1866.KE1 **C**omputer **E**mergency **R**esponse **T**eam berät bei Sicherheitsrelevanten Vorfällen. **Angriffsziele** Kommunikationsweg, Computer, Daten. **Technische Bedrohung** z.B. kosmische Strahlung, Maßnahme Redundanz, Prüfbits. Technik hilft Technik. **Nicht technisch** wäre manuelle Datenverfälschung. “Das Problem sitzt an der Tastatur, obgleich diese einwandfrei funktioniert”. **Unbeabsichtigte Bedrohung** sind Bedienfehler, **D**enial **o**f **S**ervice und jede Form von Spionage **beabsichtigt.** Jedes Abhören ist **passive Bedrohung**. Selbst aktives nutzen einer Abhör-Antenne ist passive Bedrohung. Bei **aktiver Bedrohung** wird eingegriffen durch erzeugen, verzögern oder unterdrückung von Nachrichten.
Vertraulichkeit wird durch **Verschlüsselung** oder Verstecken (**Steganographie**) gewahrt. **Integrität** kann auch durch **Verschlüsselung** geprüft werden. **Authentizität** kann durch **Signaturen** überprüft werden. **Verfügbarkeit** wird durch **redundante** Systeme verbessert.
Lokale Netze als **Stern-** (contra: Switch kaputt, alles kaputt pro: Daten nur von Sender zu Empfänger), **Ring-** (contra: ein Computer kaputt, alles kaputt, alle Daten an allen Computer), oder **Bustopologie** (pro: ein Computer kaputt, nicht schlimm, contra: alle Daten an allen Computern). Hardware: Repeater, Router, Bridge, Switch.
Viren vervielfältigen und kopieren sich selber. **Makroviren** innerhalb von Dokumenten (Word). **Wuermer** brauchen Wirtsprogramme, **Trojaner** geben vor was harmloses zu sein.
Passworte **raten** (Datum oder aus Woerterbuch), **ausprobieren** (alle Kombinationen durchgehen: Brute Force), **ausspähen** (Keylogger). Gutes Passwort nutzt großes Alphabet (A-Z, a-X, 0-9, Sonderzeichen), lang genug, zufällig - keine Namen+Wörter+Datum+Autokennzeichen, Herstellerpasswort ändern, selber regelmässig ändern, nicht notieren, nicht im Klartext speichern.

1866.KE2 **Leitungsverschlüsselung** (pro: nur Nachbar-Computer kennen Verschlüsselung contra: alle Rechner müssen Vertrauenswürdig sein , **Ende zu Ende Verschlüsselung** (pro: kein Unterwegs-Rechner sieht Klartext contra: Sender muss sich mit jedem Empfänger auf Verschlüsselung einigen).
Verschlüsselung durch **Ersetzung** (substitute) oder **Umordnung** (transposition). **Symmetrisch Verschlüsselt** (auch Private Key Verfahren): **ein** Schlüssel. Asymmetrisch Verschlüsselt (auch Public-Key-Verfahren): mehrere Schlüssel. Symmetrisch weniger Aufwand als Asymmetrisch. **Hybrides Verfahren**: Schlüsselübertragung für symmetrische per asymmetrisch übertragen. **Blockverschlüsselung**: Klartext wird in gleich große Stücke aufgeteilt. Wenn zu wenig Text für einen Block (z.B. letzter) **padding** hinzu. **Stromverschlüsselung** (auch Online-Algorithmus) nutzt Leistungskapazität besser aus aber nicht immer optimal. **Hardwaresicherh**.: Umgebendes Bauwerk, Versorgung, Schutzräume wie Server-, Datenträger. **Betriebssystemsicherh.**:verschiedene Benutzerkennungen mit versch. Zugriffsrechten wie lesen, schreiben, ausführen. Persistent (Dateien) Transient (Speicher). **Secret-Key-Verfahren:**
Mono- (Caesar, Rot13) **Polyalphabetisch** (Vigenere, Enigma) **Transposition** (vertauschen, Permutation) . **Feistel** Klartext K in 2 Teile L+R, Schluessel S bestehend aus Teilen S0..n-1; Eine Runde: 1.) K=[L0|R0] 2.) L0=L0 xor f(S0,R0), R0=R0 3.) L1=R0, R1=L0 (1.teilen, 2.codieren, 3.tauschen).
