Introduction

Threats

- Ransomware & Malware
- Social Engineering
- Denial of Service (DoS)
- Supply Chain Attacks
- Data Threats

Threat Actors

- **State-Nexus**: Espionage, disruption, national funding.
- **Cybercrime / Hackers-for-Hire**: Financial gain.
- **Private Sector Offensive**: Corporate profit.
- **Hacktivists**: Ideology, political or social

Goals of Cybersecurity

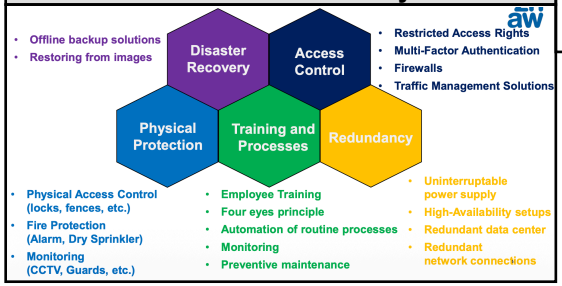
- The CIA Triad (Core):
  - **Confidentiality**: Prevent unauthorized reading/access.
  - **Integrity**: Prevent unauthorized modification.
  - **Availability**: Prevent system/data outages.
- **Extended Goals**: Authenticity, Non-repudiation, Freshness.
- **Business Goal:** Business Continuity (keeping operations running despite attacks/disasters).

Legal Aspects

- **Criminal Law (Strafrecht):** Punishes breaches of trade, official, and professional secrecy.
- **Cybercrime Articles:** Outlaws data theft, unauthorized access, data damage, and computer fraud.
- **The "Ethical Hacking" Rule:** Unauthorized hacking is *always* illegal. Pentesting/security research strictly requires a formal contract.
- **Data Protection (nFADP/revDSG):** Unlike GDPR, individuals (not just companies) can be held personally liable for breaches.

Software Security

Business Availability



The 7 Kingdoms + 1 Kingdom

1. **Input Validation and Represent** (fn!)
2. **API Abuse** (bypass, misuse)
3. **Security Features** (bad crypt, bad auth)
4. **Time and State** (race conditions, t-out)
5. **Errors** (leaking logs, unhandled)
6. **Code Quality** (complexity, unused)
7. **Encapsulation** (unclear bounds)
- **Environment** (not app code, depl. stack)

Security Testing

1. SAST (Static Appl. Security Testing)
   - IDE/ CI - AST - analyze - rep
   - sign: pattern match - taint: track flow
   - lint etc NOT security, just defects
2. DAST: run code, not just text analysis
   - dicovery - probing - analyze - rep
   - for uuids in.. for cookies..
   - Fuzzig:
     - Standard: random input
     - Mutation: Partial start mut via seed
     - Greybox: Underlying sys knowledge
3. Human: bored/overwhelmed/expensive
4. Pentesting / Red Teaming
   - Metasploit, Nmap, Burp, ZAP
   - Red teams: simulate, blue teams: defend

Measurement

Analysis	Rep corr	Rep incorr
Reported a defect	**(TP):** Correctly reported a defect	**(FP):** Incorrect report ("Type I error")
Did not report a defect	**(TN):** Correctly did not report a 'secure' issue	**(FN):** Incorrect because it failed to report ("Type II error")

Weakness

- CWE (Common Weakness Enumeration)
- CVE (Common Vulnerabilities and Exposures)

Encryption

Work Factor

Average n attempts until correct key is found
Influenced by:
- Algorithm
- Key length
- Key "randomness"

$> 2^{128}$ is considered secure for the foreseeable future

Perfect Secrecy

Ciphertext does not reveal any information about the plaintext Requires:
- Key is as long as the message
- Key is truly random
- Key is used only once (**one-time pad**)
- Key must be exchanged

Hash Functions

Insecure:
- 2^{64}: MD5 128-bit hash
- 2^{80}: SHA-1 160-bit hash

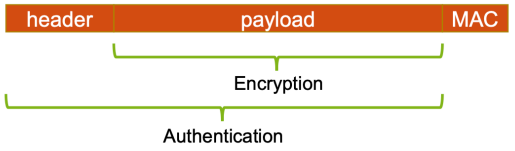
Secure:
- $2^{112-256}$: SHA-2 224-512-bit hash
- $2^{128-256}$: SHA-3 224-512-bit hash

