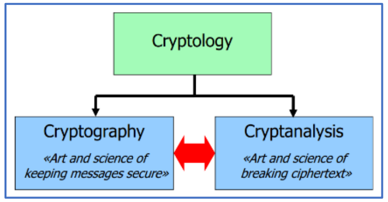
Introduction to Security and Cryptology

Ziele der Kryptographie

- Confidentiality — Only readable by desired receiver.
- Integrity — The receiver can determine if the message has been altered.
- Authenticity — The receiver can validate its origin.
- Freshness — The message is new and cannot be resent.



Abkürzungen

- P — Plain-Text
- C — Cipher-Text
- K — Key
- E — Encryption
- D — Decryption

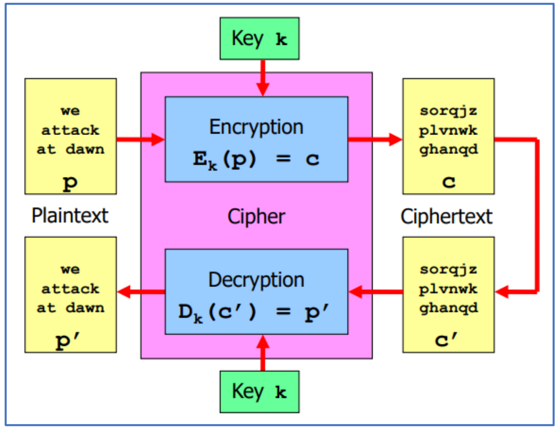
Encryption and Properties

Encryption is a transformation from P (plaintexts) to C (ciphertexts) under control of some key, chosen from K (key space).

The idea is to have a whole family of transformations, where each key gives a different transformation.

- $c = E(k, p) = E_k(p)$
- $p = D(k, c) = D_k(c)$

Each transformation must be reversible. It follows that $|C|$ can not be smaller than $|P|$. In fact most often we have $P = C$.



Passive Attacks (Eve) - Type of Attacks

- *Cypher-Text-Only Attack* — Bad Encryption — Find out plaintext or even the key
  Maybe the encryption isn't good and Eve can find out the plaintext just by looking at ciphertexts.
- *Chosen-Plain-Text attack* — Encrypt machine stolen — Find out plaintexts
  Maybe someone stole Alice's encryption machine for a few hours. Then Eve can encrypt some plaintexts of her choice and maybe find the plaintexts to ciphertexts she has intercepted.
- *Chosen-Siphertext attack 1* — Decrypt machine stolen — Learn about the key
  Someone has stolen Alice's decryption machine for a few hours. How much can Eve learn about the key by being allowed to feed ciphertexts to the machine?
- *Chosen-Siphertext attack 2* — Decrypt machine stolen — Learn about the key
  Someone has stolen Alice's decryption machine. How much can Eve learn about the key by being allowed limited information about decryption.

Introduction to Security and Cryptology

Cryptographic Work Factor

The number of times it takes to come upon the correct key is called (cryptographic) work factor.

- Work factor (WF) = average number of keys to try
- Work factor is usually given in bits: $\log_2(n)$, $n = key\ space\ size$

Entropy

- Entropy is maximal if all outcomes are equally likely

$$H = \sum_{i=1}^{n} p(i) \cdot \log_2\left(\frac{1}{p(i)}\right), \qquad H_{binary} = \sum_{i=1}^{n} \frac{1}{n} \log_2(n) = \log_2(n)$$

Entropy of Cryptographic Keys

Cryptographic keys are typically created with random generators, so they can be considered as elements of a random variable.

What's the entropy of a key with 128-bit?

- Independent with equal probability: $128 \cdot 1 = 128\ bits$
- Dependent with inequal probability: $p(0) = 0.25,\ p(1) = 0.75 \rightarrow 128 \cdot 0.81 \approx 104\ bits$

Relation between entropy and work factor

- Work Factor ≈ entropy, key size = max. entropy ≈ max. work factor

Information-theoretically secure

- Intercepting a ciphertext tells you nothing about the plain

Computationally secure

- Work factor ≈ key entropy

Modern Ciphers with One key

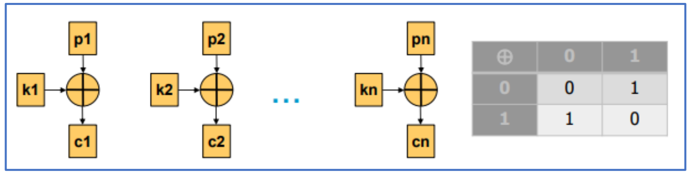
- Same key is used to encrypt and decrypt
- Wrong key gives high-entropy output:
  can tell when correct key used
- Good modern ciphers: key entropy ≈ work factor
- Cryptographic workhorse today: AES

Properties of Modern Ciphers

- Information-theoretical security is out of question
- Next best thing: computational security
- No cipher is known that can be proved to be computationally secure
- Cipher created → public review → gains acceptance

Vernam Cipher (a.k.a. One-Time-Pad)

- Key K: Independent bits with equal probability
- $C_X = K_X \oplus P_X$
- Each key may be used once
- Same Key used multiple times: $P_1 \oplus P_2 = C_1 \oplus C_2$



Random Oracle Model

- Different inputs give random unrelated outputs
- If A leads to B then A always leads to B

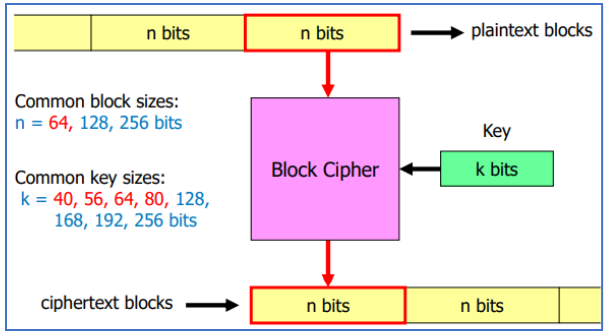
Secret Key Cryptography

Principle of Secret Key Cryptography

- The same key is used for encryption and decryption

Block Ciphers

- The plaintext is usually not a multiple of the block size.
  → Therefore a padding is needed.
- Use *Block Cipher modes* (ECB) / (CBC) for long messages



Block Padding (PKCS7)

- Padding data is used to fill up the final block
- Removal of padding must be easy
- PKCS7
  - n bytes must be filled, fill them with byte value n

`5F 6E 25 A3 36 54 90 0D D4 7F FA 05 05 05 05 05`

  - 0 bytes must be filled, add padding block with "10"

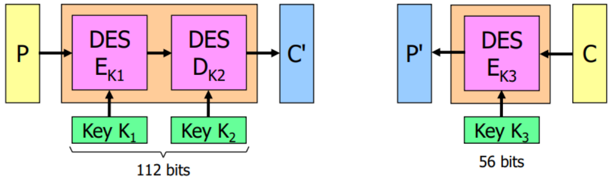
`10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10`

DES

- Key size: 56 bits → totally insecure
- Brute force attack is still the best attack known

Triple-DES

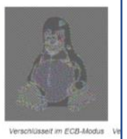
- Known-Plaintext Attack (MITM) → Cryptographic strength = $2^{112} + 2^{56} \approx$ 112 bits
- Secure but relatively slow



