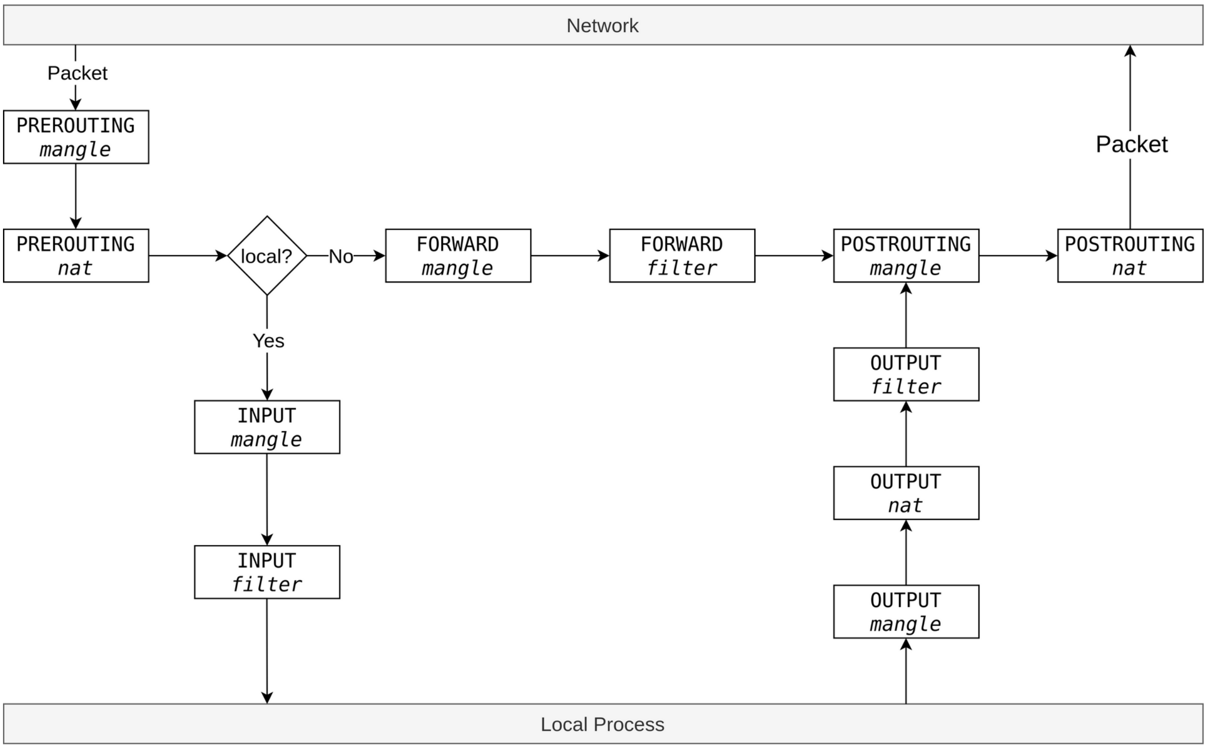
- Actions which the iptables can perform on the packets
- `ACCEPT`: The packet can pass
- `REJECT`: The packet is rejected and an error message is sent
- `DROP`: The packet is ignored and no response is sent
- `LOG`: Writes an entry in the syslog
- `REDIRECT`: The destination address of the packet is changed such that it is sent to the local computer
- `MASQUERADE`: The source address of the packet is replaced by the IP address of the interface on which it leaves the computer

(Standard) Chains:

- Chains define the path in which a packet can travel
- Chains are made up of *rules*, which define what action should be taken on packets
- `INPUT`: packet is delivered locally, i.e., the firewall is the target.
- `OUTPUT`: packet is created by the firewall.
- `FORWARD`: packet is routed (and not delivered locally).
- `PREROUTING`: modification (NAT) of packets before a routing decision is made.
- `POSTROUTING`: modification (NAT) of packets after the routing decision.

Tables

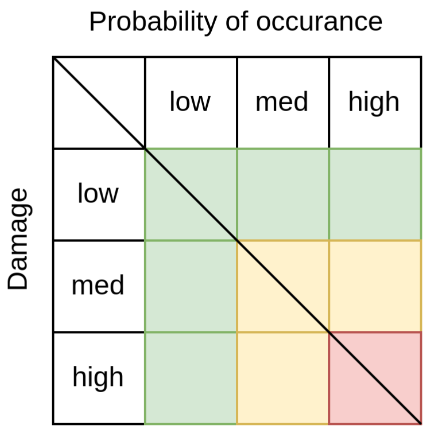
- Groups the standard chain in tables
- Used to test characteristics of packets and to accept or reject the packets based on the results of that evaluation
- `filter`: Contains filter rules for filtering packet (drop or accept)
- `nat`: Used to translate IP addresses and ports (network address translation).
- `mangle`: Used for packet manipulation.