DES Data Encryption Standard, 56Bit Schlüssel, 64Bit Block, 16 Runden, unsicher. **Blowfish** basiert auf Feistel aendert aber beide Haelften pro Runde, 32..448Bit Schlüssel, 64Bit Block, 16 Runden **Twofish** ist Nachfolger, **CAST-128** 128Bit Schlüssel, 64Bit Block, 16 Runden. **AES** Advanced Encryption Standard 10-14 Runden, je Runde Linear-Mixing=>grosse Diffusion, Nonlinear=>ersetzung durch SBoxen, KeyAddition=>Rundenschlüssel zum Block xored. Block als Matrix mit immer 4 Zeilen. Mehr Bit pro Block=>mehr Spalten. **One Time Pad** symmetrisch, nicht zu knacken, Vernam Chiffre. Verfahren: XOR. Schlüssel muss so lange wie Klartext sein, absolut zufällig, nur einmal verwenden, muss geheim sein. Ist Vigenere mit sehr langem Schlüssel. **ECB** Electronic Code Block jeder Block wird mit Schlüssel verschlüsselt. Gleicher Block resultiert in gleichem Geheimtext. Gefahr: **Replay Angriff**. **CBC** Bevor ein Block verschlüsselt wird, Klartext mir vorher verschlüsseltem Block verxoren. Erster Block braucht hier **IV. Counter Mode** statt auf den Vorgängerblock zu warten, kann auch ein Counter mitlaufen der zu jedem Block gexord wird. Vorteil: Verschlüsselung kann parallel an Blöcken arbeiten. **Public Key Verfahren:** Jeder hat 2 Schlüssel private und public. Wer mir was schickt verschlüsselt mit meinem public Key, ich entschluessle mit meinem private Key. Z.B. **RSA** Wenn ich eine Nachricht mit meinem privaten verschlüssle kann zwar jeder sie mit meinem public entschlüsseln, aber das waere dann eine **digitale Signatur**. Etwa 100x langsamer als DES daher oft nur Schluesselaustausch via RSA (Session Key) und dann symmetrisch verschluesseln. Mitm-Attake moeglich wenn oeffentliche Schluessel nicht nachweisbar zum Empfaenger gehoeren. Trust Center. **El-Gamal** weiteres verfahren basierend of diskreten Logarithmen. **Diffie-Hellmann** ist kein Public Key Verfahren um einen symmetrischen Key zu uebertragen. Anfaellig für Mitm Attacken. Variante **authentisiertes Diffie-Hellmann** besser. **Hashfunktionen:** Gut wenn schnell, einfach und gut streut. Hash um Authentizitaet zu pruefen. Auch Fingerabdruck, kryptographische Pruefsumme oder Message digest genannt. **Schwache Kollisionsresistenz:** Finde keine M2 != M1 **zu gegebener** M1 für die H(M1) != H(M2) gilt. **Starke Koll.Res.:** Finde **2 beliebige** M1 != M2 für die H(M1) == H(M2) gilt. **MD5** aus Nachrichten bis zu 2^64bit wird 128bit hash erzeugt - gilt als nicht mehr sicher. **SHA-1** aus Nachrichten bis zu 2^64bit wird 160bit hash erzeugt. **SHA-3** aus Nachrichten bis zu 2^64bit wird 224-512bit hash erzeugt. Arbeit mit Sponge Construction (absorbing + squeezing). **Message Auth. Code** Hash der nicht nur die Nachricht selbst sondern der Hash wird **symmetrisch** verschlüsselt. Sende M, Empfaeger erhaelt M’; MAC=C(key,H(M)); Empfaenger bildet H(m’) wenn H(M’) == D(key,MAC) is alles ok. Macht man das gleiche mit **asymmetrischer** Verschlüsselung hat man eine **Digitale Signatur**. **Zertifikatsmanagement** per **Zertifizierungstelle** (Certification Authority CA, Trust Center) Zertifikat besteht aus (Inhaber, public key, seriennummer, valid to, CA-Name) plus digitaler Signatur des Datensatzes. Zertifikate koennen Qualitaeten haben (Identitaetspruefung). **Certificate Revocation List** falls privater Schlüssel verloren. **X.509** ist int. Zertifikatstandard auch um globale PubKey Infrastrukturen zu betreiben. **Hier Fehlt noch**: (neuer Perso, Identity Based Encryption, erz. Zufalls- und Primzahlen)