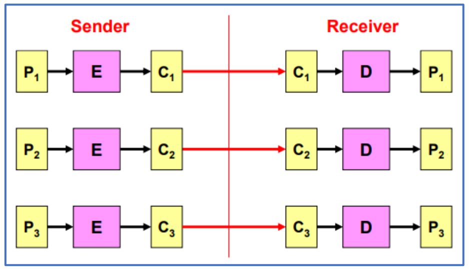
AES

- Cryptographic strength = 128 / 192 / 256
- Secure and modern standard
- Publicly defined

Electronic Code Book Mode (ECB)

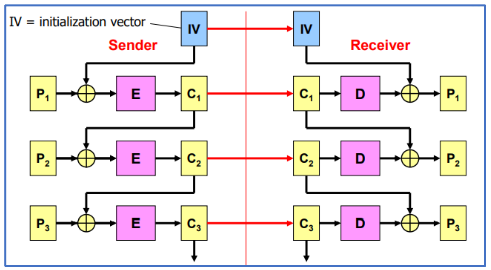


- Same P is always same C
- Susceptible to *replay attacks*
- Patterns may remain (Pinguin)



Cipher Block Chaining Mode (CBC)

- IV transmitted openly
- IV may only be used once
- Does not ensure the integrity



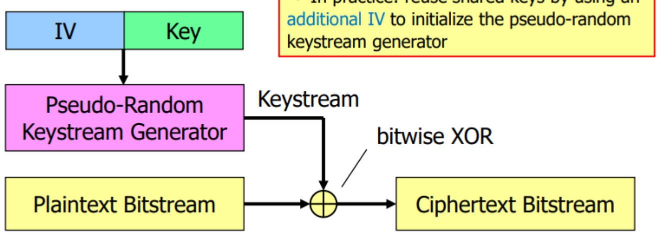
Secret Key Cryptography

Stream Ciphers

- No need to wait for full block
- Faster than block ciphers
- Most stream ciphers are broken
- Can be implemented with block ciphers (OFB, CTR)
- Decryption works by reapplying the keystream

$$C = P \oplus K \rightarrow P = C \oplus K$$

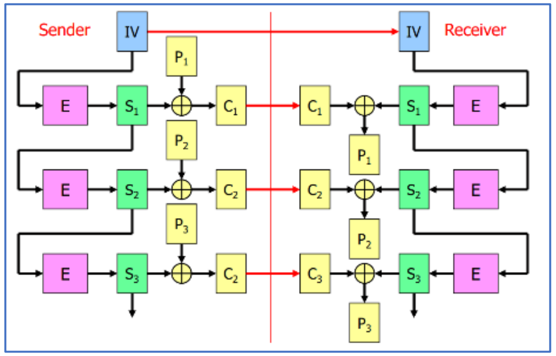
Do never use the same keystream twice!
→ In practice: reuse shared keys by using an additional IV to initialize the pseudo-random keystream generator



Plaintext Stream	1	1	1	1	1	1	1	1	0	0	0	0	0	0 …
Keystream	1	0	0	1	1	0	1	0	1	1	0	1	0	0 …
Ciphertext Stream	0	1	1	0	0	1	0	1	1	1	0	1	0	0 …

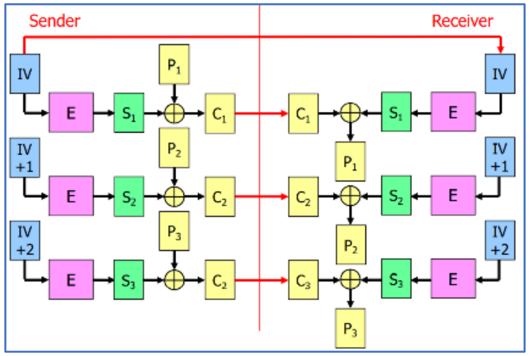
Output Feedback Mode (OFB)

- Stream cipher with block cipher



Counter Mode (CTR)

- Stream cipher with block cipher
- Allows for parallel encryption/decryption



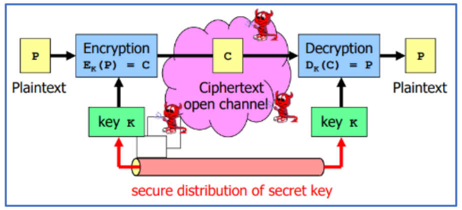
Security of Stream ciphers

- RC4, GSM A5/1 and A5/2 — Considered broken
- ChaCha20 — Considered secure (Supported by TLS) → Today's standard

Public Key Cryptography

Secure key distribution problem

- Secret key cryptography only works if the keys are exchange securely
- How can we obtain the key from *someone unknown*?
- How do we manage keys for *large number of participants*?

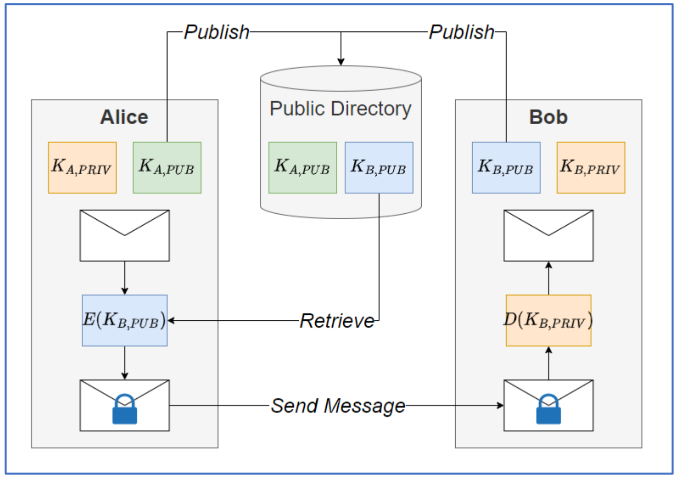


Principle of public key cryptography

- Public Key — Known to anyone — Encryption
- Private Key — Known to owner — Decryption

Properties

- Public and Private key fit together $D(E(M)) = M$
- Private key can't be computed easily from public key



RSA (Rivest-Shamir-Adleman)

Let (n, e) be RSA public key and (n, d) the corresponding private key. Let m be an integer with $0 \leq m < n$. Then $(m^e \; mod \; n)^d \; mod \; n = m$.

- Public Key (n, e)
- Private Key (n, d)
- $0 \leq m < n$
- $\varphi(n) = (q - 1) \cdot (p - 1)$

Key-Pair Creation

- Choose large primes $p, q \rightarrow pq = n$
- Choose e so that $\gcd(e, \varphi(n)) = 1$
- Compute d so that $ed \; mod \; \varphi(n) = 1$

Encrypt / Decrypt

- Encrypt $c = E(m) = m^e \; mod \; n$
- Decrypt$m = D(c) = c^d \; mod \; n$

RSA – Attacks

- Short message attack (Plain-Text-Attack)
  - Message m is short → Ciphertext c easy to decrypt
  - Prevent by using padding for encryption
- Low public exponent attack e
  - e is "low" and the same message m is sent e times
  - Prevent by using higher values for value for e
- Common factor attack
  - *gcd* of two moduli not sharing a common prime factor = 1
  - *gcd* of two moduli sharing a common prime factor → product of all
    1. *gcd = 1 → no common primes, else $p = gcd$*
    2. *if p has been found: $n/p = q$*

Public Key Cryptography

Groups

- $\circ$ Associative operation — For all $a, b, c \in G$ — $(a \circ b) \circ c = a \circ (b \circ c)$
- e Neutral element — For all $a \in G$ — $e \circ a = a$
- a' Inverse element — For all $a \in G$ — Exists: $a' \in G \rightarrow a' \circ a = e$

A generator of a group creates all elements when it operates on itself n times.

