IT SECURITY

Part I - Security Goals

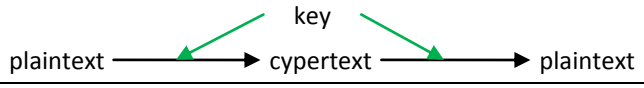
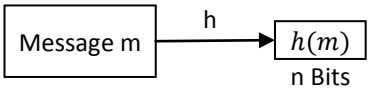
CIA	**Confidentiality** (Vertraulichkeit) - Die Nachricht kann nur vom Sender und Empfänger gelesen werden. **Integrity** (Integrität) - Sicherstellen, dass die Nachricht nicht verändert wurde. **Availability** (Verfügbarkeit) - Die Nachricht muss am gewünschten Zielsystem verfügbar sein.
AAA	**Authenticity** / Authentication - Ich weiss, dass mir die diese Person diese Information gegen hat. **Authorization** (=Zugriff) - Wer darf was sehen, verändern **Accounting** - Ein Benutzer bekommt die gewünschten Information während einer Session.
Non Repudiation	Der Urherber eines Dokuments kann das Schicken einer Information nicht abstreiten.

Verschlüsseln ist nur ein Teil der Sicherheit!!!

Part II - Security Threats (=Bedrohungen)

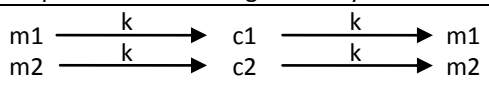
Terms	**Risiko** = Wahrscheinlichkeit * Schaden **Vulnerability / Weakness** (=Verlässichkeit) **Threat** (=Bedrohung)
Attacks	**Passiv** durch Sniffing (=Elektronischen Verkehr lesen) **Active** durch Spoofing (=Ich gebe mich als jemand anders aus) z.B. DNS-Umleitung oder Denial of service (=Ein Computer unerreichbar machen)
Malware (malicious software)	**Virus** (Eigene Kopien in laufendes Programm einbinden -> brauchen Hosts) **Worm** (Eigenständiges Programm) stand-alone **Trojan** (ein Programm dass funktioniert, jedoch noch zusätzlicher Code ausführt) **Trapdoor / Backdoor** (Undokumentierter Eingangspunkt) **Logic Bomb** (Triggert auf einen Event) **Zombie** (Triggert auf einen Remote-Zugriff) **Spyware** (sammeln und senden von geheimen Informationen)
Denial of Service Attacks (DoS)	Viele Zugriffe auf Server machen, sodass CPU-Time, memory oder bandwidth überlastet wird. **Distributed DoS** = Viele verschiedene Angreifer (über Zombies)
Other attacks	**Root Kits** Software zum öffnen eines Systems **Network sniffers** Netzwerkverkehr lesen **Phishing sites** Als eine andere Seite ausgeben **Key loggers** Tastatur mitlesen **Social Engineering** User nach Passwort fragen
Enemies	**Hacker** = gutwill, technisch geschult, kein Schaden zufügen **Cracker** = böswillig, technisch geschult, will Schaden zufügen **Skript Kiddies** = keine Schulung, nur aus Spass
Prevention	Enciphering, Passwords, minimal rights, restrictions (=reduzieren), monitoring

Part III - Cryptography

Vorgang		plaintext —encryption→ cypertext —decryption→ plaintext
Kerckhoffs'principle	Ein System sollte sicher sein, wenn alles ausser dem Schlüssel bekannt ist.	
Types of Functions	**symmetrisch (Secret Key Cryptography)** Ein sicherer Schlüssel	
	asymmetrisch (Public Key Cryptography) public key for encryption private key for description	
	Hash Functions one-way-transformation MD5, SHA	

Jede Verschlüsselung kann durch Versuchen gebrochen werden!!!

