# PERSONAL - IT-Sicherheit: Begriffe



- **EDITOR'S NOTE**: Dieses Dokument erhält nur übersichtliche Definitionen für manche fachchinesische Begriffe. Insbesondere werden hier Algorithmen und Übungsschemas **nicht** behandelt. Für die genaueren Funktionsweisen oder weitere Definitionen empfehle ich stattdessen die Vorlesungsfolien und Tutorblätter.

## Grundlegende Begriffe

- **Security**: Schutz eines Systems vor äußeren Angriffen
- **Safety**: Schutz eines Systems von Innen (z.B. Hardware- oder Programmierfehler)
- **Asset**: zu schützendes Gut (z.B. Datenbank)
- **Schwachstelle (vulnerability)**: Problem, welches das Umgehen von Sicherheitskontrollen eines Systems erlaubt (z.B. Erlaubnis, ein schwaches Passwort zu benutzen)
- **Bedrohung (threat)**: Potential, die Sicherheit eines Assets zu beeinträchtigen (z.B. Speichern von Passwörtern als Klartext)
- **Angriff, Angriffsvektor (attack, attack vector)**: Angriffsweg, der eine (oder mehrere) Schwachstelle(n) ausnutzt, um die Sicherheit eines Assets bzw. Systems zu gefährden (z.B. DDoS, XSS...)

### Grundlegende Angriffsklassen

- **Sniffing**: Belauschen des Netzwerkverkehrs
- **Spoofing**: tue so, als ob du eine andere Entität wärst
- **Buffer Overflow**: Schreibe- bzw. Lesezugriff außerhalb der Grenzen eines Buffers (Puffers)
- **Code-Injection**: nicht validierte Eingabedaten werden von dem Interpreter als Code ausgeführt
  - **SQL-Injection**: Einfügen von ungefiltertem, bösartigem SQL-Code in einem Eingabefeld, um vertrauliche Datenbankeinträge zu lesen bzw. modifizieren
  - **Cross-Site-Scripting (XSS)**: Einbetten von bösartigem JavaScript-Code auf einem Server, meistens mit `<script>`-Tags, dann Ausführen des Codes innerhalb des Browsers des Opfers
    - **Persistent XSS (stored XSS)**: XSS wird auf der Website bzw. auf dem Server gespeichert und bleibt vorhanden (z.B. XSS in einem Kommentarbereich)
    - **Reflected XSS (non-persistent XSS)**: einmaliges XSS, z.B. via schädlichem, ungeprüftem URL
- **Cross-Site-Request-Forgery (CSRF)**: verleite Browser dazu, unbeabsichtigt eine Anfrage an eine Website zu stellen, auf der der Benutzer authentifiziert ist
- **Man-In-The-Middle**: Angreifer belauscht bzw. modifiziert geheim den Datenaustausch zwischen zwei oder mehreren Kommunikationspartnern
- **(Distributed) Denial of Service (DoS, DDoS)**: gezielter Angriff, um durch Überlast die Verfügbarkeit eines Systems zu beeinträchtigen
- **Social Engineering**: Täuschen von Menschen (z.B. Fake Mails, Fake Telefonate...)
- **Virus**: nicht selbstständiges Schadsoftware, das sich bei Ausführung in andere Dateien selbst weiter kopiert; benötigt Wirtsprogramm
- **Wurm**: eigenständiges Schadsoftware, welches sich in einem Netzwerk verbreitet; benötigt *kein* Wirtsprogramm
- **Trojaner**: Programm, das neben der spezifizierten Funktionalität eine weitere, versteckte, schädliche Funktionalität enthält (z.B. Keylogger in einem Programm mit einer Login-Funktion)
- **Ransomware**: Schadsoftware, welches Dateien auf dem System des Opfers verschlüsselt
- **Phishing**: Nachbilden einer Website und Verleitung der Benutzer zu dieser Fake-Website, in der Hoffnung, das diese dort ihre Nutzerdaten (u.a. Email bzw. Username und insb. Passwort) eingeben
- **Spamming**: mehrfaches, ungewünschtes Senden von Nachrichten (Emails)
- **Rootkits**: Backdoors und Keylogger, die im OS-Kern installiert sind
- **Advanced Persistent Threat (APT)**: aufwändige, langfristige, unerkannte Angriffe
- **Dictionary Attack**: brute-force Passwort-Cracking durch sequentielles Probieren aller möglicher Passwörter anhand einer Liste von (Pass-)wörtern