1866.KE3 **PGP** verschl. Daten, sicher löschen, komprimiert und verschlüsselt Nachrichten [session key zum symm. versch. wird asymm. verschl. und mit Nachricht geschick.senden(sym(N)+asym(keySym))], signiert [Hash(N) mit privkey assym.versch. und an N hanegen], erstellt Schlüsselpaare, Schlüsselmanagement, hybrides Verfahren, kein Trust Center, Web of Trust, Vertrauensstufen von Ultimate (man selber) bis Unknown. **S/MIME** zum eindeutigen beschreiben woraus eine Nachricht besteht und ob und wie sie verschlüsselt ist. **SSL/TLS** Schicht zwischen Anwendung (http/ftp,...) und Transport (TCP) - SSL Selber ist in Handshake, Alert, Change Cipher und Application unterteilt. **Handshake** [Client Hello, Server Hello, Server Auth.Key.Exchange, Server Hello Done, Client Auth.Key.Exchange, Change Cipher Spec CS, Change Cipher Spec SC]. Nutzeridentifikation innerhalb einer SSL Sitzung früher per Client Cert, heute Username/Pim. Zusätzlich einzelne Transaktionen per Einmalpasswort (TAN) vorproduzierte Liste, an Handy geschickt oder berechnet aus Daten am Bildschirm mit Geraet. **Cookies** http ist zustandslos, um Nutzer wiederzuerkennen setzt Browser cookies. Set-Cookie: NAME=WERT pfad=... domain=... beim wiederkehren muss [<--domain/pfad→] matchen. Bei domain muss das Ende passen und beim Pfad der Anfang. Wenn das passt wird der cookie zum Server übermittelt. Max 3000 pro Browser, je cookie 4kB, pro domain 50 cookies **SSH** secure shell um sich auf entfernten Rechner einzuloggen. SSH [Transport- Authentication- Connection- Filetransfer- Protokoll] Server-Auth. per RSA, DSA, Client-Auth. per RSA, DSA, Password, symmetrische Verschlüsselung als Block oder Strom. Hilfsprogramme SCP, SFTP **X11** protokoll um ein GUI übers Netz zu übertragen. lokal ein X11-Server entfernt ein X11-Client (!) an sich unsicher aber via SSH per ssh -X wirds sicher getunnelt. **VNC** simuliert die komplette Grafikkarte des entferten Rechners lokal. VNC-Server auf entferntem Rechner, VNC Client lokal. (!) VNC ist ueber Portnummer erreichbar (typisch 5900+x x=Nr. Display), da per Port auch via ssh nutzbar: [ssh -L 6666: 127.0.0.1:5901 entfRechner lokal dann vncviewer 127.0.0.1:6666] **RDP** remote Desktop zeigt Ausgaben eines Programms auf dem Server an, ubertraegt Maus und Tastatur Ereignisse zum Server. Kann auch copy & paste vom entfernten zum lokalen Rechner, oder Dokumten von entfernt lokal drucken. **Virenscanner** Schutz vor Bootsektor V. Bootreihenfolge immer Festpl zuerst falls verseuchte DVD/USB Stick angedockt. Schutz vor Datei-V. und Würmer: AntiV. Softwareregelm. aktualisieren, soll Batchbetrieb können-> manuell startbar dazu im Hintergrund ständig laufen und überwachen. Makrovirenschutz im Officeprog. einschalten, eine Anhänge mit exe/com/msi unbedacht öffnen, **Firewall, personal** Angriffe von außen abwehren, Schadprogramme auf dem PC dürfen nichts nach außen übertragen **ACL** nutzen um sichere Windowssysteme zu konfigurieren. ACL einem Objekt zugeordnet, enthält ACE die spezifizieren User oder Gruppen und deren Zugriffsrechte auf das Objekt. Verbote und Erlaubnisse, Verbote haben prio ueber Erlaubnis. **Fehlt noch** (UAC)