Example $(Z7,*)$

- $*$ Associative operation — For all $a, b, c \in G$ — $(a * b) * c = a * (b * c)$
- 1 Neutral element — For all $a \in G$ — $1 * a = a$
- a' Inverse element — For all $a \in G$ — Exists: $a' \in G \rightarrow a' \circ a = 1$

3 is a Generator

- $3 * 1 \ mod\ 7 = 3$
- $3 * 2 \ mod\ 7 = 6$
- $3 * 3 \ mod\ 7 = 2$
- $3 * 4 \ mod\ 7 = 5$
- $3 * 5 \ mod\ 7 = 1$
- …
- $3 * 13 \ mod\ 7 = 4$

*	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Discrete Logarithm Problem

- Idea — One-Way function
  - Direction A: Easy to compute
  - Direction B: Hard to compute
- Example: $(Zn^*,^*)$ for large n

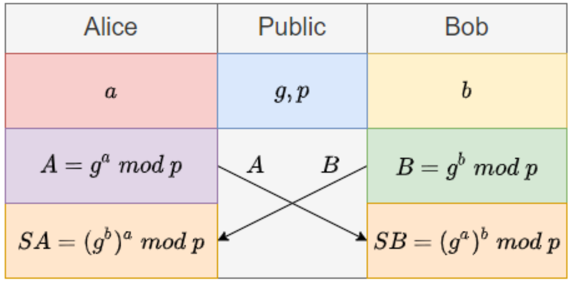
Integrated Encryption Scheme (IES)

- Solves Offline Problem (of DH)

Diffie-Hellman key exchange

1. Alice and Bob — agree on large prime p
2. Alice and Bob — agree on some generator g of $(Zp^*,^*)$
3. Alice — chooses a, $(1 < a < p)$
4. Bob — chooses b, $(1 < b < p)$
5. Alice — sends $A = g^a \ mod\ p$
6. Bob — sends $B = g^b \ mod\ p$
7. Alice — computes $SA = B^a = (g^b)^a \ mod\ p$
8. Bob — computes $SB = A^b = (g^a)^b \ mod\ p$

$SA = SB$ is the shared secret only Alice and Bob know.



The secret may not be suitable for to use as a symmetric key

- Some bits may always be zero or one
- The secret may be too long or too short

The secret is put through a *key derivation function* (KDF) to get the key.

Problems

- What if Bob and Alice are not speaking to each other?
- Eve can make a Man-in-the-Middle Attack.
- Bob and Alice need to be Online.

Data Integrity and Authentication

Cryptographic Hash Functions

A cryptographic *one-way hash function* maps a variable-length input bit string to a fixed sized output bit string.

Important Properties

- *Efficient computation*
- *Pseudo-random*
- *Preimage resistance* — Given a hash, it is practically impossible to find a massage that produces the hash
- *Collision resistance* — It is practically impossible to find any two messages that map to the same hash

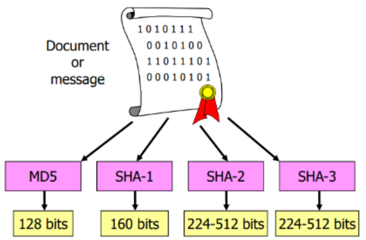
A hash function should behave like a random oracle. Different inputs → totally unrelated outputs.

Check if a message has been tampered

- Use a hash function together with a secret key (*Message Authentication Code*)
- Use a hash function together with *digital signatures*

Hash Functions

- MD2 — Very bad
- MD5 — Very bad
- SHA-1 — Getting weaker
- SHA-2 — considered secure
- SHA-3 — considered secure (best)
- SHA-256 good



Authentication with passwords

- Popular, easy to implement, changeable

Password security problems

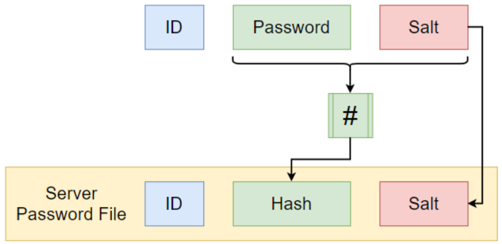
- Sniffing — Passwords are transmitted in plaintext — Prevent with Protected links (TLS)
- Phishing — Show fake login screen to capture password — Prevent with Certificates
- Online Attacks — Password guessing — Prevent with Slowdown / blocking
- Offline Attacks — Compromises system and gets pw from db — Do not save passwords in plaintext
- Password Re-use — Remove password-strength

Cracking Hashed Passwords

- Try passwords — Compute the hash and check if fit matches any hash in the password file
- Test strategically — Dictionary attack, Minor variations…
- Precompiled attack — Compute a huge list of passwords and hashes (beforehand)

Protect password hashes

Add a random value called salt before hashing to prevent *precompiled attacks*.



Collision calculation

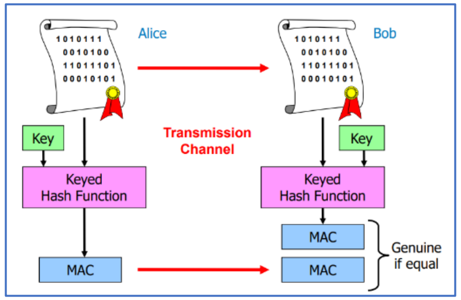
$$n = Hash_{Length}\ (in\ bits)$$

- Two random Hashes — $2^{\frac{n}{2}}$
- Collision with given Hash — 2^{n-1}

Data Integrity and Authentication

Message Authentication Codes (MAC) = Message Integrity Check (MIC)

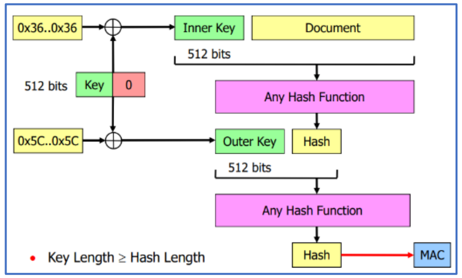
The idea is to use a key in addition to the hash function to prevent an attacker from changing the message and calculating a new hash…



Workflow for document (Using Secret Key Cryptography)

1. Alice Send document
2. Bob Apply “Keyed Hash Function” = MAC_1
3. Alice Apply “Keyed Hash Function” = MAC_2
4. Alice Send MAC_2
5. Bob Compare MAC_1 and MAC_2

HMAC is a concrete implementation of the MAC concept.