## Schutzziele

- **CIA**: **C**onfidentiality, **I**ntegrity, **A**vailability
  - **Vertraulichkeit (Confidentiality)**: Schutz vor unautorisierter Informationsgewinnung

    - **Integrität (Integrity)**: Schutz vor unautorisierter und unbemerkter Änderung
    - **Verfügbarkeit (Availability)**: Schutz vor unbefugter Beeinträchtigung der Funktionalität
- **Authentizität**: glaubwürdiger Nachweis der Identität (z.B. Passwort)
- **Verbindlichkeit, Zurechenbarkeit (accountability)**: Nachweis des Eigentums einer Handlung (quasi Aktion X gehört der Person Y) (z.B. digitale Signatur)
- **Privatheit (privacy)**: Fähigkeit einer Person, die Nutzung bestimmter persönlicher Daten zu kontrollieren (z.B. Anonymisierung)

### Schutzmechanismen

- **Same-Origin-Policy**: JavaScript-Code hat nur lesenden Zugriff auf Inhalte auf demselben *Domain*, mit demselben *Protokoll*, auf demselben *Port*
- **CSRF-Tokens**: füge randomisiertes Token zu jedem Request hinzu, welches vom Server validiert werden muss, um CSRF-Angriffe zu vermeiden
- **HttpOnly-Attribut**: verhindert JavaScript, Cookies mit `document.cookie` auszulesen
- **SameSite-Attribut**: regelt Umgang von JavaScript mit Cookies (None, Lax, Strict)
- **Secure-Attribut**: erlaubt Senden und Modifizieren des Cookies nur via HTTPS
- **Input Sanitization**: Ersetzen von gefährlichen Charakteren in der Eingabe

## Kryptographie

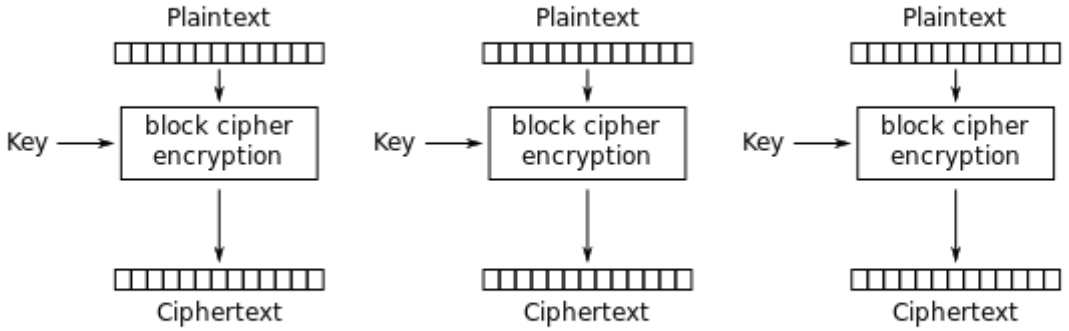
- **Kryptographie**: Methoden zur Ver- und Entschlüsselung
- **Kryptoanalyse**: Wissenschaft von Methoden zur Entschlüsselung
- **symmetrische Verfahren (Secret-Key-Verfahren)**: kryptographische Verfahren, wo die Ver- und Entschlüsselungskeys gleich und geheim sind (z.B. AES)
- **asymmetrische Verfahren (Public-Key-Verfahren)**: kryptographische Verfahren basierend auf Zahlen- und Gruppentheorie, wo jeder Kommunikationspartner je einen öffentlichen Schlüssel zum Verschlüsseln und einen privaten Schlüssel zum Entschlüsseln besitzt (z.B. RSA)
    - **Einwegfunktion**: eine Funktion $f : X \to Y$ mit den Eigenschaften, dass $f(x) = y$ effizient berechenbar ist, $f^{-1}(y) = x$ hingegen nicht
        - **Trapdoor-Einwegfunktion**: Einwegfunktion, die es mit Zusatzinformationen erlaubt, $f^{-1}(y) = x$ effizient zu berechnen
        - **Faktorisierung**: Einwegfunktion $f(p, q) = p \cdot q$
        - **diskreter Logarithmus**: Einwegfunktion $f(x) = g^x \mod p$
- **Kerckhoffs-Prinzip**: die Stärke eines kryptographisches Verfahrens sollte nur auf die Geheimhaltung des *Schlüssels*, nicht des Verfahrens bzw. des Systems basieren
- **Brute Force Angriff**: Erraten eines Schlüssels durch Probieren aller möglichen Kombinationen
- **Elliptic Curve Cryptography (ECC)**: nutze elliptische Kurven statt Modulorechnung für weniger Berechnungs- und Speicheraufwand

