1 Kryptographie

1.1 Kommunikationsmodell

- **Passiver Angriff:** Abhören der Daten
- **Aktiver Angriff:** Manipulation der Daten

1.2 Schutzziele

- **Vertraulichkeit:** Verbindung zwischen A(lice) und B(ob) ist sciher O(scar) kann nicht mithören.
- **Integrität:** Naricht wird nicht verändert, bzw. A und B stellen Änderung fest.
- **Datenauthenzität:** B weiß sicher das Nachricht von A stammt. (Mehrere Emails mit selber IP und unbekannter Autor - fallen in diese Kategorie)
- **Instanzauthenzität:** B kann die Identiät von A zweifelsfrei feststellen.
- **Nichtabstreitbarkeit:** B kann Naricht von A zweifelsfrei auch gegenüber einer dritten Partei als Narich von A nachweisen.

1.3 Kryptographie für Schutzziele

- **Vertraulichkeit:** Verschlüsselung
- **Integrität:** Hashfunktion, Message Authentification Code MAC, Signaturen (da geb ich dir Brief und Siegel und für)
- **Datenauthenzität:** MAC, Signaturen
- **Nichtabstreitbarkeit:** Signaturen
- **Instanzauthenzität:** Challange Reponse Protokoll (Hashwert gemeinsam bekanntes Passwort ist der Challange , siehe Beispiel Captcha)

1.4 Auguste Kerckhoff 1883

- Ist ein System nicht beweisbar sicher, so sollte es parktisch sicher sein.
- Das Design eines Systems sollte keine Geheimhaltung erfordern (Feind darf es wissen).
- Ein Kryptosystem muss einfach bedienbar sein.
- Ein Angreifer kennt das kryptographische Verfahren, nur die privaten oder symmetrischen Bratwürste sind geheim.

1.5 Symmetrische und Asymmetrische Verfahren

- **symmetrische Verfahren:** A und B selben Schlüssel, deswegen darf ihn O nicht kennen.
- **asymmetrische Verfahren:** A und B besitzten jeweilse einen öffentlichen sowie einen privaten Schlüssel. Die öffentlichen Schlüssel müssen sicher ausgetauscht werden.

1.6 Symmetrische Verschlüsselung im Detail

1.6.1 Ceasar

- Verschiebt Buchtaben um festen Betrag.
- Da es nur 26 Mögliche Schlüssel gibt -¿ Für die Sicherheit muss der Schlüsselraum (Menger möglicher Schlüssel) groß genug sein.
- Modifiziertes Ceasarverfahren hat 26! Möglichkeiten (SEHR SEHR VIEL) läßt sich aber über relative Häufigkeit knacken.
- Relative Häufigkeit = Beispiel für statische Angriffe -¿ alle Kryptoanalytischen Mehtoden müssen berücksichtigt werden großer Schlüsselraum alleine reicht nicht aus.

1.7 One-time Pad

- Basiert auf Xor und Bitzahlen (siehe Folige 36-40)
- Absolut sicher bei Unkenntnis des Schlüssel - jeder sinnvolle Text gleich warhscheinlich.
- Ciphertext liefert keine Information über den Klartext oder Schlüssel.

1.8 Absolute Sicherheit

- **Absolute Sicherheit:** Wenn ein Angreifer mit unbeschränkten Ressourcen (Zeit, Rechenleistung) das Verschlüsselungsverfahren nicht knacken kann - bsp. One-Time-Pad.
- **Nachteil** Schlüssel muss genauso lang sein wie Botschaft selbst (Krasses Beispiel 10 GB benötigen 10 GB) Claude Shannon, 1948

1.9 Praktische Sicherheit

- **Praktische Sicherheit:** Schutz vor Angreifer mit beschränkten Ressource

1.10 Modifiziertes One-time Pad

- Zweimaliges Anwenden desselben Schlüssel auf unterschiedliche Texte ist nicht besonders effizient - über die Summe der beiden Geheimbotschaften erhält Angreifer die Summe der Klartexte
- Man kennt also Summe der Klartexte ohne Schlüssel kennen zu müssen
- Also müssen wir nur noch zwei sinnvolle Klartexte finden die entsprechende Summe liefern -¿ selbst kleine Veränderungen des kryptographischen Verfahren können zu einer massiven Senkung der Sicherheit führen