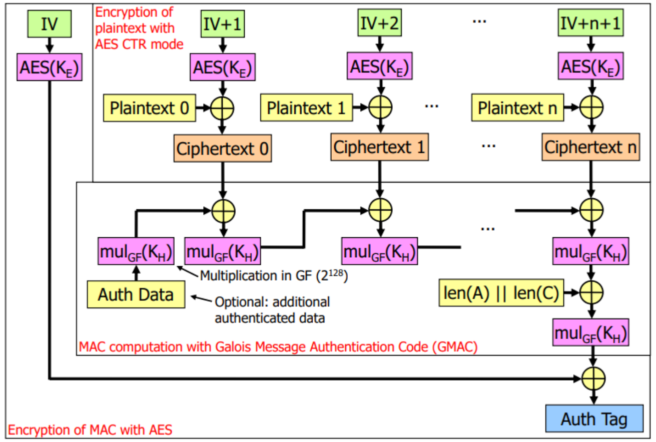


- Generate Inner K_I and Outer Key K_O
- Hash $H(K_I, M) = H_1$
- Hash $H(K_O, M) = H_2 = MAC$
- Keyed Hash Function: $H\big(K_O, H(K_I, M)\big)$

Galois Counter mode (GCM)

Only encrypting a message doesn’t provide Integrity-Protection → GCM

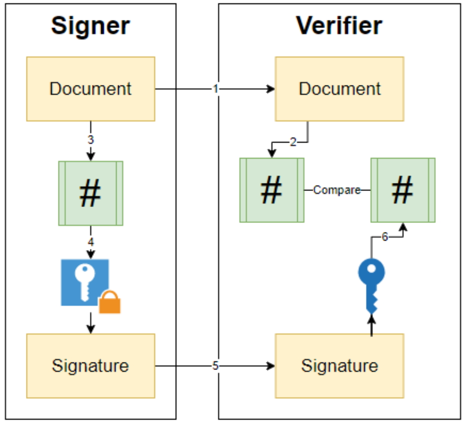
- Combine Integrity and Encryption
- Approach «Encrypt then MAC»



Digital Signature

Workflow for document (Using Public Key Cryptography)

1. Signer Send document M
2. Verifier Hash document $M \rightarrow H(M) \quad = X$
3. Signer Hash document $M \rightarrow H(M)$
4. Signer Encrypt document $H(M) \rightarrow E(H(M))$
5. Signer Send document $E(H(M))$
6. Verifier Decrypt document $D\big(H(M)\big) \rightarrow H(M) = Y$
7. Verifier Compare hashes $X == Y$



MAC

- Fast and efficient
- Receiver must know the secret key

Digital Signatures

- Public key is openly available
- En- / Decryption are time intensive

Digital Certificates

A certificate is a signed statement by a trusted third party that a certain public key belongs to a certain name. It is a means to authenticate a public key. It is not a means to authenticate a communication partner.

Digital Certificates

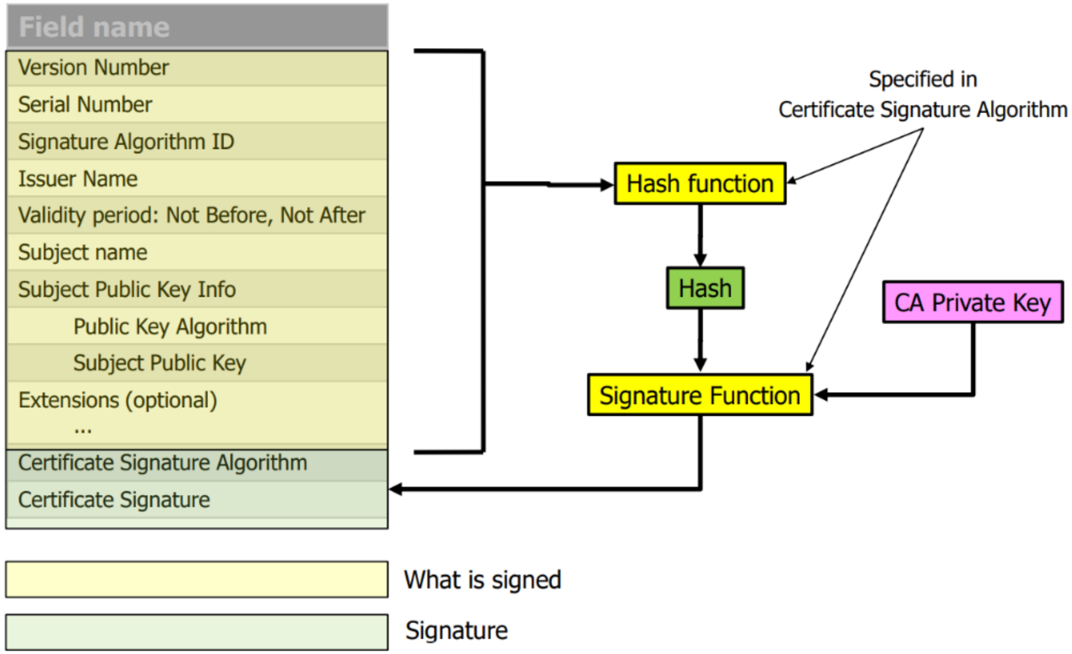
- Bind together certain things
- Issued and signed by trusted third party called Certificate Authorities (CA)
- Valid for a certain period

Public Key Infrastructure (PKI)

- Verify public key in the certificate by verifying the signature

The X.509 standard

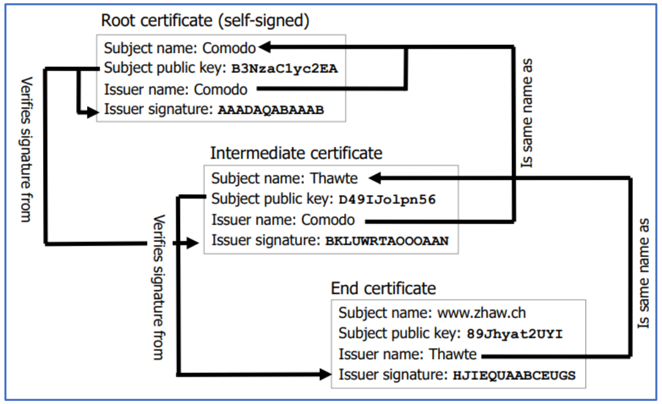
- Dominating standard for certificates these days



Concept of trust hierarchies

For $i = [1, n-1]$

- $C[i].issuer = C[i+1].subject$
- $C[i].signature$ can be verified with $C[i+1].publicKey$



Root Certificates

Any certificate chain ends with a root certificate.

- Can be verified with its own public key
- Root certificates are always self-signed
- Usually stored in your browser (from trusted parties)

Validation

- Obtain and verify (validity) root certificate
- Verify server certificate signature with root public key
- Compare root certificate subject with server issuer

Digital Certificates

Certificate revocation: CRLs and **OCSP**

If the private key gets stolen the certificate should be revoked

- Wait until the certificate expires
- Contact issuer to invalidate the certificate

Root Certificate Revocation

- Remove root CA from applications and systems
- All certificate issue by the root CA will also be invalid

Certification Revocation List – CRL

CA provides a list of all revoked certificates that can be downloaded to verify if a certificate has been revoked.

Issues → Limited support

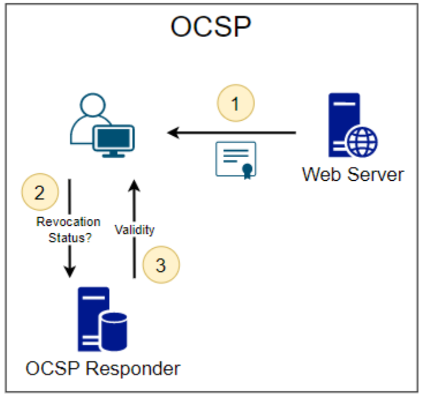
- Only updated every couple of days
- Can’t remove a certificate from CRL
- Many certificates may go onto CRL
- CRLs only get larger (delay)

Online Certificate Status Protocol – OCSP

Issuing CA can be asked (Client) for the status of a specific certificate.

- Status information is updated more often
- Only little information is exchanged
- OCSP must always be available
- OCSP is the single point of failure
- Vulnerable to DoS attack against OCSP

Most clients implement soft fail (good), if the response is not received (in time).