## Block- und Stromchiffren

- **Blockchiffre**: symmetrisches Verfahren, Unterteilung eines Klartext in Blöcke fester Länge, Ver- und Entschlüsselung der jeweiligen Blöcke mit dem gleichen Schlüssel (z.B. AES)
  - **Diffusion**: jedes Klartextbit beeinflusst jedes Ciphertextbit
  - **Konfusion**: Verschleiern des Zusammenhangs zwischen Key und Ciphertext
- **Stromchiffre**: symmetrisches Verfahren, Klartext wird mit individueller, pseudorandomisierter Schlüsselfolge der gleichen Länge ver- und entschlüsselt (z.B. ChaCha20, AES-CTR)
  - **Seed**: "Initialisierungswert" eines Pseudo-RNGs; beim selben Seed wird dieselbe Zahl mehrmals generiert

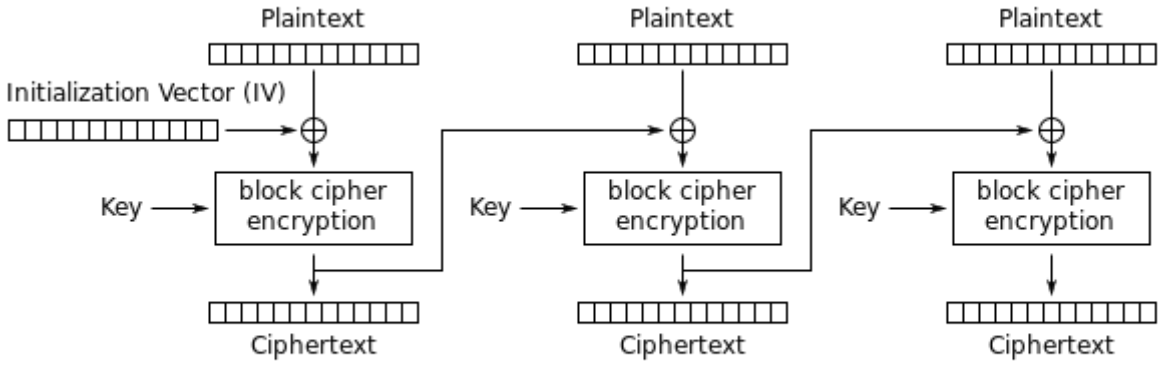
## Betriebsmodi von Blockchiffren

- **Electronic Code Book (ECB)**: jeder Block wird gleich ver- und entschlüsselt

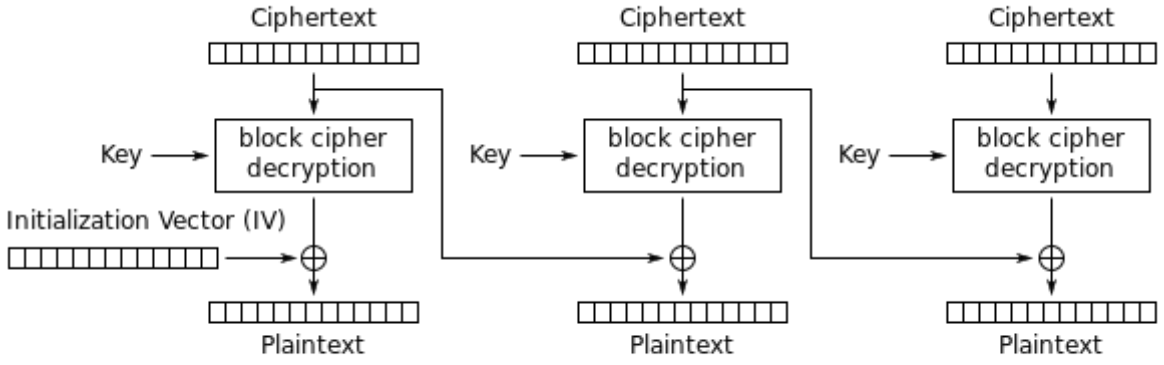



Electronic Codebook (ECB) mode encryption

- **Cipher Block Chaining (CBC)**: Klartextblock wird mit dem vorherigen Ciphertextblock vorher noch ge-XOR-t

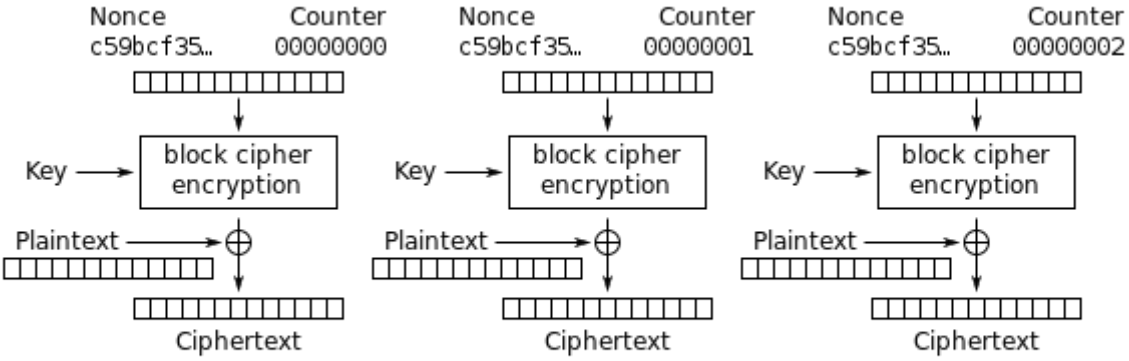



Cipher Block Chaining (CBC) mode encryption

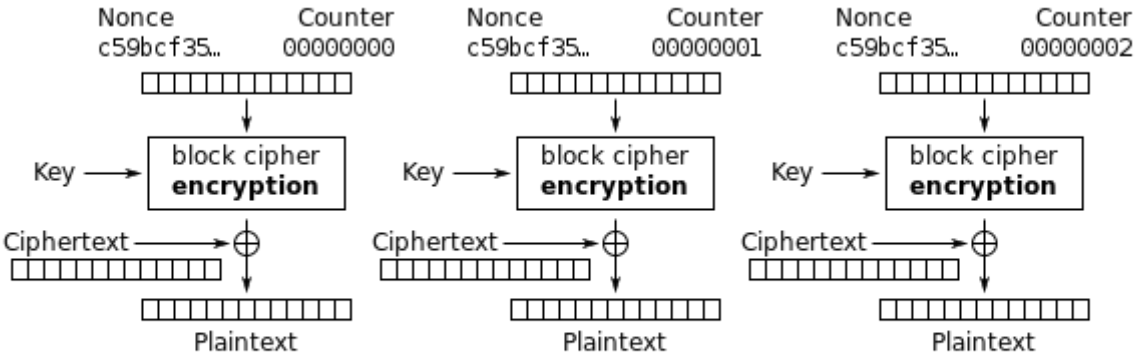



Cipher Block Chaining (CBC) mode decryption

- **Counter Mode (CTR)**: Verschlüsseln eines Initialwertes Nonce + ctr (ctr wird pro Block erhöht), dann XOR mit Klartext