Secret Key Cryptography

= Secure channel vor key exchange
- block-cipher vs stream-cipher (blocks vs single bit XOR)
- $\frac{2^n+1}{2} \approx 2^n$ approx for key length > 218
- IV: public random value preventing identical ciphertext for identical plaintexts

AES: Authenticated Encryption

- Encryption: Confidentiality
- MAC: Authenticity and Integrity



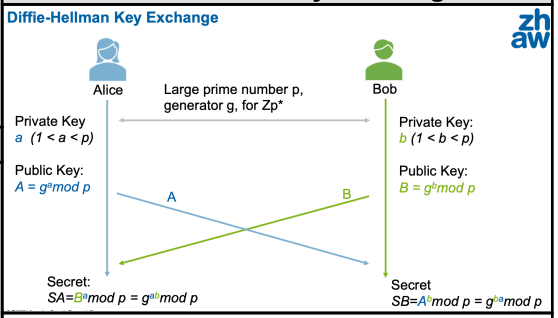
- **ECB**: No IV, no MAC
  - **Do not use.** no real data protection.
- **CBC**: IV only, no MAC
  - **Do not use.** no data modification prot.
- **CTR**: IV only, no MAC
  - **Do not use.** no data modification prot.
- **CCM**: IV + MAC → slower than GCM
- **GCM**: IV + MAC → **current standard**

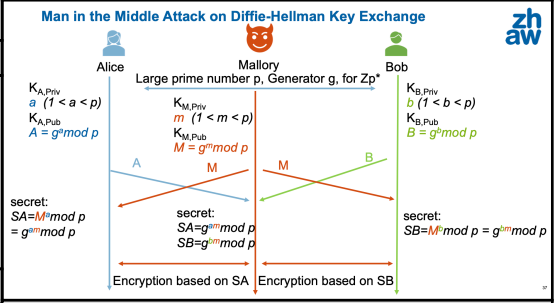
Public Key Cryptography

Work Factor of Public Key Algorithms
- **Classical algorithms:**
  - key of 4096 bits $\rightarrow 128$ bit
- **Elliptic Curve algorithms:**
  - key of 256 bits $\rightarrow 128$ bit

- Longer keys require more computing power
- only use public key crypto when secret key crypto won't work (e.g., key negotiation)
- Quantum computers could theoretically break Public Key Cryptography

Diffie Hellman Key Exchange





RSA

Usage
- Signatures of Key Exchange Messages
- Encryption (in theory, not used)

Mathematics
- Elliptic curves not possible => resource intensive

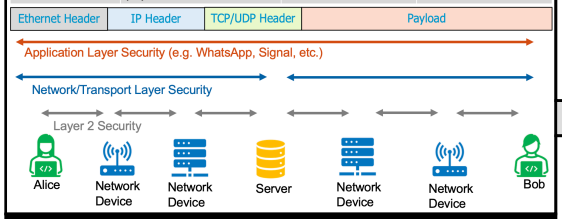
Terminology
- e: public key exponent, usually 65537
- d: private key exponent
- n: Modulus, large prime number. The length of n corresponds to the key length
- Public key (e, n)
- Private key (d, n)

Encryption: $c = p^e \bmod n$
Decryption: $p = c^d \bmod n$

Signature generation: $s = p^d \bmod n$
Signature verification: $p = s^e \bmod n$

TCP/IP Stack

Application Layer	Prepare data for sending	FTP, HTTP	SFTP, HTTPS, SSH
Transport Layer	End to end communication	TCP / UDP	TLS, QUIC
Network Layer	Route packets between networks	IP / ICMP	IPSec
Data Link Layer	Fragment packets for physical network	Ethernet	WPA3, MacSec

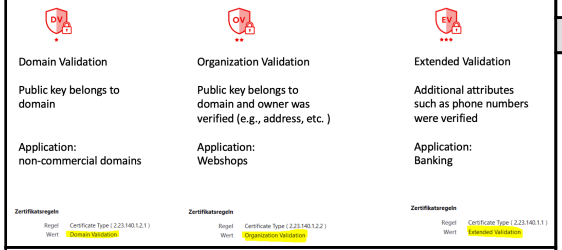


Secure Transport Layer Protocols

Security Goals

- **Confidentiality**: Keeping secrets (encryp)
- **Integrity**: No tampering (MAC, signat.)
- **Authentication**: Proving identity (certificates, signatures, MAC)