Network
Packet
PREROUTING mangle
PREROUTING nat
local?
No
Yes
FORWARD mangle
FORWARD filter
POSTROUTING mangle
POSTROUTING nat
Packet
INPUT mangle
INPUT filter
OUTPUT filter
OUTPUT nat
OUTPUT mangle
Local Process

6. Network Security

6.1 Information Security (IS) Risk

- A risk is a potential impact that can have a either positive or negative impact on an organization
- An information security risk is the probability that a protection goal (confidentiality, integrity, availability, authenticity) will be broken, and the amount of damage resulting therefrom
- $\text{risk} = \text{probability} \cdot \text{damage}$


Probability of occurance

Damage

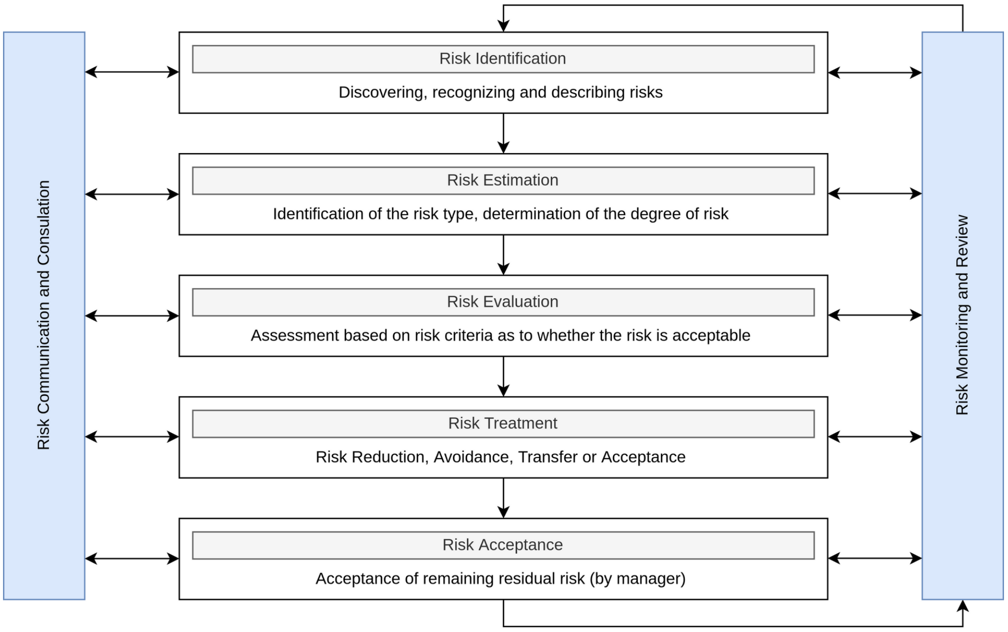
	low	med	high
low	🟩	🟩	🟩
med	🟩	🟨	🟨
high	🟩	🟨	🟥

6.2 Information Security Risk Management

6.2.1 Objectives

- Coordinated management and control of risks → effective use of resource to reduce most important risk
- Helps to prioritize measures to be implemented

6.2.2 Steps of Information Security Risk Management


Risk Communication and Consulation
Risk Identification
Discovering, recognizing and describing risks
Risk Estimation
Identification of the risk type, determination of the degree of risk
Risk Evaluation
Assessment based on risk criteria as to whether the risk is acceptable
Risk Treatment
Risk Reduction, Avoidance, Transfer or Acceptance
Risk Acceptance
Acceptance of remaining residual risk (by manager)
Risk Monitoring and Review

- **Risk Identification:** Disover and describe the risk
- **Risk Estimation:** What does it cost
- **Risk Evalulation:** Do we need to fix the risk?

- **Risk Treatment:** How do we treat the risk
  - **Risk Reducation:** Modify the risk in such a way it's reducing (add a technology, procedure or employee training)
  - **Risk Avoidance:** The conscious decision not to perform any action that make the risk become present
  - **Risk Transfer:** The risk is transferred to someone else (e.g insurance or outsourcing)
  - **Risk Acceptance:** The conscious decision to accept the existing risk but not to treat it
- **Risk Acceptance:** The manager checks if the done steps to reduce the risk are acceptable or more needs to be made (List of all risks is checked)

6.3 Information Security Management System

6.3.1 Basics

- Defines policies, methods, processes, and tools to ensure sustainable information security

Four areas of measures that should be addressed within an ISMS:

- **Technical:** Technical measure to solve problem
- **Organizational:** Who is responsable for what (Roles)
- **Personnal:** Traing personal
- **Infrastructure:** Phyiscal security of devices, building, etc.

Why ISMS should be a cyclic process:

- A cyclic process aims for constant improvement
- New components/products added will may cause new vulnerabilities
- The threat landscapes changes
- Change of technologies (technologies that hasnt been there before)

6.3.2 Standard for Information Security Management Systems

- ISO 27001
- IT-Grundschutz
- NIST Cybersecurity Framework

ISO 27001

- 14 domains of ISO 27001 provide the best practices for an information security management system (ISMS)
- **Access Control:** Ensure restricted access to employee, so they only see and use information which are assinged to their roles
- **Cryptography:** Use encryption for important data to ensure data integrity and confidentiality
- **Physical and Environmental Security:** Prevents unpermitted physical access and or damage to the environmental equipment of an organisation

- **Communication Security:** Protect the communcation of an orgainzation
- **Compliance:** Identify laws and regulations that apply in order to understand legal requirements
- . . .