Counter (CTR) mode encryption




Counter (CTR) mode decryption

# Hashfunktionen, MACs

- **kryptographische Hashfunktion**: Funktion, die verwendet werden kann, um Daten beliebiger Größe auf Werte fester Größe (Message Digest) abzubilden, um durch bestimmte Anforderungen (deshalb "kryptographisch") Integrität zu gewährleisten (z.B. MD5, SHA1, SHA2)
  - **Einwegeigenschaft (preimage resistence)**: aus einem gehashten Wert soll es ineffizient sein, die / eine ursprüngliche Nachricht zu bestimmen
  - **Schwache Kollisionsresistenz (second-preimage resistence)**: mit einer gegebenen Nachricht soll es ineffizient sein, eine zweite Nachricht zu finden, die auf denselben Hashwert abbildet
  - **Starke Kollisionsresistenz (collision resistance)**: es soll ineffizient sein, jedwelche Paare von Nachrichten zu finden, die auf denselben Hashwert abbilden
- **Passworthashfunktionen**: Hashfunktionen, die geeignet sind, um Passwörter zu speichern, dadurch, dass der Abgleich langsam und speicher- sowie CPU-intensiv ist
- **Message Authentication Code (MAC)**: Nachweis der Authentizität des Datenursprungs durch Hashen eines Shared Keys $k_S$ zusammen mit der Nachricht $m$, formal $\mathrm{MAC} = H(k_S \mid\mid m)$
  - **Encrypt-Then-MAC**: verschlüssele Klartext, bilde MAC aus Shared Key und Ciphertext und sende Ciphertext zusammen mit MAC
  - **MAC-Then-Encrypt**: bilde MAC aus Klartext, füge es zum Klartext hinzu und verschlüssele die gesamte Nachricht
- **klassischer Kollisionsangriff**: finde zwei beliebige Nachrichten, die denselben Hashwert haben ( $H(m_1) = H(m_2),\ m_1 \neq m_2$)
- **Chosen-Prefix-Angriff**: finde für zwei gegebene, unterschiedliche Präfixe zwei Anhänge, so dass die zwei vollständigen Nachrichten denselben Hashwert haben ($H(p_1 \mid\mid m_1) = H(p_2 \mid\mid m_2),\ p_1 \neq p_2,\ m_1 \neq m_2$,)
- **Length Extension Attack**: möglicher Angriff auf Merkle-Damgard-Hashfunktionen, wo ein Angriffer die ursprüngliche Nachricht und den MAC abfangen kann und aufgrund der Konstruktion von MD-Hashes neue Daten am Ende hinzufügen kann, einen neuen (validen) Hash berechnet und die modifizierten Daten an den Empfänger weiterschickt, der nicht bemerken kann, dass die ursprüngliche Nachricht modifziert wurde, formal $\mathrm{data}, h$ gegeben, $\mathrm{data'} = \mathrm{data} \mid\mid \mathrm{fake}$ mit $\mathrm{MAC} = H(k_S \mid\mid \mathrm{data'}) = H(k_S \mid\mid \mathrm{data} \mid\mid \mathrm{fake}) = H(h \mid\mid \mathrm{fake})$
- **Hash-based Message Authentication Code (HMAC)**: geschachteltes Hashing (berechnet internen, dann äußeren Hash), so dass Length Extension Angriffe nicht mehr möglich sind
- **AEAD (Authenticated Encryption with Associated Data)**: zusätzliches Senden von (Header-)Daten zusammen mit Ciphertext zum Nachweis der Authentizität des Ciphertexts und asoziierter Daten (z.B. AES-GCM)
- **Elektronische / Digitale Signatur**: Nachweis der nicht-abstreitbaren Urheberschaft einer Nachricht durch "Signieren" des Nachrichtenhashes mit einem Public-Key-Verfahren (z.B. RSA) oder einem

dedizierten Signaturverfahren (z.B. DSA); Verschlüsseln mit private Key, Entschlüsseln / Validieren mit public Key