Certificates



- Root certificates are anchor of trust
- Preinstalled in browsers and OS

Verification of Certificates

- Issuer and chain of trust
- Validity Period
- Revocation (see next section CRL)
- Domain name (subject)

CRL (Certificate Revocation List)

- List of revoked certificates
- Downloaded by browsers/clients centrally
- Compressed with bloom filters
- Browser checks vs local compressed CRL

Root certificates cannot be revoked via CRL, only removed from trust store.

Optaining Domain Certificates

Agent

1. Create key pair “S”
2. Create CSR: Certificate Signing Req.
   - Add domain and pub key “S”
   - Sign with “S”
   - Sign with “A” (optional author. wrap)

Lets Encrypt (CA)

3. CA verifies domain ownership: Challenge
4. CA verifies signatures
5. CA creates/issues certificate

TLS (Transport Layer Security)

- Successor of SSL (Secure Sockets Layer)
- Provides confidentiality, integrity and authentication on **transport layer**
- Used in HTTPS, FTPS, SMTPS, etc.
- Uses certificates for authentication on
- Uses symmetric encryption for data transfer, asymmetric encryption for key exchange
- Strong block ciphers (AES)
- Authenticated encryption (GCM, CCM)
- Diffie-Hellman Agreement
- Authentication through signatures

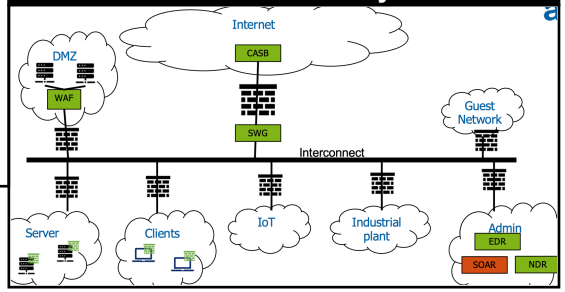
DTLS (Datagram Transp. Layer Security)

- Variant of TLS for datagram protocols (e.g. UDP)
- Similar security guarantees as TLS
- Used in applications like VoIP, online gaming, etc.
- **TLS handshake** in UDP: via reliability functions: timeout/retransmission/sequence numbers

QUIC (Quick UDP Internet Connections)

- Developed by Google, standardized by IETF
- Protocol encapsulated in **UDP**
- Used in HTTP/3, designed for the web
- **TLS 1.3** for key negotiation
- Also on **transport** layer

Network Security



Buzzwords

- Firewall: Filter traffic
- Segmentation: Divide network into segments (segments have more internal trust)
- DMZ (Demilitarized Zone): Network segment between internal and external network)

Firewalls

- Policy: Who can access what
- Rules: Implemented by Admim

Limitations

- Cannot protect against insider threats
- Application specific attacks

Next Generation Firewalls NGFW

- Deep packet inspection
- Intrusion prevention
- Application awareness
- Anti-malware
- Sandboxing

Micro Segmentation

- Divide network into smaller segments
- Host local firewall
- Tradeoff: Securit vs Manageability

Zero Trust

- Never trust, always verify
- Least privilege access enforcement
- Security monitoring

Solutions

- Not easy to implement, follow existing models, such as CISCA Zero Trust Maturity Model

Zero trust often misused as a marketing buzzword.

Threats

- Single point of failure
- Stolen or misused credentials
- Lack of visibility (due to encryption)
- Monitoring access contains valuable information for attackers
- Misuse of agents with more rights than users

Attack Detection

- Endpoint Detection and Response (EDR): Detect and respond to threats on endpoints

Protection

- Local firewall
- Antivirus

Monitoring

- Anomaly detection
- Vulnerability scanning
- Integrity checks

Investigation and Response

- Isolation of devices
- Logout all sessions
- Rollback to known good state

Protecting Applications with WAF

- Web Application Firewall (WAF): Protect web applications by filtering and monitoring HTTP traffic

Protection

- Block malicious traffic
- Prevent common attacks (e.g. SQL injection, XSS)

