Möglichkeit der Authentifizierung

- Sendung der Nachricht mit privatem Schlüssel verschlüsselt → Entschlüsselung mit öffentlichem Schlüssel → Nachteil: Jeder kann die Nachricht entschlüsseln

Ergänzungen Hash-Algorithmen

- Signatur = eindeutiger Bezeichner → auch eine nur winzig kleine Änderung am Dokument (z.B. ein Buchstabe anders) ändert die Signatur komplett
- Prüfsumme mit privatem Schlüssel verschlüsseln → Authentifizierung beim Empfänger
- Angriffe:
    - Finde eine zweite Nachricht mit demselben Hash-Wert wie die erste
    - Finde zwei (beliebige) Nachrichten mit demselben (beliebigen) Hash-Wert
- Länge: mind. 128 Bit, besser 160 Bit
- Wichtige Algorithmen: SHA-1 (Secure Hash Algorithm), MD5 (Message Digest), Snefru

Integrität (= Unversehrtheit)

- Mit Hash-Funktionen wird eine eindeutige Prüfsumme aus dem Dokument berrechnet (nur diese wird verschlüsselt)
- Bei eifachen Hash-Funktionen kann es Kollisionen geben (zwei Dokumente mit demselben Hash-Wert)
- Anforderung: es darf mit realistischem Aufwand nicht möglich sein, eine Nachricht so zu verändern, dass sie denselben Hash-Wert wie das Original hat

MD5

- Einweg Hash-Funktion, die einen 128-Bit-Wert erzeugt
- Der Eingabetext wird in Blöcke zu 512 Bit zerlegt, die wiederum in 16 Teilblöcke à 32 Bit weiterverarbeitet werden
- Die Ausgabe besteht aus 4 32-Bit-Blöcken, die zusammengesetzt 128 Bit ergeben
- Zu Beginn werden 4 32-Bit-Variablen mit vorgegebenen Werten initialisiert (A=01234567, B=89ABCDEF, C=FEDCBA98, D=76543210)
- In der Hauptschleife werden diese Variablen mit Hilfe der 512-Bit-Datenblöcke modifiziert
- danach werden die nun modifizierten Variablen mit dem nächsten Block verarbeitet

Funktion F

$$F_1(x,y,z) = (x \ _{AND} \ y)_{OR} (\overline{x} \ _{AND} \ z)$$

$$F_2(x,y,z) = (x \ _{AND} \ y)_{OR} (y \ _{AND} \ \overline{z})$$

$$F_3(x,y,z) = x \ _{XOR} \ y \ _{XOR} \ z$$

$$F_4(x,y,z) = y \ _{XOR} (x \ _{OR} \ \overline{z})$$

MAC (= Message Authentification Code, nicht MAC-Adresse)

- zwei Varianten:
    - Hashwert bilden und das Ergebnis verschlüsseln
    - HMAC (Keyed-Hash MAC): Verknüpfung des Schlüssels mit der Nachricht und anschließend Verarbeitung im Hash-Verfahren

Vertrauensmodelle

- Direct Trust (direkter, persönlicher Austausch der Schlüssel) → Problem: Sperrung des Schlüssels (wie bekommt es der andere Nutzer mit?), Verbindlichkeit (Abstreiten, dass eigener Schlüssel verwendet wurde), Policy (Regeln für Schlüssel)
- Web of Trust:
    - Vertrauen
    - Validität
    - dem zertifizierten Schlüssel wird die eigene Unterschrift angehängt
    - Probleme: Sperrung kann nicht jeden User erreichen → evtl. helfen Listen zum Abgleich aus, Verbindlichkeit verbessert sich nur wenig, es kann niemand zu Policy-Anwendung gezwungen werden
- Hierarchical Trust:
    - Verwaltung der Schlüssel bei einer Zertifizierungs-Instanz
    - Trust-Center, Certification Authority CA
    - Infrastruktur erforderlich (PKI)
    - Vertrauenswürdigkeit der CA ???
    - CAs können hierarchisch organisiert sein
    - PKI-Standards: X.509, PKIX, Identrus, OpenPGP