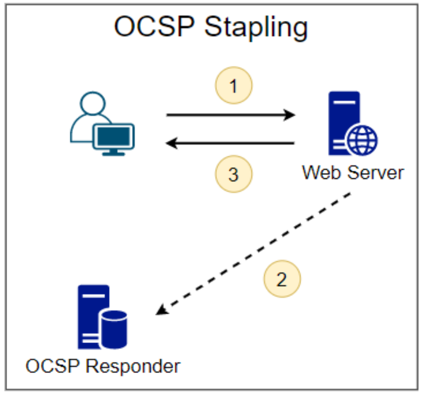
OCSP stapling

The server queries the OCSP-Responder listed in the certificate itself and caches the responses.

- Client gets OCSP response from server during TLS handshake
- Less dependent on the availability of OCSP-Responder

Limitations

- Vulnerable time window like in CRL
- Fallback to OCSP if no valid response is cached



Secure Communication and Layer 2 Security

Goals

- Confidentiality — Only the communication endpoints can read the data
- Integrity — The endpoints can detect if data was manipulated
- Authenticity — Masquerading as an endpoint is not possible

Secure communication protocols at different OSI layers

- Higher Layers — end-to-end, easier to deploy, less general
- Lower Layers — hop-to-hop, difficult to deploy, more general

Layer	Protocols
Application Layer	S/MIME, PGP
Transport Layer	SSL/TLS
Network Layer	IPsec
Data Link Layer	EAP, 802.1x, WEP, WPA, WPA2/802.11i
Physical Layer	Quantum Cryptography (Appendix only)

Extensible Authentication Protocol (EAP)

- Most used standard for L2 authentication
- Supports a wide range of authentication mechanisms

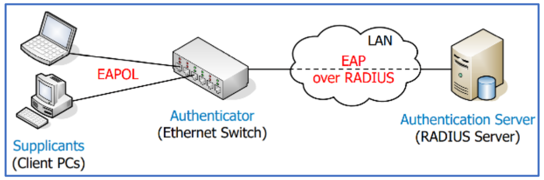
Security of IEEE 802.1x

Protect access to LANs

- LAN ports are not open per default
- Authentication (EAP) is required before accessing a port
- Protects against someone plugging in an unauthorized device
- Attacker needs access to an authorized physical device

RADIUS (Remote Authentication Dial-In User Service)

- Server / protocol for authentication



Temporal Key Integrity Protocol – TKIP

- RC4 (not secure)

Counter Mode CBC-MAC Protocol – CCMP

- Based on AES (confidentiality and authenticity / integrity)

Limits

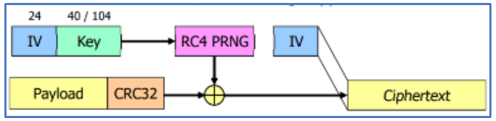
- Software vulnerabilities, malware, DoS attacks
- Protocol design and implementation flaws
- Configuration and usability flaws

Security mechanisms of WLANs

- No cable → sniffing packets is very easy
- Authentication to network is needed

Wired **E**quivalent **P**rivacy (WEP)

- All users use the same key long-term
- 40- / 104-bit keys — only 104 is good
- 24-bit IV — IV to short → repetitions



Wi-Fi **P**rotected **A**ccess (WPA) and IEEE 802.11i (WPA2)

- Authentication at Access Point required
- Key-Exchange during authentication
- WPA: TKIP (default) and CCMP
- WPA2: TKIP and CCMP (default)

TKIP (Temporal Key Integrity Protocol)

Uses a MAC to protect integrity, however it uses algorithms with well-known weaknesses → *Insecure*

CCMP (Counter Mode CBC-MAC Protocol)

Uses block cipher to achieve encryption, authenticity, and integrity protection. → *Considered Secure*

Firewalls

Firewall basics

A firewall is a device that sits *between two or more networks* to *control the packet flow* between them. Based on the security policy one or more firewalls are installed and configured.

A firewall can

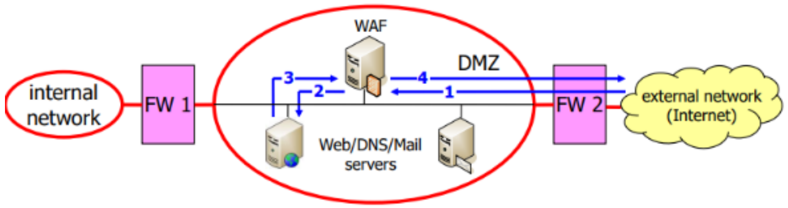
- Control access from internal network to the Internet and vice versa
- Block malicious incoming web traffic

Packet-filtering firewall (Network Layer)

- Using rules, that decide whether to forward packets
- Inspect headers of network and transport layer protocols
- Pro: Very fast as they only check layer 3 and 4 protocol headers
- Limitation: Only control who is allowed to talk to whom

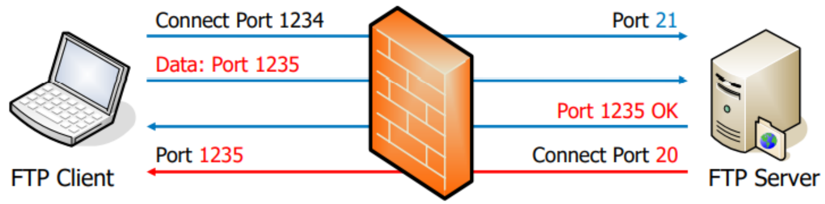
Application layer firewall (Application Layer)

- Split end-to-end communications
- Inspect application layer data
- Pro: Allows deep inspection of all data exchanged
- Limitation: Relatively slow, limitations with encrypted data



Stateless firewall

- Both directions need to be configured
  - Every IP packet is handled completely isolated from all others
  - Firewall does not keep track of ongoing communications
- More open than needed:
  Replies from a server are allowed without a previous request from a client
- Limited support for complex protocols



Stateful firewall

Checks individual packets, tracks communication relationships, and maintains state information.

- Today standard
- Easier to configure (fewer rules)
- Return traffic is only allowed on demand
- Allows support for complex protocols
- A packet can be in one of four states:
  New, Established, Related, Invalid

Benefits and Limitations

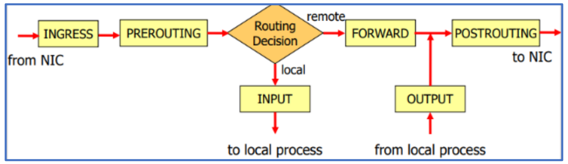
Firewalls are only useful against attackers outside the internal network.

Firewalls

Linux netfilter/nftables

netfilter is a mechanism that allows to access the packets in the network stack to analyze, modify, extract, and delete them.

- Hooks that are called at different points during packet processing



nftables is a packet classification and mangling framework that runs on rulesets that are applied to the packets

- Table — Container for specific type of package
- Chain — Container with rules for a specific hook
- Hook — Called at different points of processing
- Priority — Lowest priority first until rule is accepted
- Policy — Default behaviour
- Rule — Classification + Action
- Classification — What packets does a rule apply to
- Action — What to do with the packet

```
# Table: Table used for packetType
table packetType tableName
{
  chain chainName {
    # Type:         Type of Chain
    # Hook:         Applied when hook is called
    # Priority:     Defines order of packets (lowest first)
    type filter hook input priority 0;

    # Policy:       Default behavior
    policy accept;

    # Rule:         Classifications + Action
    ip saddr 8.8.8.8 type echo-request drop
  }
}
```

Port scanning

Technique to determine the services that run on a host.