Monitoring

- Analyze traffic patterns
- Detect anomalies

Secure Web Gateways SWG

- URL filtering
- Data Leak Prevention (DLP)
- TLS inspection

Cloud Access Security Brokers CASB

- Shadow IT discovery
- Cloud usage control
- Data leak prevention
- Anomaly detection

Implementation

- API Scanning
- Forward Proxy, similar to SWG
- Reverse Proxy, redirect traffic through CASB

Network Detection and Response NDR

- Changes in firewall configuration
- Isolating an infected device
- Speed up reacting to incidents

Drawbacks

- Needs interfaces to other network devices to respond
- Wrong classification lead to unnecessary downtime

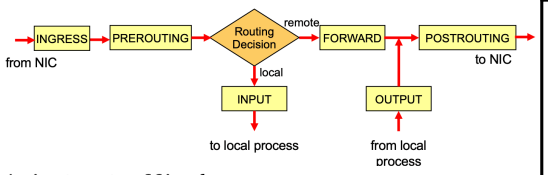
Sec. Info. and Event Management SIEM

- Correlation of events of different sources to detect attacks
- Considolated single dashboard view
- Report generation for compliance

Sec. Orchestr, Autom. and Resp. SOAR

- Extend SIEM with auto resp capabilities

Packet-Filtering with netfilter/nftables




```
chain tcp-traffic {
  type filter hook input priority 0; policy drop;
  tcp dport { https, http } jump http-traffic
}
chain http-traffic { # non-base-chain
  type filter; policy drop;
  count accept
}
```

Policies

- **accept**: allow traffic (**default**)
- **drop**: silently discard traffic
- **reject**: discard traffic and send err msg
- **jump**: send traffic to another chain

(Distributed) Denial of Service (D)DoS

- Overwhelm target with traffic
- Types of DDoS attacks:
- Volumetric: Consume bandwidth (e.g. **DNS ampplification**)
- Protocol: Exploit protocol weaknesses (e.g. **SYN flood**)
- Application Layer: Target specific applications (e.g. **HTTP flood**)

Mitigation

- Rate limiting
- Traffic filtering
- Scrubbing centers

Secure Lower Layer Protocols

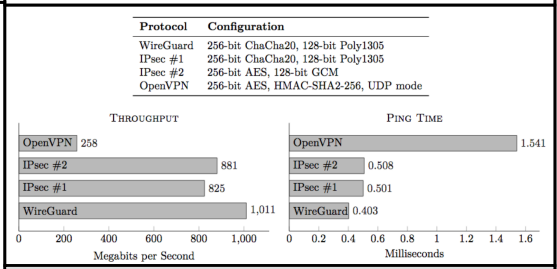
VPN Use-Cases

- Change Geo-location of IP address
- Hide from ISP
- Connect to corp. network from outside
- Con. two nets. securely over the internet
- Remote support for IT-Services

VPN Protocols

- IPSec: Operates on **network layer**, can be used in tunnel mode
- OpenVPN: Operates on **transport layer** (UDP) in user space, can be configured
- Proprietary protocols: Cisco
- WireGuard: Similar to OpenVPN, but in kernel space. Focus on speed rather than configurability

Protocol	Configuration
WireGuard	256-bit ChaCha20, 128-bit Poly1305
IPsec #1	256-bit ChaCha20, 128-bit Poly1305
IPsec #2	256-bit AES, 128-bit GCM
OpenVPN	256-bit AES, HMAC-SHA2-256, UDP mode



Wireless Security

- Packet sniffing: Anyone can capture wireless traffic
- Unauthorized access: Everyone in range can use the network

Encryption and Authentication at Layer 2 Required

Wired Equivalent Privacy (WEP)

- Even with high bit-key WEP is insecure
- Because: Only 24 bit IV → we can attack Keystream instead of key
- CRC is noth enought for authentication (not a MAC)

Wi-Fi Protected Access (WPA)

- WPA: Still uses RC4, but with TKIP (Temporal Key Integrity Protocol)
- WPA2: Uses AES with CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol)
- WPA3: Uses Simultaneous Authentication of Equals (SAE)

User Authentication

Access Control Components

- **Identification**: Claiming an identity (e.g. username)
- **Authentication**: Proving the claimed identity (e.g. password)
- **Authorization**: Granting access to resources based on identity and permissions
- **Audit Trails**: Logging access