Part IV - Symmetric Cryptography

Types of Ciphers	**One-time pad (or Vernam ciper)** gut falls: key random, nur einmal gebraucht	ciphertext c = message m + key k
	Stream cipher keystream sequence repeats, fast	keystream ks = key k + keystreamgenerator ciphertext c = message m + keystream ks
	Block cipher message aufteilen, z.B. DES, IDEA, AES key und blockgrösse genügend gross	
MAC	**=Message Authentication Codes**	Hashwert und Message schicken -> Vertraulichkeit

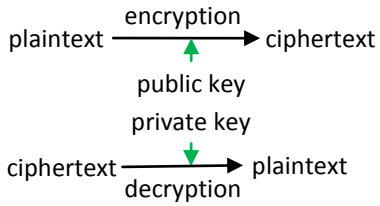
Vorgang	1. generiere zwei Schlüssel k_1 und k_2 2. Verschlüsse Nachricht mit k_1 3. Produziere ein MAC mit k_2 4. Füge MAC hinten an Nachricht an 5. k_1 und k_2 mit KEK (key encryption key) verschlüsseln 6. ciphertext, MAC und k_1, k_2 verschlüsselt schicken

Part V – Asymmetric Cryptography

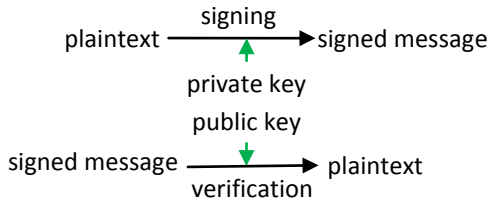
Gebrauch

transmitting over an insecure channel, secure storage on an insecure medium, authentication, digital signaturs

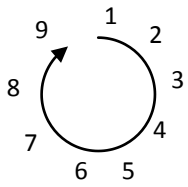
Encryption



Digital Signature



Modular Arithmetic



$$8 + 9 = 17 \equiv 7$$

Addition, $n = 10$

9 ist das additive Inverse von 1;
1 ist das additive Inverse von 9;

Multiplikation, $n = 10$

Nur 1,3,7,9 haben multiplikative Inverse,
da sie relativ prim zu 10 sind

$$a^{-1} = b$$
$$3^{-1} = 7, \quad da\ 3 * 7 \equiv 1$$

Totient Function $\varphi(n)$

berechnet Anzahl coprimes (relatively prime)

$$\varphi(10) = 4, \quad nämlich\ 1,3,7,9$$
$$\varphi(p\ q) = (p-1)(q-1)$$

Exponential

$$4^6 = 4096 \equiv 6$$
$$\boldsymbol{x^y = x^{y\ mod\ \varphi(n)}}$$

RSA (Rivest, Shamir, Adleman)

Key Generation

Wähle zwei Primzahlen q und p (je 512 bits)
Berechne $n = p * q$
Wähle e welche realtiv prim ist zu $\varphi(n) = (p-1)(q-1)$, als ein public key
Wähle d, welche das multiplikative Inverse: $d * e = 1(mod\ \varphi(n))$
veröffentliche e, n und behalte d,p,q geheim

Sicher da, folgende zwei mathematische Probleme exisitieren

- Lange Zahl faktorisieren
- Die eth-Wurzel von x ($c = m^e(mod\ n)$)

Algorithm

Encryption	$c = m^e\ mod\ n$
Decryption	$\boldsymbol{c^d \equiv m(mod\ n)}$
Signature	$\boldsymbol{y = x^d\ mod\ n}$
Verify Signature	$\boldsymbol{y^e\ mod\ n = x}$

Teil 4 – Data Integrity and Authentication

Integrity	detection of bit errors-> with Hash protection of unathorized modification symmetric -> MAC assymetric -> Digital Signature
Hash Functions	grosses Dokument in kleinen Hash (128, 160,256 bits) umwandeln Hash Funktionen müssen Kollisionsfrei sein (zwei Dokumente haben den selben Hash) sollte one-way sein z.B. MD5, SHA-1 (beide nicht mehr sicher)
MAC	=Message Authentication Codes Author und Rezipient berechnen MAC aus Dokument und Key. Bestätigt falls MAC identisch ist.
Digital Signatures	based on public key cryptography sign -> private key verify -> public key