1866.KE4 **Webserversicherheit** durch **Minimales System** keine grafische UI, Compiler, LInker, Treiber für Geräte die nicht da sind, nur notwendige Ports offen. Symlinks nicht folgen aus dem webbereich, keine sensiblen Daten im webbereich. Selbst hacken, Integritätstest, Logfiles kontrollieren, regelm. backup, admin via ssh **Firewalls** NAT: Adressumsetzung von öffentlichen zu privaten, **Paketfilter** entscheidet nach Absender IP:Port und Empfänger IP:Port ob erlaubt oder verworfen. Liste von Regeln wird sequentiell abgearbeitet und sobald eine passt wird die angewendet und dann schluss. Speziellere Regeln zuerst, allgemeinere Regeln danach. **Paketfilter kann kein NAT. Stateful Inspection Filter** statt für jede Flussrichtung eine Regel nur pro Verbindung. 2n zu n+1 Regeln. **Application Level Gateway ALG** auch Proxy genannt ist ein Stellvertreter. Nutzer hat keinen direkten Kontakt zum Internet, nur zum Proxy. Proxy kann Nutzer authentifizieren, kann seiten cachen, **WAF** wenn der Proxy http nachrichten filtert, cookies bleiben auch im WAF und werden vorgehalten. Nutzer sieht anstelle nur WAF cookies. **DMZ** auch screened subnet, netz zwischn inter und internem Netz durch 2 Paketfilter begrenzt, zusätzlich auch mit ALG. Folgende Dienste durchlassen: http,smtp,ssh,nntp,dns **Reaktion auf Zwischenfälle** kein Panik, Sofortmassnahmen, Beweise sichern, Problemanalyse, Massnahmen **Organisatorisches BSI Grundschutzhandbuch** listet Gefahren:Bausteine:Maßnahmen, Sicherheitskonzept, Plan Do Check Act, **Schaden Kategorien** Finanz, Juristisch, Operationell, Persönlich, Image **Organisationsstrukturen** muessen geschaffen werden.

1868.KE1 **Anonymität** dem Namen nach unbekannt. **Senderano**. (Client Anonym.) Anruf Beratungsstelle, **EmpfängerAno**. (Server Anonym.) Chiffre Anzeige, **Komplette Anon. Verkettbarkeit** Verbindung zwischen Handelndem und Handlung. **Pseudonymität** abgeschwächte Anonymität, lässt versch. verkettungen zum P. zu. Person tritt für gewissen Zeitraum, oder geg. gewissen Gruppen oder Personen unter Pseudonym auf. **Personen.P**: Person nutzt es allen Kommunikationspartnern ggu. **Rollen.P**: Person nutzt es in einer bestimmten Rolle, **Beziehungs.P:** Person nutzt es nur einer anderen person ggu. **Rollenbeziehungs.P** je nach Kommunikationspartner und Rolle anderes P. **Transkations.P** nur für eine Transaktion nutzen. Kleinstmoegliche Verkettungsmenge. **Gruende Anonymitaet**: Schutz von Finanzdaten, Konsumdaten, Kommunikationsdaten, Aufenthaltsdaten. Identitätsdaten Internet: beim Surfen: IP, ggf. Nutzerkennung, Datum/Uhrzeit, aufgerufenen URL, von wo kam man (refererr), Info über Browser, Sprach, OS. **Einfache A-Techniken**: **Broadcasts**, nachricht an alle verschleiert 1:1 beziehung Sender:Empfänger, **Dummy Traffic**: eigentliche Nachricht fällt in der Vielzahl unsinniger Nachrichten nicht auf, **Proxis.**