Passwords

- Phishing
- Intelligent Guessing when not random
- Multiple Use
- Darknet reselling of leaks/hacks
- Humans can't remember good passwords so they tend to make weak choices

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	57 minutes	2 hours	4 hours
6	Instantly	46 minutes	2 days	6 days	2 weeks
7	Instantly	20 hours	4 months	1 year	2 years
8	Instantly	3 weeks	15 years	62 years	164 years
9	2 hours	2 years	791 years	3k years	11k years
10	1 day	40 years	41k years	238k years	803k years
11	1 weeks	1k years	2m years	14m years	56m years
12	3 months	27k years	111m years	917m years	3bn years
13	3 years	705k years	5bn years	56bn years	275bn years
14	28 years	18m years	300bn years	3tn years	19tn years
15	284 years	477m years	15tn years	218tn years	1qd years
16	2k years	12bn years	812tn years	13qd years	94qd years
17	28k years	322bn years	42qd years	840qd years	6qn years
18	284k years	8tn years	2qn years	52qn years	463qn years

Storing Passwords

- **Never** store passwords in plaintext
- Use a strong hash function (e.g. bcrypt, scrypt, Argon2
- Use a unique salt for each password
- Consider using a pepper (a secret value stored separately from the database)
- Rounds (key stretching)

Multi-Factor Authentication (MFA)

- Something you know (password)
- Something you have (security token, smartphone)
- Something you are (biometrics)

Issue with MFA

- Does not prevent inline phishing attacks.
- MFA Fatique: Inconvenience for users

Passwordless Authentication

- FIDO2/WebAuthn: Uses public key cryptography for authentication, no passwords involved
- Windows Hello/Apple Face ID: Biometric authentication

Federated User Authentication

SSO (Single Sign-On)

- User can access multiple applications with one set of credentials
- Reduces password fatigue and improves user experience
- Centralized authentication and authorization

OAuth 2.0

- Authorization framework that allows third-party applications to access user resources without sharing credentials
- Roles: Resource Owner, Client, Authorization Server, Resource Server
- Flows: Authorization Code, Implicit, Resource Owner Password Credentials, Client Credentials

OpenID Connect (OIDC)

- Authentication layer on top of OAuth 2.0
- Provides user identity information in the form of ID tokens (JWT)
- Allows clients to verify the identity of the user and obtain basic profile information

SAML (Security Assertion Markup Lang.)

- XML-based framework for exchanging authentication and authorization data between parties
- Used for SSO in enterprise environments
- Roles: Identity Provider (IdP), Service Provider (SP)

Shibboleth and Kerberos

- Shibboleth: Open-source SSO solution based on SAML, used in academic institutions
- Kerberos: Network authentication protocol that uses tickets to allow nodes to prove their identity securely, often used in Windows environments

Self Sovereign Identity (SSI)

Principles

1. **Existence**: users have the right to a digital identity.
2. **Control**: users have complete control over their identity.
3. **Access**: users have access to their identity at all times
4. **Transparency**: systems and algorithms must be open and traceable.
5. **Persistence**: identities should exist in the long term without being tied to specific providers.
6. **Portability**: identities must be usable between different platforms.
7. **Interoperability**: systems should be compatible with each other.
8. **Consent**: users must explicitly consent to the use of their data.
9. **Minimization**: only the minimum amount of data should be shared.
10. **Protection**: identities must be secure and protected from unauthorized access.

Comparison and Trade-offs

- direct: easy, but no scalability
- federated/SSO: scales, but tracking and SPoF
- SSI: user control, but complex + user responsi.

Authorization

Methods

- **DAC (Discretionary Access Control)**: The owner decides who has access. Flexible but vulnerable to errors/malware (e.g., Confused Deputy). Implemented via ACLs (Access Control Lists) or Capabilities.
  - Personal computerrs, file sharing, small nets
- **MAC (Mandatory Access Control)**: The system/admin enforces access based on strict security labels/clearance. Users cannot override it.
  - Military, OS-level security (e.g. SELinux)
- **RBAC (Role-Based Access Control)**: Access is tied to the user's job function (role) rather than their individual identity. Reduces administration overhead.
  - Enterprise environments, cloud services, complex web applications
- **ABAC (Attribute-Based Access Control)**: Extends RBAC by using dynamic context (e.g., time of day, location, age) to make access decisions.
  - Enterprise environments, cloud services