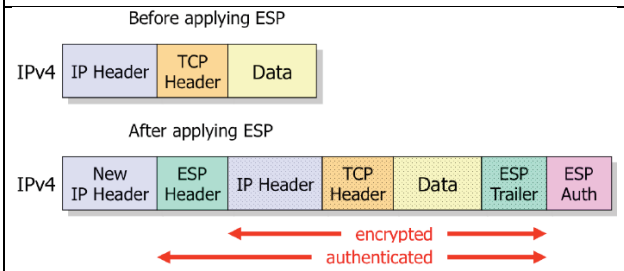
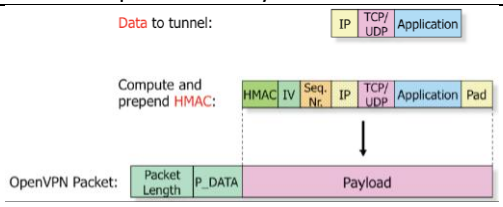
Digital Signatures with RSA	RSA key pair: public key (n,e) private key (n,d) 1) Digitale Signatur der Message x erstellen mit $y = x^d \ mod \ n$ 2) Message x und Signatur y an Empfänger schicken 3) Empfänger verifiziert mit $x = y^e \ mod \ n$
Entropie (Mass für die Unbestimmtheit)	$H(M[n]) = -\sum_{i=1}^{n} p[i] * \log_2(p[i])$ M: Nachricht; n: Anzahl Zeichen; p: Wahrscheinlichkeit 0,1 gleichverteilt, 1bit $H(M[2]) = -(0.5 * \log_2 0.5 + 0.5 * \log_2 0.5) = 1$
Salt	Zusatz (mehr als 8bit)
Digital Cerificates	Ziel: weltweite sichere und authentisierte Kommunikation Umsetzung: mit public keys Problem: Wie weiss ich, dass es der richtige public key ist? Lösung: Digital Ceritifactes
	Standard: X.509 certificates überprüft durch: Certification Authorities (CA)

Teil 7 – End-to-End Communication Security

TLS	=Transport Layer Security ermöglicht mit SSL (Secure Sockets Layer) eine sichere Kommunikation über das Internet über dem TCP Layer provides authenticated, integrity-protected and confidential data exchange Handshake erforderlich
Variationen von TLS	HTTP over TLS = https POP3/IMAP over TLS = pop3s/IMAPS SMTP over TLS = SMTPS FTP over TLS = FTPS
IPsec	=secure communications at the network (IP) layer ist ein Schlüssel-Austausch-Machanismus (IKE=Internet Key Exchange) IP packets should be encrypted and authenticated and integrity protected (ESP) IKE in IPsec = TLS handshake in TLS
TLS vs. IPsec	TLS is clearly the first choise IPsec ist für virtual private networks (VPNs) nützlich

Teil 8 – Virtual Private Networks (VPNs)

Definition	VPN ist ein geschütztes Netzwerk in einem öffentlichen Netzwerk (z.B. Internet)
Gebrauch	Sichere remote-Verbindung von zwei Computern in einer Firma Kunden bestimmte interne Services nutzen zu lassen Mobilen Firmenarbeitern Zugriff zum Firmennetzwerk zu verschaffen
Aufbau	Sicheren Tunnel aufbauen
VPN gateways	Enpunkt zu einem Sicheren Kanal um kryptographischen Schutz hinzuzufügen oder zu entfernen
Protocols	IPsec (offers a tunnel mode) OpenVPN (partly based on TLS)

VPNs with IPsec	**VPNs with OpenVPN**
use ESP (Encapsulating Security Payload) to encrypt, authenticat and integrit IP packets.	layer that runs on top of UDP/TCP available for all plattforms uses the OpenSSL library
	

Firewalls		
Definition	Ein Firewall steuert den Paketfluss zwischen zwei oder mehreren Netzwerken	
Arten	Paketfilter-Firewall	Operiert auf der Netzwerkschicht Wertet Header-Informationen der Netzwerkpakete aus End-to-End Verbindungen werden nicht aufgetrennt es wird nur kontrolliert wer-mit-wem spricht, nicht über was schnell, einfach
	Proxy-Firewall (Application Layer)	Operiert auf der Anwendungsschicht Überprüft Header und Nutzdaten End-To-End Verbindungen werden aufgetrennt
netfilter/iptables	netfilter	Softwareschicht innerhalb des Linux-Kernels
	iptables	Tabellenstruktur innerhalb des Linux-Kernels
Filterfunktionen	ACCEPT (Akzeptanz), DROP (Ablehnung ohne Antwort), REJECT (Ablehnung mit Antwort), LOG (Eintrag)	
Stateless/statefull	stateless	jedes IP-Paket wird isoliert behandelt; getrennte Regeln für Ein-Ausgänge
	statefull	jede Kommunikation wird mitverfolgt; einfacher, gut für TCP (SYN bis FIN)
Network Address Translation, NAT	z.B. PAT (Source Port Address Translation) Rechner hinter einem NAT Router/Firewall sollte private IP-Addressen besitzen Private IP-Addressen werden nicht ins Interent geroutet Vorteil: um IP-Adressen zu sparen, bessere Sicherheit	