Mixe brechen Verkettung ein- und ausgehender Nachrichten auf. Nimmt eine bestimmte Menge Nachrichten gleicher größe auf und dann in abweichender reihenfolge wieder versandt, zudem **asymmetrisch** verschlüsselt. Nachricht N, Sender S, Mix M, Empfaenger E mit je zwei Schlüsseln G(eheim) P(ublic). **Ablauf:** S macht crypt(E+crypt(N, PE), PM) schickt das an M, der packt mit GM aus und erhält E+crypt(N), M schickt weiter an E und der packt crypt(N) mit GE aus. **Mix Kaskade** - gleiches Prinzip: Sender verschlüsselt zuerst mit PE, dann mit PM des letzte Mix, usw. **Onion Router (TOR)** Initiator handelt per Diffie Hellmann **symmetrische** Schlüssel (session key) mit jeder Station (hop) aus, es wird eine schicht-weise Verschlüsselung wie bei den Mixen gemacht, nur eben symmetrisch. Tor ist außerdem effizienter als Mix, Last besser verteilt und es gibt bei Tor hidden Services. Um anonym zu surfen kann man auch einen **Rewebber** nutzen. Der entkoppelt den User vom Server und entfernt zudem alles was Rückschlüsse zulässt aus dem Request. Aehnliches gibt es auch als **Remailer.**
JavaScript kann auf history oder cookies im Browser zugreifen. **Risiko**: Javascript kann indirekt über URL-mit-parameter Aufruf an entfernte Computer machen und so Daten übermitteln. **Same-Origin-Policy** besagt, dass JS die von einer andere Domain geladen werden, keinen Zugriff auf z.B. history oder cookies haben. **Java-Applets <applet>** laufen im browser in einer Sandbox und können nicht: lokale Dateien lesen, ändern, schreiben; andere programm lokal starten; Netzverbindung zur zum computer starten von dem sie geladen wurden. **ByteCodeVerifier** sorgt dafür dass keine NullPointerExceptions oder anders was den Rechner abstürzen lassen, auftreten. **JavaClassLoader** sorgt dafür, dass wichtige Basisklassen nicht überladen werden können **SecurityManager** regelt den Zugriff auf sicherheitsrelevante Ressourcen. **ActiveX <Object>** von Microsoft uns basiert auf COM. Risiko: Kann auf alle Betriebssystem-Ressourcen zugreifen. ActiveX kann **zertifiziert** werden. Das zeigt aber nur, dass es unverändert übertragen wurde und wer der Ersteller ist. Das ActiveX kann aber immer noch Schaden anrichten. Selbst **PDFs** koennen aktive Inhalte enthalten - man kann z.B. Programme starten, wenn man weiss wie der Pfad ist. **Computer Forensik** Angriff erkennen (IDS), Beweise sichern, Angriff analysieren, Angriffsspuren und Schwachstellen restlos beseitigen. Beweise sichern (Befehl in UNIX): Hauptspeicherinhalte (/proc/meminfo), Festplatteninhalt, angemeldete Benutzer (who), laufende Prozesse (ps -elf), offene Netzverbindungen (netstat), Systemzeit (date). Ausser Festplatte alles nicht persistent. Um die nicht persistenten zu sichern muss man aber den Zustand des Systems (“den Tatort”) veraendern. Selbst Hilfsprog. auf externen USBs muessen ans System angemeldet werden und werden vom gleiches ausgefuehrt. Festplatte kopieren unter UNIX mit dd if=/von/hier of=/nach/da (fehlt: noch ein paar Befehle aus dem script)

1868.KE2 **Subjekt** fuert eine **Operation** auf einem **Objekt** aus. **Hardwarezugriffskontrollen**. Anfangs 2 Modi **Systemmodus** (alle CPU Befehle), **Benutzermodus** (eingeschr. CPU Befehle) Uebergang zwischn Modi nur durch Interrupt. Neueres Konzept: **4 Ringe** 0-OSKernel, 1-Treiber, 2-Dienste, 3-Anwendungen. Genutzt werden nur 0 und 3. MMU für Segmentbasierte oder Seitenbasierte Adressierung. Trusted Computing Nutzer nimmt an, dass sein System wie vermutet funktioniert. Isolation of Programs, Separation of User and Admin Processes, Long-Term protected storage, identification of config, verifiable report of platform identity and config, hardware base for protection. TPM Trusted Platform Module prüft Integrität des Rechners, meldet ans OS.