	Based on	Rules made by	Configured by	Enforced by
DAC	**identity** e.g., computer, user, group	**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}	**security officer**	**admin(s)** labels and rules	**OS**
RBAC	**role** e.g., job function	**business/security officer**	**application or OS admin(s)**	**RBAC System** transparent such as in SELinux or application-aware such as in RBAC enabled applications

Linux Standard Permissions (DAC)

- **Traditional Permissions (UGO)**: Single Owner (**User**), a single **Group**, and **Others**.
- Uses three basic rights: Read (r/4), Write (w/2), and Execute (x/1).
- Example: `chmod 755 script.sh` gives the owner read, write, and execute permissions, and gives the group and others read and execute permissions.

owner, **group**, **others**

–-rwx------	is equal to	700
–-rwxr-x---	is equal to	750
–-rwxr-xr-x	is equal to	755
–-rwxrwxrwx	is equal to	777

Linux ACLs (DAC)

- **Access Control Lists (ACLs)**: More fine-grained permissions than traditional Unix permissions
- Can specify permissions for individual users and groups
- Example: `setfacl -m u:username:rwx file.txt` to give user `username` read, write, and execute permissions on `file.txt`

Seeing ACL in linux:

- `getfacl file.txt` to view ACLs for `file.txt`
- `ls -l` will show a `+` at the end of the permissions if ACLs are set for a file

Access Control List	Capabilities
+ Subjects (users, groups) are often associated with clearly identifiable entities and are therefore intuitive to use. + Determining who is allowed to do what with a specific resource is efficient. + Revocation is straight forward.[1]	+ Authorization is efficient (check token). + **No designation without authority.**
- Determining what a specific subject is allowed to do is inefficient. - Checking of ACLs can be complex. - **Confused deputy problem.**	- Tokens can't be revoked. - Determining which subjects can access a specific object is difficult. - Auditing is difficult: No token-to-principal binding.
Delegation: By the owner and/or administrator only.	Delegation: Anyone having the token can pass it on.

Securiity and AI

AI and Cybersecurity

- Novel: Still learning about LLMs
- Malicious use: Attackers can use LLMs too
- Training data: LLMs are trained on public data, even malicious data
- Complex: LLMs have a lack of transparency
- Non deterministic: LLMs output can vary

Trustworthy / Responsible AI Principles

- Safety
- Fairness
- Sustainability
- Transparency
- Accuracy
- Privacy

OWASP Top 10 AI Risks

- **Prompt Injection**
- Sensitive Information Disclosure
- Supply Chain Vulnerabilities
- **Data Mand Model Poisoning**
- Imporper Output Handling
- Exessive Agency
- System Prompt Leakage
- Vector amd Embedding Weaknesses
- Misinformation
- Unbounded Consumption

Privacy and Data Protection

Privacy vs Security vs Data Protection

- Privacy: Indetified or identifiable natural person
- Data Prot.: legal framework enforcing privacy
- Security: Company data

GDPR

- EU citizens
- Anonyumous data is exempt
- Invidiuals have notice and choice about their data
- Privacy by design requirement
- Regular security testing required

GDPR vs nFADP

- Fines: Up to 20 million or 4% of global turnover vs up to 250k CHF
- Accountability: The company vs the responsible individual
- Breach notification: Within hours vs "as soon as possible"
- Consent: Explicit vs only explicit for high-risk profiling

Privacy by Design

- Minimize data
- Automate data deletion
- Access controls
- Notice and Choice

Fundamentals

- **Anonymization**
- **Pseudonymization:** Replacing identifiers with temporary IDs (reversible with key).
- **Privacy Enhancing Technologies (PETs):** Homomorphic Encryption, Secure Multi-Party Computation (SMPC), Zero-Knowledge Proofs (ZKPs).
  - Use with sensitive data (medical, financial) across untrusted envs,

Data Obfuscation

- Hide data while keeping it usable.
- Methods: masking, tokenization, substitution, shuffling, blurring.

Differential Privacy

- Prevent re-identification from query output.
- Add calibrated noise so group trends stay useful.

Source Code

https://github.com/josias-r/zhaw-sem5-summaries