## Schlüsselmanagement

- **Entropie**: Maß der Zufälligkeit (je höher, desto besser)
- **True Random Number Generator (TRNG)**: Zufallszahlgenerator basierend auf physikalischen Phänomenen (z.B. Mausbewegung)
- **Pseudorandom Number Generator (PRNG)**: quasi-Zufallszahlgenerator basierend auf deterministischem Algorithmus mit einem *geheimen* Anfangswert (Seed) (z.B. XORShift32)
- **Cryptographically Secure Pseudorandom Number Generator (CSPRNG)**: Pseudozufallszahlgenerator, der zusätzliche kryptographische Anforderungen erfüllt (nicht vorhersagbar, kein Hinweis auf vorhergehende Zufallszahlen, statistisch gleichviele 0en und 1en) (z.B. Hardware-Schieberegister, AES-CTR)
- **Key Distribution Center (KDC, AuC?)**: zentraler Schlüsselverteilungsserver, wo jeder der zwei Kommunikationspartner $A, B$ jeweils einen pre-shared Key $k_A, k_B$ und einen neuen gemeinsamen, frischen Shared Key für die Kommunikation $k_{AB}$ beantragen können
- **Key Encapsulation Mechanism (KEM)**: symmetrischer Schlüssel $k_{AB}$ wird asymmetrisch ver- und entschlüsselt (z.B. RSA-KEM)
- **Perfect Forward Secrecy (PFS)**: wird ein Schlüssel unsicher, dürfen andere, zu früheren Zeitpunkten genutzte Schlüssel nicht auch unsicher werden; ein neuer Schlüssel darf nicht von einem alten Schlüssel abhängen

## Digitale Identität, Authentisierung

- **Authentisierung / Authentifikation**: Prüfung der Authentizität auf Basis einer eindeutigen Identität
- **Autorisierung**: Vergabe von Zugriffsrechten
- **Pseudonymisierung**: Verarbeitung personenbezogener Daten so, dass man diese ohne zusätzlichen Informationen alleine nicht mehr der Person zuordnen kann (z.B. Nickname, LRZ-Kennung)
- **Anonymisierung**: Verarbeitung personenbezogener Daten so, dass man diese auf keine Art und Weise der Person (bzw. nur mit sehr viel Aufwand) zuordnen kann (z.B. Ersetzen von Namen durch Zufallszahlen)
- **Challenge-Response-Verfahren (CR)**: Nachweis der Identität durch ein- bzw. wechselseitiges Senden einer Challenge von einer Seite (z.B. RAND) und Senden einer *eindeutigen* Antwort von der anderen Seite, die nur diese Person lösen kann (bzw. können sollte)

### Authentisierungsklassen

- **Wissen**: Authentisierungsinformationen, die man selber im Gedächtnis hat (z.B. Passwort, Sicherheitsfrage)