1.11 Definition Sicherheitsniveau

- **Sicherheisniveau:** Ein Kryptoverfahren hat ein Sicherheitsniveau von n-Bit wenn der Angreifer 2 hoch n Versuche benötigt um es zu brechen.
- praktische Sicheheit bei größer gleich 100 Bit (heutzutage)

1.12 Stromchiffre

- Aus Schlüssel der größer gleich 100 bit ist wird ein peseudozufälliger Schlüssel generiert
- Der Schlüsselstrom wird dann mit dem Klartext addiert (XOR)
- erste Idee gleichen Schlüssel mehrmals hintereinander in den Pseudozufallsgenerator - das ist aber unsicher
- Aus 100 bit Zufalls schüssel läßt sich mit einem Pseudozufallszahlengenerator natürlich kein echter 200 bit Zufallschlüssel erzeugen - Schwierigkeit liegt darin den 100 Bit Schlüssel zu erraten und dann den Schlüsselstrom zu berechnen.
- wichtig aus teilen des Schlüsselstroms dürfen keine Vorgänger und Nachfolger bestimmbar sein

1.12.1 Stromchiffre:LFSR = Linear Feedback Shit Register

- 1. Schritt Register wird mit zufälligem Schlüssel initialisiert
- Schlüssel wird von links nach rechts durchgeschoben: Was rechts rausfällt ist die Pseudozufallszahl
- ausgewählte Bits im Register werden mit XOR addiert und dann rechts eingefügt.
- bei einem 11 Bit register lassen sich nach 11 rausgeschobenen Zeichen bei Kenntnis des LFSR ... Nachfolger bestimmen

1.12.2 Die SUPER POWER: LFSR = Linear Feedback Shit Register

- Jetzt werden die wesentlichen Schwächen verhindert!
- Shrinking Generator: Zwei LFSR R1 und R2 wenn R1 1 ausgibt wird Ausgabe R2 genommen falls R1 0 ausgibt wird sie verworfen.
- Summations Generator: Easy Peasy - R1 und R2 addiere einfache die Ausgaben.
- boah jetzt haben wir potentielle Angreifer krass verwirrt Manipulation
- Beispiele: Krasses Handy 1990 / Sprachverschlüsselung zwischen Mobiltelefon und Funktmast

1.13 Blockchiffre

- Blockchiffren bilden Bistrings fester Länge auf Bitsrings fester Länger ab
- moderner Blockchiffren verarbeiten Bitstrings der Länge 128 (Klartextblock)
- Genutzte Schlüssel müssen eine Bitlänge ¿= 100 haben
- Für längere Klartexte (128 Bit) werden sogenannet Betriebsarten genutzte
- Zwei Arten: Feistel Chiffren - Prominenentes Beispiel DES und Substitutions-Permutations-Netzwerk SPN - Prominentes Beispiel AES nachfolger von Design - wenden beide im Prinzip dasselbe an:
  - Diffusion: Verteilung der Information des Klartext über den gesamten Chiffretext
    jedes Bit des Chiffretext hängt von jedem Bit des Klartext ab
    Bei Änderung eines Klartextbits ändern sich 50% der Geheimtextbits
    wird durch Permutationen realisiert
  - Konfusion: Zusammenhang zwischen Klartext und Chiffretext soll hochgradig komplex sein gleiches gilt für Schlüssel und Chiffretext -realisiert durch Subsitituionen (nicht lineare Abbildung)
  - Rundenbasiert: Wiederholte Anwendung von Diffusion, Konfusion und Schlüssel