Anwendung der Verschlüsselung im Internet

- TCP/IP-Protokolle verschlüsseln nichts
- in IPv6 sind automatische Verschlüsselungen auch auf unteren Ebenen vorgesehen
- heute verbreitet: Verschlüsselung auf Schicht 7
- Verschlüsselung auf Schicht 1 und 2:
    - ISDN (keine Standards)
    - GSM
    - PPP
- Virtual Private Networks (VPN)
    - Kopplung zweier LANs über das Internet
    - Varianten: Tunneling über PPP oder IPSec
    - Protokolle:
        - L2F (Layer 2 Forwarding Protocol)
        - PPTP (Point-to-Point Tunneling Protocol)
        - L2TP (Layer 2 Tunneling Protocol)

VPN-Ablauf

- Rechner A schickt eine Nachricht an den Router (OSI Schichten von 7-1 werden durchlaufen)
- Router entpackt bis zur Schicht 3 und setzt auf dieser das Tunnel-Protokoll auf
- Danach wird das Paket wieder mit TCP/UPD und IP verpackt und gesendet
- Entsprechend wird auf der Empfängerseite zunächst im Router die Verschlüsselung rückgängig gemacht, bevor an den Empfänger weitergeleitet wird

  ➔ Es liegt bei der Übertragung zwischen den Routern entschlüsselt nur der IP-Header des Routers, das TCP und das Tunnelprotokoll vor, alle weiteren Daten sind verschlüsselt (dazu gehört auch die IP-Adresse des Senderrechners A)

IPSec

- Kommunikation mit anderem Protokoll auf Schicht 3
- Problem: IPSec muss auf beiden Geräten vorhanden sein
- Vorteil der Verschlüsselung auf Schicht 3 → als Benutzer braucht man nicht viel zu tun → Infrastruktur muss allerdings gegeben sein

IPSec ESP-Header (ESP = Encapsulating Security Payload)

… IP Header …
Security Payload Index SPI (Bezeichner für verschiedene Verbindungen)
Sequenz-Nummer
Payload (ggf. Padding) / Next Header (z.B. für Init.-Vektoren)

ISAKMP (Internet Security Association Key Management Protocol)

- Dient zum Austauschen der Schlüssel bei Übertragungen mit IPSec Protokoll

SSL (= Secure Socket Layer)

- SSL ist ein Erweiterungsprotokoll zur Verschlüsselung auf TCP-Ebene
- ursprünglich proprietäre Entwicklung (Netscape), heute als Transport Layer Security TLS standardisiert
- SSL greift nicht direkt in TCP ein, sondern arbeitet zwischen Anwendungen und TCP (es gibt jedoch keine UDP-Variante)
- Vorteil: die Anwendung greift wie üblich auf einen Socket zu
- SSL ist verbindungsorientiert und zustandsbehaftet
- verschiedene Krypto-Verfahren sind aushandelbar
- SSL gliedert sich in zwei Teilschichten:
    - SSL-Record-Protocol
    - weitere Schicht mit Handshake-Protocol,Change-CipherSpec-Protocol,Alert-Protocol, Application-Data-Protocol

Verschlüsselung im WWW

- HTTP-1.0 enthält keinerlei Sicherheits-Protokolle
- es gibt lediglich eine so genannte ”Basic-Authentification“ mit einer Passwort-Abfrage
- Version 1.1 unterst¨ utzt einen Authentifizierungs-Mechanismus
- Diese ”Digest Access Authentification“ erweitert die Passwort-Abfragemit einem Challenge-Response-Verfahren

Remote Login

- die Standard-Protokolle telnet, rlogin, rsh unterstützen keine Verschlüsselung
- die Authentifizierung basiert auf dem in Klartext übertragenen Passwort
- Problem: Abhören des Passworts
- einfachste Möglichkeit: Einweg-Passwörter (S/Key)

Secure Shell SSH

- ssh ist ein sicheres Login-Verfahren zum Ersatz von telnet, rlogin, rsh, rdist
- Unterstützung verschiedener Algorithmen (TripleDES, IDEA, RSA, Diffie-Hellman)
- Funktion: per asymmetrischem Verfahren wird ein Schlüssel ausgetauscht, der dann in einem symmetrischen Verschlüsselungs-Verfahren verwendet wird
- Verwendung von zwei Schlüsseln, die stündlich gewechselt werden