Betriebssystemzugriffskontrolle ACL jedes Objekt weiss welches Subjekt welche Operationen ausführen darf. Capabilities jedes Objekt weiss für welche Subjekte es welche Operationen ausführen darf. **Biba-Modell** stellt **Datenintegrität** sicher, **“No read down”** man liest keine niedrigstufigen Informationen, da angenommen wird, dass unzuverlässig. “**No write up**” Man darf keine höherstufigen Informationen schreiben, da man selber nie so schlau wie sein Chef ist. Bell-La-Padula-Modell ist über Vertraulichkeit **“No read up”** Man darf nur lesen, was der eigenen und weniger vertraulichen Stufen entspricht. **“No write down”** Subjekte die Objekte erzeugen verleihen diesen Objekten die gleiche Vertraulichkeit. Wenn ein General, der geheime Pläne lesen und erstellen darf in einem Kantinenspeiseplan einen Tippfehler behebt, dann ist damit der Speiseplan auch streng geheim. Um das zu vermeiden, gibt es das **High-Watermark Prinzip**: Nur wenn das bearbeiten des geheimen Plans und des Speiseplans im gleichen Prozess stattfindet, dann wird der Speiseplan geheim. Wenn der General das aber in 2 Prozessen macht, bleibt der Speiseplan ungeheim. Bei **Datenbanken**, sie zunächst einmal auch an die Betriebssystemzugriffskontrollen gebunden ist, wird das noch erweitert. Eine Datenbank kann Rechte für **Nutzer** (Subjekte) auf einzelne (Objekte) T**abellen, Datensätze, Attribute** in einem Dokument spezifizieren. SQL Kommando **grant.**
Benutzerauthentisierung per **Passwort** Einmalpasswort … **Kerberos** trennt direkte Zuordnung von User zu Server, Server kennt seine User nicht. Es gibt einen Authentication Server AS, der beide “trennt” der AS kennt alle (**symmetrischen**) Schlüssel aller User und aller Server. Will User A auf Server 2, dann wird ein Kerberos Ticket (aus SessionKey, NameClient, Beginn und Ende Gültigkeit) verschlüsselt mit Server Key, User kann es nicht lesen und manipulieren. Der gleiche SessionKey wird vom AS zusammen mit Zeitwert aus User Anfrage an AS per Userkey verschlüsselt. Mit dem Secret Key, kann er nun auf den Server drauf. 2 User 3 Server und 1 Ticket ergibt 6 Schlüssel zur gleichen Zeit im Umlauf. **Biometrie** zur **Identifikation, One-to-Many-Matching** ein Videosystem finden am No-Pants-Day in der NYC subway die Beinkleidträger heraus. Oder zu **Verifikation, One-to-one-Matching** ein Zutrittssystem oeffnet die Tuer nachdem ich am Fingerabdruck erkannt wurde. **False-Rejection-Rate FRR** wie oft wird ein legitimer Benutzer abgewiesen **False-Acceptance-Rate FAR** wie oft wird ein nicht legitimer Nutzer angenommen. **Equal-Error-Rate** Punkt an dem FRR==FAA ist ein guter Toleranzwert. **RADIUS Remote Auth Dial In User Service** Radius Client zu Radius Server is IP basiert, beide nutzen preshared key zur Sicherung. Es werden nur Hashwerte übertragen, Hash(Passwort + presharedsecret + request authenticator). **Challenge Response** der Server stellt dem Client eine Aufgabe (Challenge) die mit der korrekten Antwort (Response) gelöst werden muss. z.B. beim Banking die Challenge: “Gib mir TAN Nummer 42”.