1.14 Diffusion und Konfusion unter der Lupe

- Reihenfolge der Bits werden über den gesamten Block (128 Bit) vertauscht + mehrere Runden nötig damit Diffusion zum tragen kommt
- S-Box = nichtlineare Abbilung
  KLARTEXTBLOCK (128 Bit) 1) Aus Schlüssel der länge 128 bit werden n Rundenschlüssel der länge 128 Bit hergeleitet

  dann wird der erste Runden Schlüssel auf den Klartext addiert (XOR) dann wenden wir die nicht lineare S-Boxen(haben die länge 8-16 bit) auf den Schlüssel an und anschließend permutieren wir das ganze, dies wiederholen wir n -mal und erhalten den Chiffretext
  Lassen wir die Substitution weg - ist die Blockchiffre eine lineare Abbildung - deswegen lineares Gleichungssystem und effizient lösbar
  Lassen wir Permutionen Weg hängt Bit 1- 16 des Geheimtextes nur nur von Bit 1-16 vom klartextblock ab damit (Attacke auf 1 Bit 17 usw über monoalphabetische Verfahren möglich)

1.15 Betriebsarten

- Blockchiffren: verschlüsseln zunächst nur Klartexte einer bestimmten Blockgröße - heute meist 128 Bitlänge
- Für längere Klartexte wird der Text in entsprechende Blöcke, falls der letzte Block zu klein ist wird dieser mit einem Padding aufgefüllt

1.15.1 ECB-Mode = Electronic code Book

- Plaintext + immer gleich Key auf jedem Block mit Blockcipher encryption
- gleiche Klartextblöcke werden zu gleichen Chiffretexte verschlüsselt
- AES im ECB = Wenn man Pinguin kommt Pinguin etwas schwarz wieder raus
- aus den Punkten davor folgt, dass wir weitere Parameter und nicht nur den Schlüssel und Klartext zur Erstellung des Chiffretextes benötigen
- daraus folgt wir verwenden andere Betriebsmodi CFB und OFB

1.15.2 CBC-Mode = Cipher Block Chaining

- Initialisierungsvektor IV - wird benutzt und vor Verschlüssung auf Plaintext (summiert XOR) angewendet (um Rückschlüsse auf Key und Klartext zu verhindern)
- entstandener Ciphertext wird als nächste IV für den Block benutzt
- Entschlüsselung entprechen umgekehrt

1.15.3 Counter Mode CTR

- Zuerst erzeuge Wertepaar Nonce = Zuffalls Number only used once und verknüpfe sie mit Counter für jeden Block (z.B. Counter 0000 für erst nächster Block hätte Counter 0001 usw.) die Verknüpfung von Nonce mit der Counter erfolgt beispielsweise über simples dranhängen oder über Addition (XOR)
- Mit Hilfe der Block-Cipher Encrytion und dem Key wird dann aus diesen Werten ein Schlüsselstrom erzeugt
- Addition auf den Plaintext ergibt dann entsprechenden Ciphertext - zur Entschlüssung muss dann natürlich entsprechender Schlüsselstrom einfach auf den Cipher addiert werden.
- Damit dieser Modus sicher ist muss die Blockchiffre -1.) Die Elemente müssen Permutiert werden -¿ kleine Änderungen (Zähler) führen zu großer Änderung Ciphertext und Blockchainig muss nicht linear sein damit auf den Schlüssel nicht geschlossen werden kann - Zusammenfassung nicht linear für Schlüssel - Permutation für Eingabe in die Chiffre

1.15.4 Was passiert mit dem Pinguin?

- ECB und CTR töten Pinguin

1.16 Hashfunktionen

- Ziel einer Hashfunktion Integritätschutz
- Bilden Bitstrings beliebiger Länge auf Bitstrings fester Länge ab.
- Für Datenintegrität
- Hashfunktion ist kryptisch stark wenn: Preimage resitance: Es ist praktisch unmöglich einen Eingabeparameter zu finden der einen entsprechnden Hashwert liefert. Collision resistance: Es ist praktisch unmöglich zwei Eingabeparameter so zu finden das sie denselben Hashwert liefern.
- H soll effenzient berechenbar sein, die Umkehrfunktion soll aber praktisch nicht berechnbar sein (Einwegfunktion)
- Untersschied Hash und MAC - Hash für Datenauthenzität - MAC für Datenauthenzität und Integrität
- Wie funnktioniert das Ganze Merckle Damgard Konstruktion: man könnte z.B. f als blockchiffre nehmen und dann ein Initialisierungsvektor + den Block in f reingeben - was da rauskommt stecken wir mit dem nächsten Klartextblock + den Chiffreblocktext wieder in f rein und wiederholen das Ganze bis wir so den ganzen Text durchgegangen sind - der letze Cipherblock ist dan der Hashwert.