- **Besitz**: physische Objekte bzw. Geräte, die zur Authentisierung dienen (z.B. RSA-Token, SIM Card)
  - **One Time Password (OTP)**: (meist) zeitbasierter Code, der von einem Authentisierungsgerät generiert wird und mit dem Server synchronisiert ist
- **Biometrie**: eigene, körperbezogene Authentisierungsmethoden (z.B. Fingerabdruck, Gesicht)
  - **physiologische (statische) Merkmale**: langlebige Merkmale, die sich (normalerweise) im Laufe des Lebens nur gering ändern können (z.B. Fingerabdruck, Iris...)
  - **Verhaltensmerkmale (dynamisch)**: Merkmale, die sich ändern können (z.B. Stimme, Tippverhalten...)
  - **Enrollment**: ein- und erstmalige digitale Registrierung eines biometrischen Merkmals des Benutzers
  - **False Negative**: berechtigter Nutzer wird abgelehnt
    - **False Rejection Rate (FRR)**: Wahrscheinlichkeitsmaß für False Negatives
  - **False Positive**: unberechtigter Nutzer wird akzeptiert
    - **False Acceptance Rate (FAR)**: Wahrscheinlichkeitsmaß für False Positives
  - **Equal Error Rate (ERR)**: FAR und FRR sind gleich; Maß für die Güte eines biometrischen Systems
- **Multi-Faktor Authentisierung (2FA, MFA)**: Kombinierung verschiedener Authentisierungsfaktoren (z.B. Passwort + Authenticator Code)

## Public Key Infrastructure, Single-Sign-On

- **X.509-Zertifikat**: bindet eine Identität mithilfe einer digitalen Signatur an einen öffentlichen Schlüssel
- **Public Key Infrastructure (PKI)**: eine Menge von Rollen, Richtlinien, Hardware, Software und Verfahren, die zum Erstellen, Verwalten, Verteilen, Verwenden, Speichern und Widerrufen digitaler Zertifikate sowie zum Verwalten der Public-Key-Verschlüsselung erforderlich sind
  - **Registration Authority (RA)**: verantwortlich für die Annahme von Anfragen nach digitalen Zertifikaten
  - **Certificate Authority (CA)**: signiert und stellt digitale Zertifikate aus
  - **Validierungsstelle (VA)**: validiert Zertifikate
  - **Verzeichnisdienst (DIR)**: Verzeichnis mit ausgestellten Zertifikaten
  - **Personal Security Environment (PSE)**: sichert Speicherung des privaten Schlüssels
- **Online Certificate Status Protocol (OCSP)**: ein Internetprotokoll zum Ermitteln des Sperrstatus eines digitalen X.509-Zertifikats, wo der Server regelmäßig die Gültigkeit eines Zertifikats von dem OCSP-Responder der CA erfragt und diese dem Client mitschickt (valid bzw. invalid)
- **Policy Decision Point (PDP)**: Ausstellung einer Bescheinigung
- **Policy Enforcement Point (PEP)**: Prüfen der Gültigkeit der Bescheinigung (des Tickets)

- **Single-Sign-On**: Authentifizierungsschema, das es einem Benutzer ermöglicht, sich mit einer einzigen ID bei mehreren verwandten, aber unabhängigen Softwaresystemen anzumelden
- **Kerberos**: ticket-basiertes Single-Sign-On-Authentisierungsprotokoll für Kommunikation eines Benutzers mit verschiedenen Services
  - **Ticket ($T^{A,S}$)**: notwendig für SSO-Zugriff von $A$ auf Service $S$, wird vom Ticket Granting Service (TGS) ausgestellt
  - **Authenticator ($\mathrm{Authent}^{A}$)**: Nachweis des berechtigten Ticket-Besitzes von $A$