- Useful for attackers and system admins
- Port scanners…
  - Check if the host is available by pinging it
  - Establish TCP connections to the ports
- Nmap: Most popular port scanner
- TCP scan of port 80 of www.zhaw.ch
  - *nmap -p80 www.zhaw.ch*

Nftables Examples

Drop all packets going to IPv4 address 8.8.8.8

- *Ip daddr 8.8.8.8 drop*

Accept all packets coming on interface eth2

- *Iifname eth2 accept*

Accept all IPv6 packets carrying TCP

- *Ip6 nexthdr tcp accept*

End-to-End Communication Security

End-to-end communication security

Secure communication protocols that protect the traffic flow between two end hosts or two applications on these end hosts.

- People that can access the communication channel only see encrypted data and cannot access the plaintext data.
- Transport Layer Security (TLS) on layer 4 (for TCP), IPsec on layer 3

TLS overview

- TLS works on top of TCP
- Does not have to worry about data loss / retransmission
- Authenticated, integrity-protected and confidential data exchange
- TLS connections are closed when the underlying TCP connection is closed.

TLS 1.3

- Client and Server have no previous association
- Want to boot a connection where…
  - Client and server agree to secret key material
  - Server is authenticated to the client

TLS 1.3 Building Blocks

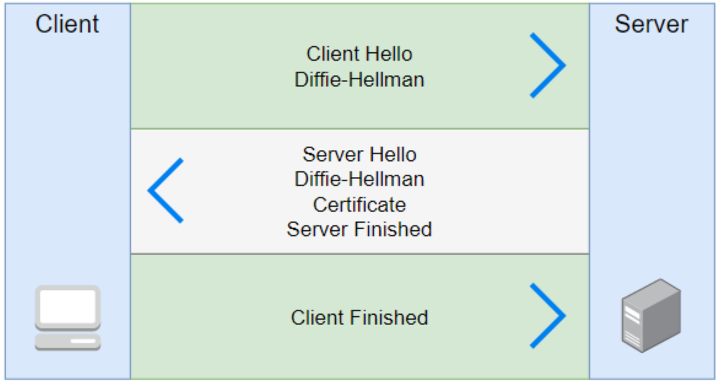
- Block Ciphers, Encryption Modes, DH, Public Key with Certificates…

TLS Handshake – Steps

1. Client and server negotiate crypto algorithms
2. Client and server perform Diffie-Hellman
3. Client and server generate handshake keys
4. Server authenticates to client
5. Client and server prove to one another that no one has altered previous messages
6. Client and server generate data keys

TLS 1.3 Handshake

1. Client Hello
   - Supported TLS Version (TLS 1.3, …)
   - Supported Algorithms (AES, GCM,…)
   - Diffie-Hellman Key Exchange (Step 1)
2. Server Hello
   - Selected TLS Version (TLS 1.3)
   - Selected Algorithm (AES, GCM,…)
   - Diffie-Hellman Key Exchange (Step 2)
   - ChangeCipherSpec (optional)
3. Certificate / Certificate-Verify
   - Certificate and Certificate chain
   - Signature over previous messages
4. Server Finished
   - Hash over all messages
5. Client Finished
   - Hash over all messages

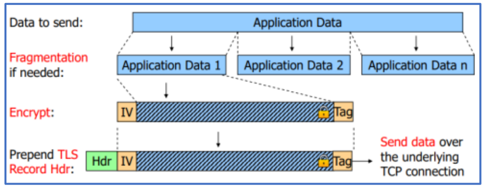


Pascal Isliker

TLS Session Resumption

If there is an existing TLS session between a client and a server application, additional TLS sessions can be established without any public key computations.

TLS Data Exchange



1. Fragmentation
2. Encryption + prepend IV
3. Prepend Header

A sequence number is used to ensure the order of all TLS records, when they arrive.

Cipher suites are a set of cryptographic algorithms: “TLS_AES_128_GCM_SHA256”

State of TLS *(Transport Layer Security - Wikipedia)*

- SSL 2.0, 3.0 major vulnerabilities supported by some servers
- TLS 1.0, 1.1 meh supported by 45% of servers
- TLS 1.2 good supported almost universally
- TLS 1.3 good and fast

Latest browser versions are secure

- Only support TLS 1.0 – 1.3 and implement browser-side fixes for TLS 1.0 or simply disable it
- Most browsers don’t support SSL 2.0 / 3.0 anymore
- TLS 1.3 is supported

DTLS

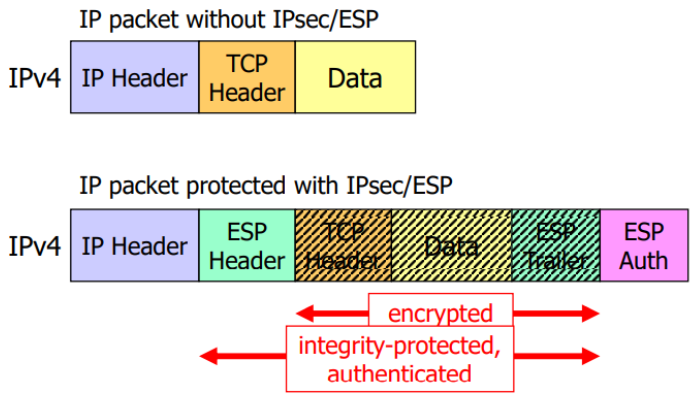
- Works on UDP instead of TCP

IPsec

- Protects IP packets payload
- Protocol for network layer (IP)
- Security between two hosts
- Internet Key Exchange (IKE) in IPsec is like TLS handshake

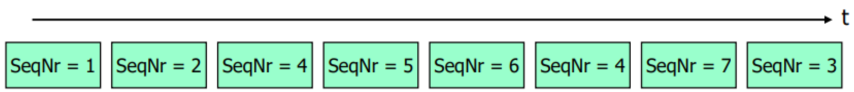
IP Packet Protection with ESP

- ESP = Encapsulating Security Payload
- ESP Provides *confidentiality, authentication, and integrity*



IPSec Sequence Number

- Order packets and drop duplicates



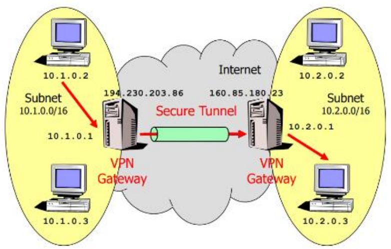
Virtual Private Networks

A Virtual Private Network VPN is a private network within a public network.