- Beispiel Download von Datei - Hashwert soll stimmen
- da Hashfunktionen öffentlich zugänglich sind - läßt sich der Absender einer Naricht nicht ermittel (Schutzziel Datenauthenzität ist nicht gewährleistet)

1.17 Macfunktionen

- Bei symmtrisscher Verschlüsselung (nur A hat k und B hat k) liefert der mac(k,m) eine Prüfsumme mit der B feststellen kann, dass die Nachricht wirklich von A stammt
- so erfüllte es aber Nichtabstreitbarkeit weil nur A und B - den Schlüssel k kennen (dafür werden Signaturen benötigt)
- mac wendet blockchiffren an CBC-MAC, XOR-MAC (kryptische starke Hashfunktion = HMAC)
- CBC Geht für Mac - aber CTR geht nicht für MAC (weil der Blockweise auswertet)

1.18 Secure Messaging

- Ziel Sicherer Kanal zwischn Alice und Bob
- Sowohl Mac als auch Klartext werden verschlüsselt (IPsec)
- MAC wird zuerst über Klartext geschickt und dann folgt der verschlüsselte Klartext (SSH)
- MAC wird zuerst über Klartext geschickt und Klartext mit Mac nochmach verschlüsselt nachgeschickt (SSL)
- Nur IPsec ist sicher
- für Mac und Klartext zum berschlüsseln andere Schlüssel **(Trenne wo du trennen kannst.)**

1.19 Instanzauthentisierung

- Verfahren zur Intanzauthentisierung sind: Benutzername/Passwort - biometrische Merkmale/Fingerabdruck - Challenge-Response Verfahren

1.20 Challange Response Verfahren (symmetrisch)

- Ziel: Nachweises eines Geheimnisses ohne dies offen zu legen
- A und B proudly owning the same key A schickt challange = (Zufallszahl) -B bildet MAC von challange mit key und schickt dies A zu A überprüft ob die MAC wirklich von dem key stammt.

1.21 Asymmetrische Verfahren

- Es gibt zwei öffentliche Schlüssel pk zwei private Schlüssel ps.
- Mit sk wird die Prüfsumme berechnet mit - pk kann dann Prüfsumme von jedem überprüft werden (IA= Instanzauthentisierung)
- Grundidee asymmeterischer Verfahren: Eingwegfunktion mit Falltür - f ist effizient berechnbar - Umkehrfunktion nur mit Zusatzswissen berechenbar)
- pupliziert 1977 Rivest, Shimir und Adlemann - erste bekannte asymmetrische Verfahren / erfinder wurden reich - aber es hat etwas gedauert
- siehe mathevorlesung
- $c = m^e \% N$, $m = c^d \% N$

1.22 Hybridverfahren

- A verschlüsselt mit öffentlichem Schlüssel von B - einen random Key und schickt diesen B zu - nun kann symmetrisch weiterverschlüsselt werden.
- A muss sich sicher sein, dass der Schlüssel zu B gehört.

1.23 Signaturverfahren

- A signiert m mit privaten Schlüssel und schickt Signatur sowie Nachricht m zu B - B kann aus der Signatur über den öffentlichen Key m zurückgewinnen und mit dem zugeschickten m abgleichen.
- Bei Signaturverfahren nach RSA - wird der Hashwert der Naricht verschlüsselt. - über öffentlichen Key kriegt man also Orginal Hash der Nachricht
- erfüllt die Nichtabstreitbarkeit nur der Inhaber des privaten Schlüssel kann Nachricht signiert haben.