1868.KE3 WLAN **Sicherheitsanforderungen**: Vertraulichkeit, Authentizität, Integrität, Verfügbarkeit. **AdHoc-Modus** (wie Ethernet-Kreuzkabel) vs. Access Point - verwaltet die Kommunikation im W-Lan auch **Infrastrukturmodus** genannt. **SSID** Name für Access Point wird neben anderem per **Beacon Frame** ausgesandt damit Geraete den AP/SSID finden. SSID kann verborgen werden, ist anber Security by Obscurity da SSID auch anders als per Beacon ausgetausch wird. **WEP** Stromverschlüsselung, *Msg = concat[IV, XOR(RC4_PRNG(concat(IV,WEPKey)) , concat(Plaintext, crc-32(Plaintext)))]* WEP hat unsichere Integritätsüberwachung, unsichere Verschlüsselung **WPA-TKIP** erste verbesserung von WEP, kompatibel zu WEP-Hardware, nutzt Stromverschlüsselung kann Personal Mode (per shared secret) oder Enterprise Mode (per z.B. RADIUS), Brute-Force Attacke ist genau so aufwaendig wie bei WEP, Ablauf zu generierung eines immer anderen WEP Keys:
`cipher = WEP[ keymix( tkip-sequence-number, transmit-address, enc-key, fragment( tkip-sequence-number, michael ( michael-key, plaintext) ) )` Schutzziel **Integrität** ist gewährleistet, aber grundsätzlich anfällig für Brute Force aber nur Infos ueber das eine geknackte Paket. Keine anmeldung am ELAN möglich und auch keine falschen Pakete koennen eingeschleust werden. Alles in allem eine deutliche Verbesserung zu WEP. **WPA2** ist unabhängig vom WEP Verfahren und es nutzt Blockverschlüsselung AES. Hier auch Personal oder Enterprise wie bei WPA-TKIP. **EduRoam** authentisierung: Wenn fh-hannover besucher in fu-hagen sind, alle wlan an allen Hochschulen haben SSID eduroam, lokaler hannover besucher hat sein notebook und da seinen nutzernamen inkl, **realm** fh-hannover.de - das wird vom RADIUS in Hagen erkannt und an einen übergeordneten RADIUS weiter, bis einer den der FH Hannover kennt, an den die Authentisierungsanfrage durchgeleitet wird und die authentisierung erfolgt. **Öffentliche WLANs** von unterwegs, keine Verschluesseliung der Funkstrecke, der Accesspoint leitet zunächst alle http Anfragen an eine Login Seite des Betreibers, nach authentisierung wird die konfig des Paketfilters/Routers geandert, damit Vollzugriff aufs Internet ist. Sicherheitsprobleme: Jeder kann seinen Accesspoint “T-Mobile” nennen und Daten abfishen. Funkstrecke klartext, alle an dem Access-Point koennen alles mitlesen. SSL ist wichtig, auch bei email. Besser VPN und alles da durch abwickeln. **VOIP** Klassische Telefonnetze: **Zugangsnetz** (Endgeraet zu Vermittlung), **Kernnetz** (verbindet Vermittlungen) klassisch **Leitungsvermittelt**, voip **Paketvermittelt. Signalisierung** (das Bimmelimmm) im klassischen per **SS7 Signalling System 7**. Mobilfunk ist im Prinzip genauso, nur dass die Endgeraete zur Vermittlung, verschlüsselt per **A5**, funken. **Home Location Register** enthält jeden Nutzer mit Telefonnummer und gebuchten Diensten und MSC, **Visitor Location Register** enthält kopie aller Datensätze der Geraete im Empfangsbereich Handover wechsel von Funkzelle zum Funkzelle. **SIM** authentisiert. Voip signalisiert (Bimmelimm) per **SIP** Session Initialisation Protocol und Sprachdaten dann per **RTP** RealtimeTransportProtocol kontrolliert von **RTCP**. Media Gateway ist die Verbindung von klassisch zu voip Netz. Nummer wird zu DNS name umgewandelt. Telefonnummer umdrehen, eine Prise Punkte rein, fertig: +492211234 wird zu 4.3.2.1.1.2.2.9.4.e164.arpa. Bei Voip kann mal **CLIP** unterdrücke oder fake nummer senden. **SRTP** ist die verschluesselte Variante von RTP. **Skype** is nicht als sicher angesehen, da closed

source. (Fehlt: Smartphonesicherheit.)