## Netzwerksicherheit

- **Sicherer Kanal (secure channel)**: ein abhör- und manipulationssicherer Datentransferkanal
- **Transport Layer Security (TLS)**: kryptografisches Protokoll zur Gewährleistung der Kommunikationssicherheit über ein Computernetzwerk
  - **1-Round-Trip-Time (1-RTT)**: der Server sendet mit seiner ersten Nachricht Anwendungsdaten an den Client
  - **0-Round-Trip-Time (0-RTT)**: der Client sendet mit seiner ersten Nachricht Anwendungsdaten an den Server, verschlüsselt mit einem gecachten Schlüssel aus einem früheren Handshake
- **Virtual Private Network (VPN)**: Netzinfrastruktur, bei der Komponenten eines privaten Netzes über ein öffentliches Netz kommunizieren und via eines "Tunnels" die Illusion besitzen, dieses Netz zur alleinigen Verfügung zu haben
- **Firewall**: Netzwerksicherheitssystem, das den ein- und ausgehenden Netzwerkverkehr anhand vorgegebener Sicherheitsregeln überwacht und steuert
  - **Header**: Teil eines Datenpakets mit notwendigen Metadaten für Kommunikation
  - **Payload**: Teil eines Datenpakets, welches die eigentlichen geschickten Daten enthält
- **Paketfilter**: Form einer Firewall, prüft nur Header-Daten und antwortet mit `accept` oder `drop`
- **Deep Packet Inspection (DPI)**: Form einer Firewall, prüft sowohl Header-Daten, als auch Payload
- **Application Layer Gateway (ALG), Proxy-Firewall**: Form einer Firewall, baut neue Verbindung auf und hat Zugriff auf kompletten Verbindungszustand (i.e. kann alle Daten lesen und verändern)

## Autorisierung, Rechtemanagement

- **Autorisierung**: Vegrabe und Kontrolle von Zugriffsberechtigungen
- **Subjekt**: zugreifende Instanz (z.B. Benutzer)
- **Objekt**: genutzte Ressource (z.B. Datei)
- **Zugriffsrecht**: Berechtigung(en) eines Subjekts bezüglich eines Objekts (z.B. read, write, execute)
- **Kontext / Attribut**: Bedingung des Zugriffs (z.B. Tageszeit, Alter)
- **Prinzip der minimalen Rechte (need-to-know principle)**: nur die Rechte vergeben, die zur Aufgabenerfüllung erforderlich sind

- **Prinzip der Complete Mediation (Zero Trust)**: *jeder* Zugriff auf eine Ressource wird kontrolliert
- **Discretionary Access Control (DAC)**: Zugriffsrechte werden von Benutzer bestimmt
- **Mandatory Access Control (MAC)**: Zugriffsrechte werden vom System bestimmt
- **Role-Based Access Control (RBAC)**: Zugriffsrechte sind aufgabenorientiert
  - **statische Aufgabentrennung**: Benutzer darf nur gleichzeitig Mitglied einer Rolle sein
  - **dynamische Aufgabentrennung**: Benutzer hat nur gleichzeitig die Rechte einer Rolle

## Systemsicherheit, Secure Programming

- **System Call (Syscall)**: "Funktionsaufruf", Computerprogramm fordert einen Dienst vom Betriebssystem an
- **Kernel Mode**: privilegierter Betriebsmodus, u.A. Zugriff auf privilegierte Instruktionen
- **User Mode**: eingeschränkter Betriebsmodus für Nutzeranwendungen, kein Zugriff auf privilegierte Instruktionen
- **Trusted Computing Base (TCB)**: die Gesamtheit aller Hardware- und/oder Softwarekomponenten, die die Sicherheitskontrolle in einem System umsetzen
- **Code-Reuse Angriffe**: Wiederverwendung von bereits im Programm geladenem Maschinencode, um somit das no-execute-bit zu umgehen
  - **Return-to-libc**: Code-Reuse Angriff, wo die Rücksprungadresse auf eine Funktion der Standardbibliothek libc (die bereits geladen ist) gesetzt wird
- **Stack Canary**: Zufallszahl, die auf dem Stack vor der Return-Adresse gespeichert wird und vor Funktionsterminierung vom Programm auf Änderung geprüft wird
- **Data Execution Prevention (DEP)**: Markieren von Seiten als nicht ausführbar (no execute)
- **Adress Space Layout Randomization (ASLR)**: Stack-, Heap- und Bibliotheksadressen werden für jeden Programmstart randomisiert (aber nicht Text-Segment!)
  - **Position Independent Executable (PIE)**: Erweiterung von ASLR, Randomisierung des Text-Segments

---

Summary by Flavius Schmidt, ge83pux, 2024.
https://home.in.tum.de/~scfl/
Images from Wikimedia.