1.24 challange Response - Asymmetrisch

- Nachweis eines Geheimnisses ohne dies offen legen zu müssen - schicke challange c an B - mit sk verschlüsselt zurückbekommen - mit pk gewinnt man wieder das Geheimnisses - B kennt also sk

1.25 RSA - Zusammenfassung

- Verschlüsselung
- Datenauthentiesierung (Signaturverfahren)
- Instanzauthentisierung (Challange-Response-Verfahren)
- Für alle Verfahren - werden unterschiedliche Schlüssel eingesetzt (Trenne wo du trennen kannst) sonst sind Angriffe möglich.
- Später Beispiel angeben... .

1.26 Vertrauensmodelle

- Problem 1 + 2: - Nutzung öffentlicher Schlüssel falscher Person - keine Vertraulichkeit mehr gewahrt / falsche Zuordnung des öffentlichen Schlüssel zu einer Person - Datenauthenzität ist nicht gewährleistet
- Direct Trust: Nutzer erhält den öffentlichen Schlüssel direkt vom Schlüsselinhaber (kleine Virtual Private Networks)
- Web of Trust: Nutzer signieren sich gegenseitig ihre öffentlichen Schlüssel (PGP, GNUPG) Also Bob signiert Carls öffentlichen Schlüssel - Alice vertraut Bob nur vernünftige Leute zu signieren - Alice prüft mit Bobs öffentlichen Schlüssel den öffentlichen Schlüssel von Carl -jedem Schlüssel zugeordnet sind: Name des Schlüsselinhabers/ Key Legitimacy (3 Stufen, Grad des Vertrauens in den öffentlichen Schlüssel) leitet sich ab aus den Owner Trusts der Signierer und der Anzahl der Signaturen /Signaturen des öffentlichen Schlüssels (Schlüsselbund) - verwentet bei OpenPGP
- Owner Trust besitzt 5 Stufen: unbekannt, kein Vertrauen, geringes Vertrauen, volles Vertrauen, absolutes Vertrauen (eigener Schlüssel)

1.26.1 Nachteile Web of Trust

- Owner Trust - erfordert hohes Wissen der Nutzer
- Signaturen sind juristisch nicht bindend
- Zurückziehen Zertifikate nicht einfach umsetzbar

1.26.2 Vorteile Web of Trust

- Deutschliche Verbesserung gegenüber Direct Trust
- Umgesetzt in PGP (Pretty good Privacy) und OpenPGP
- Zahlreiche Schlüsselserver für Signaturen und öffentliche Schlüssel
- Einige Server bieten Certification Service für PGP und GPG Schlüssel an

1.26.3 Hierachical Trust

- Funktioniert über die Stelle die Zertifikate ausgibt CA = Certification Authority
- Aufgabenteilung in RA = Registration Authority (Sicherheitsrichtlinien, Zertifikatsanträge, Inhalte der Zertifikate, Leitet Anträge an CA weiter) und CA = Certification Authority (stellt eigentlichen Zertifikate aus)
- Certificate enthält öffentlichen Schlüssel des Nutzers - dieser schickt bei Übertragung öffentlichen Schlüssel der CA (der bekannt sein sollte)- sowie message und Signatur
- Certificate können entzogen werden Certificate Revokation Lists (CRL)
- All Instanzen einer PKI (Root-CA, Teil-CAs, Endnutzer) müssen ein definiertes Maß an Sicherheitsstandards einhalten.
- Certificate Policy (welche Sicherheitsvorgaben müssen eingehalten werden) + Certificate Practise Statement(wie werden die Sicherheitsvorgaben umgesetzt) : beschrieben in RFC 3647
- Zertifikate müssen: Inhaber eindeutig identifizieren, Schlüsselnutzung festlegen, Ausstellende CA identifizieren - zwei Standards: X509: Eingesetzt zur sicheren Kommunikation im Internet (https) - Card Verifiable Certificates (cvc): Eingesetzt zur sicheren Kommunikation zwischen/zu Smartcards
- OID = Knoten in einem hierachisch zugewiesenem Namensraum - Folge von Nummern die Position beginnend von Wurzel beschreibt(damit wird z.B. der umgesetzte Algorithmus beschrieben)
- Zertifikate haben momentan 3 Versionen - aber v3 dann extrainfos (Extenstions bsplweise infos : issuer und subject(Telefon email etc.)