- Private: Outsiders can neither read nor modify
- Virtual: Privacy is achieved by virtual methods

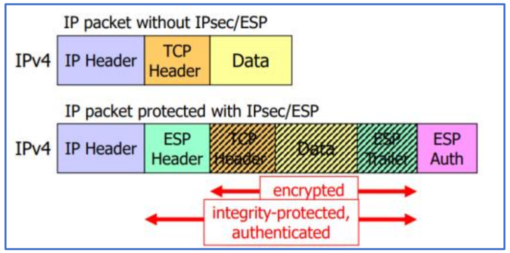
Usage

- Securely connect two remote networks
- Allow partner / customer company selectively accessing internal services



IPsec

- Offers a secure tunnel
- Hide internal IP addresses



OpenVPN

- Partly based on TLS
- Application layer tunnel
- Uses UDP as underlying protocol
- User Space

Comparison

- Equal in terms of Security
- IPsec is used more widely
- OpenVPN less loaded with features → easier configuration
- OpenVPN runs in the user space

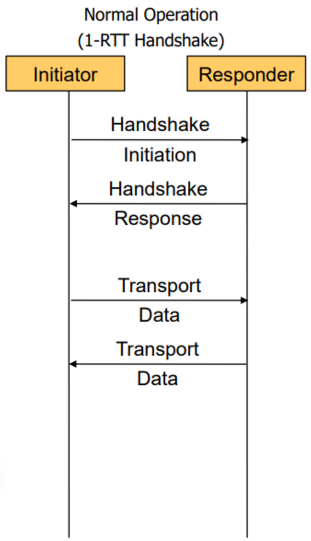
VPN gateways

- Endpoints to secure channel
- Apply and remove cryptographic protection
- Channel between VPN endpoints = secure tunnel

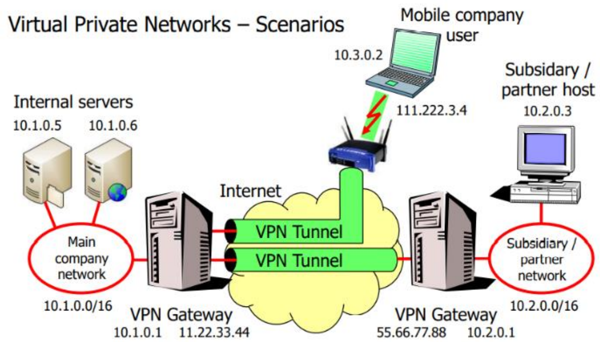
IPsec and OpenVPN are the most important protocols for VPNs.

WireGuard

- Works on Layer 3 (like IPSec)
- Little configurability
- Uses modern crypto primives
- Has a 1-RTT handshake
- Has Perfect Forward Secrecy
- Has DoS-mitigation techniques



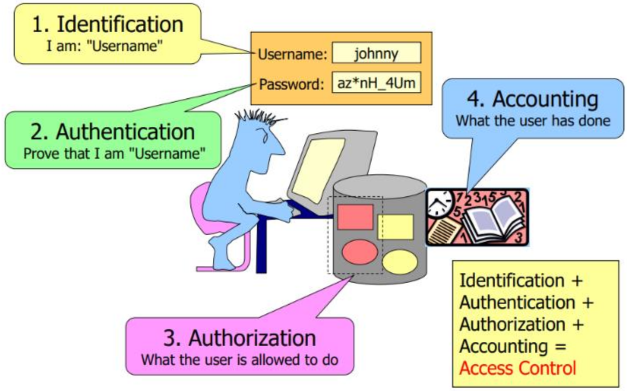
Virtual Private Network scenarios in practice



User Authentication and Authentication Protocols

Terminology

- Identification — My name is «XXX»
- Authentication — Prove identity of user
- Authorization — What the user is allowed to do
- Accounting — What the user has done



Authentication Factors

- Knowledge — Password, PIN, Shared Secret
- Possession — Smartcard, Token, Mobile, TAN List
- Biometrics — Fingerprint, Face, Voice

Two factor authentication

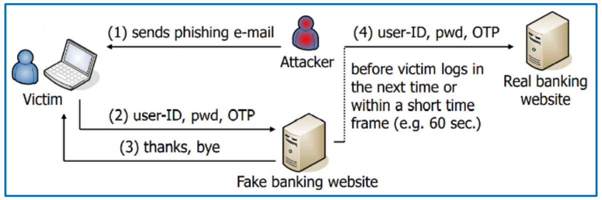
- Combination of two methods (Possession, Knowledge, Biometrics)
- Considered as strong user authentication

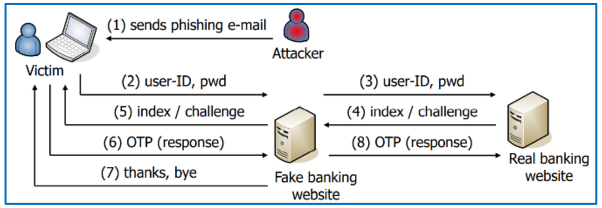
Direct and indirect authentication

- Direct — Send credentials to server, Server authenticates the user
- Indirect — Send credentials to server, Handle authentication via authentication server
  RADIUS, NTLM, Kerberos, Shibboleth

OTP Generations

- 1st Generation: Increased security
  - Mainly used Possessions (TAN and One-time password tokens)
  - Vulnerable to e-mail phishing attacks
- 2nd Generation: Prevent E-Mail Phishing
  - Challenge-Response based approach (iTAN, card readers)
  - Authentication → Challenge
  - Vulnerable to MITM attacks (Online Phishing)
- 3rd Generation: Mobile Phone-based Approach
  - Mobile TAN: Authentication → Approve with SMS code
  - MITM still possible → Same security as 2nd Generation



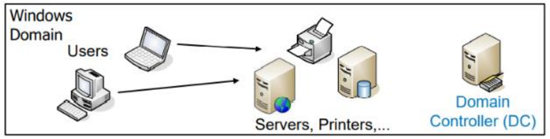


User Authentication and Authentication Protocols

Windows Domain Authentication

A windows domain is a collection of users and services, access is controlled by a Domain Controller (DC). Within a windows domain, we have centralized administration & SSO.

- One account per user and domain
- DC can authenticate users
- Credentials need to be entered only once

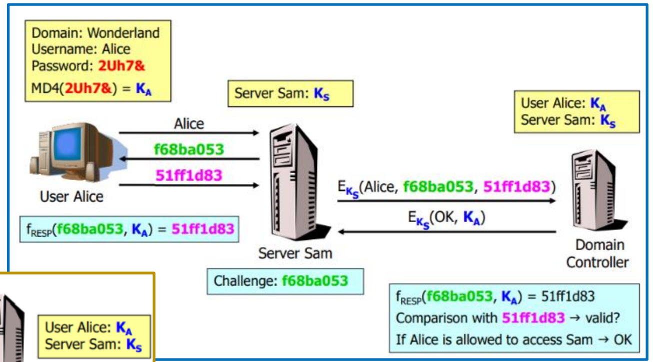


All users and servers must trust the domain controller.

Windows NT LAN manager (NTLM)

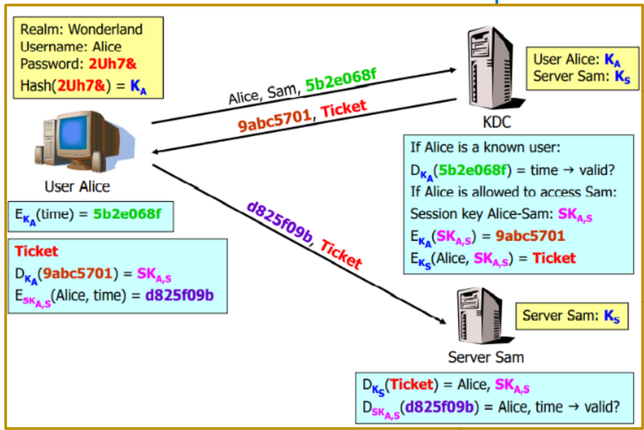
Original authentication protocol in Windows domains.

- Provides "some security" for strong passwords



Kerberos (Considered secure)

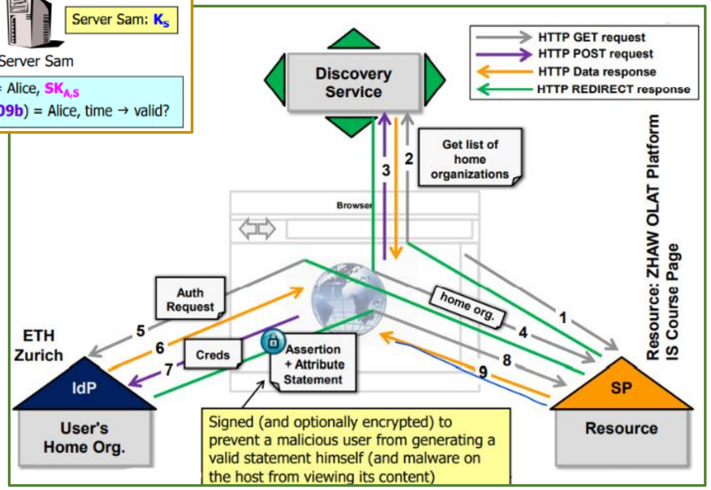
- Each party
  - shares a common secret with a centralised server
  - trusts the Key Distribution Center (KDC)
- Ticket based approach → more efficient than NTLM
  - Authentication Service
  - Ticket Granting Service
- Prevents replay attacks with timestamps
- Weak password → offline attacks possible



Shibboleth (Considered secure)

- Federated is a system for federated identity management
- Users have one credential that is stored and managed by his home security domain
- Components: Organizations, Users, Service providers (SP), Identity providers (IdP)

1. Connect to Resource (Moodle) Retrieve list of home organizations
2. Authentication Request Select organization and redirect to Identity Provider
3. Authentication and Access Enter credentials and access resource



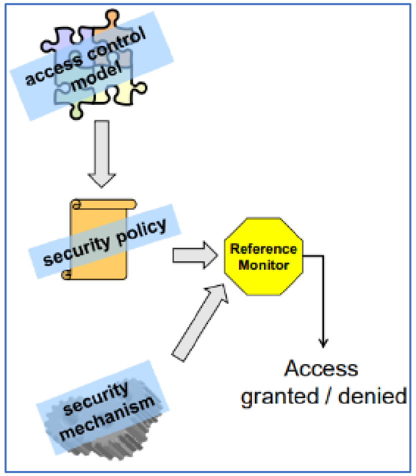
Authorization

Basics

Authorization determines if a user is allowed to access a particular resource to perform a specific operation. It's a core component of every operating system.

Reference Monitor Concept

- A reference monitor is responsible to enforce access rights
- The reference monitor must be invoked prior to execution of every security-sensitive operation
- To decide whether access should be granted, the reference monitor needs a rule set (security policy)

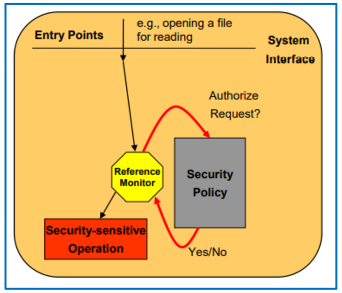


Security Policy

- Defines who is allowed to do what on a system
- Should be flexible in terms of configuration

Security Mechanism

- Method / Structure used to implement the security policy



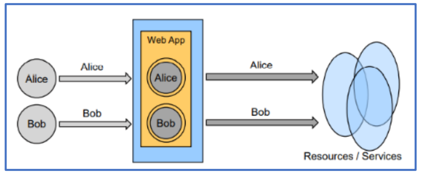
Multi-tier applications

- Based on Multi-Tier architecture
- Three-tier architecture
  - Presentation
  - Logic
  - Data

Impersonation/Delegation Model	Trusted Application Model
+ Auditing: System-integrated auditing facilities can be used. *The subject's security context is maintained across the physical tiers, auditing is possible across tiers.* + Granular access controls at the backend. No «one subject takes all» approach.	+ Minimal back-end access rights management. *Only the service account accesses back-end resources.* + No direct access to back-end resources. *Application authorization at the middle tier is required.* + Scalability through efficient connection pooling.
- Scalability: No efficient connection-pooling. *(needs to be done on a per subject basis)* - Increased administration effort. *Many back-end resources and/or subjects implies a significant administration effort to manage access rights at the back-end resources.*	- Auditing: Resource access can not be done based on information from the back-end only. - Increased risk from server compromise for back-end resources.

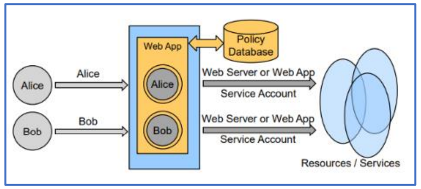
Impersonation / delegation

- The subject "survives" beyond the middle tier
- Generate impersonation token that reflects the subject's access control data.



Trusted application model

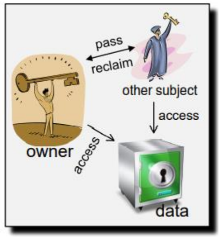
- Access to backend via. service account
- Requests are granted/denied to subjects based on a set of access policies



Authorization

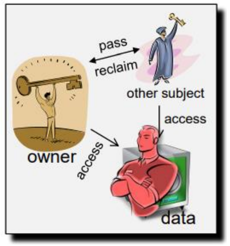
Access control models are framework dedacting how subjects access objects.

Discretionary Access Control (DAC)



- Based on *discretion of the owner* of an object
- Example: Modern OS (Windows, Linux, Mac)
- Usually includes a user that can bypass restrictions (root / admin)
- Uses **A**ccess **C**ontrol **L**ist ACL or Capabilities

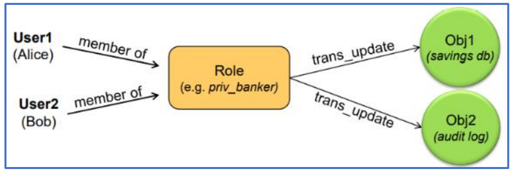
Mandatory Access Control (MAC)



- A *system wide policy* determines who is allowed to access
- Individual user cannot alter
- Policy is configured by a policy administrator
- Bell-LaPadula
  - Multi-level security model
  - Directed towards confidentiality
  - No *read up*
  - No *write down*
- Biba Model
  - Directed towards integrity
  - No *write up*
  - No *read down*

Role-Based Access Control (RBAC)

- Decision *based on the users' role* within an organization
- Example: Oracle DBMS
- A role defines a set of transactions allowed for its members
- Not provided by OS
- Principle of least privilege



	Based on	Models	Rules made by	Configured by	Enforced by
DAC	**identity** *e.g., computer, user, group*	**no standard model** (OS specific impl.) *ACLs and capabilities are two different approaches to DAC*	**owner** *typically restricted by (un)written policies/guidelines*	**owner** *administrator has the power to override*	**OS**
MAC	**security level** *e.g. {unclassified, restricted, secret, top secret}*	Traditional: ·**Bell-LaPadula** ·**Biba** Non-traditional: ·**Windows MIC (label-based)** ·**SE Linux (label-based)** ·**AppArmor (name-based)**	**security officer**	**admin(s)** *labels and rules*	**OS**
RBAC	**role** *e.g., job function*	INCITS 359:2004: ·**Core RBAC** ·**Hierarchical RBAC** ·**Constraint RBAC** Non-standard: ·**AzMan, SELinux** ·**Java EE**	**security officer**	**application or OS admin(s)**	**RBAC System** *transparent such as in SELinux or application-aware such as in RBAC enabled applications*