Vorlesung Netzsicherheit

Alice

Bob

Kryptographische Grundlagen: Meet Alice and Bob

Übersicht

- Schutzziele
  - Welche Schutzziele will ich? Wie sind diese definierbar?
- Angriffe
  - Was kann ein Angreifer tun? Wie sieht ein Angreifermodell aus?
- Kryptographische Bausteine
  - Welche Bausteinen habe ich an der Hand um sichere Protokolle zu entwickeln?
- Schlüsselaustausch
  - Wie kann ich Schlüssel über einen unsicheren Kanal aushandeln?
- Perfect Secrecy Properties
  - Welche allgemeinen Prinzipien sind bei Schlüsselprotokollen zu beachten?

Übersicht

- Schutzziele
  - Welche Schutzziele will ich? Wie sind diese definierbar?
- Angriffe
  - Was kann ein Angreifer tun? Wie sieht ein Angreifermodell aus?
- Kryptographische Bausteine
  - Welche Bausteinen habe ich an der Hand um sichere Protokolle zu entwickeln?
- Schlüsselaustausch
  - Wie kann ich Schlüssel über einen unsicheren Kanal aushandeln?
- Perfect Secrecy Properties
  - Welche allgemeinen Prinzipien sind bei Schlüsselprotokollen zu beachten?

Schutzziele

- Ein Schutzziel definiert aus Sicherheitssicht, welche Anforderungen erfüllt werden sollen
  - z.B. *Vertraulichkeit*: übertragene Daten sollen nur berechtigten Instanzen zugänglich sein
- Verschiedene Kategorisierungen
  - CIA Triad
    - *Vertraulichkeit, Integrität, Verfügbarkeit*
  - Parkerian Hexad
    - CIA Triad + *Besitz u. Kontrolle, Authentizität, Nutzen*
  - weitere Schutzziele
    - *Autorisierung, (Nicht-)Abstreitbarkeit, …*

Schutzziele

Ein Schutzziel definiert aus Sicherheitssicht, welche Anforderungen erfüllt werden sollen

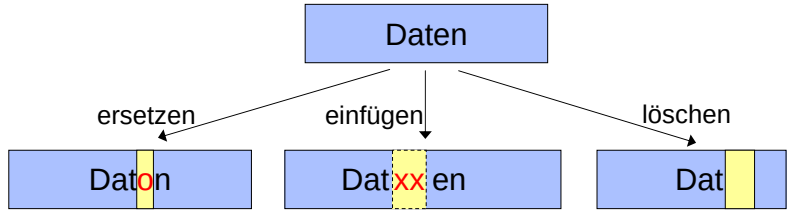
- Diskussion
  - → kann *Verfügbarkeit* sichergestellt werden?

Auswahl von Schutzzielen in der Kommunikation

- Integrität (integrity)
- Vertraulichkeit (confidentiality)
- Authentizität (authenticity)
- Autorisierung (authorization)

Integrität

- Es ist nicht möglich, die zu schützenden Daten unautorisiert und unbemerkt zu manipulieren
- Mögliche Manipulationen z.B.
  - *ersetzen* von Daten
  - *einfügen* in Daten
  - *löschen* von Daten



Integrität

- Schutz der Integrität
  - wie kann die Integrität von Daten sichergestellt werden?
  - schützen Prüfsummen wie CRC die Integrität?
    - z.B. Ethernet Header:

DEST MAC	SOURCE MAC	TYPE	DATA	CRC

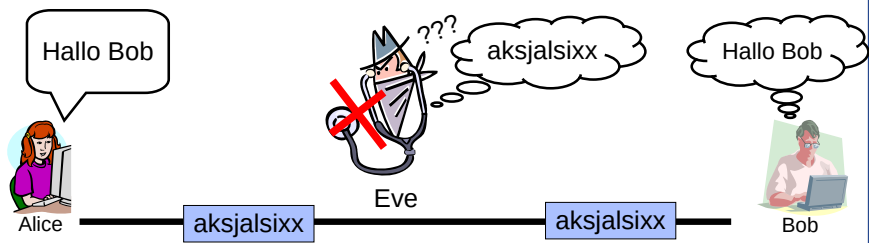
    - Diskussion: schützt der CRC die Integrität des Ethernet Header?

- Methoden zur Realisierung von Integrität
  - Grundidee: Verwendung von Hashfunktionen mit geheimen Schlüsseln (Details später)
  - → Message Authentication Codes (MAC)

Vertraulichkeit

- Übertragene Daten sollen nur berechtigten Instanzen zugänglich sein
  - d.h. es kann kein unautorisierter Informationsgewinn über die Daten stattfinden

- Alice und Bob *kommunizieren vertraulich*



Vertraulichkeit

- Methode zur Realisierung: Verschlüsselung
  - *symmetrische* Verschlüsselung
  - *asymmetrische* Verschlüsselung
- Diskussion: welche Aspekte sind bei Vertraulichkeit der Kommunikation zu beachten?
  - Art der Verschlüsselung?
  - was wird verschlüsselt?
    - Payload, Header?
  - wie werden Schlüssel ausgehandelt?
  - wann werden Schlüssel erneuert?
  - …

Authentizität

- Echtheit und Glaubwürdigkeit von Daten oder Subjekten, die anhand eindeutiger Identität oder charakterisierender Eigenschaften überprüfbar ist
  - Echtheit von *Subjekten*
    - Bob will sicherstellen, dass er wirklich mit Alice redet
  - Echtheit von *Daten*
    - Bob will sicherstellen, dass die Daten wirklich von Alice sind

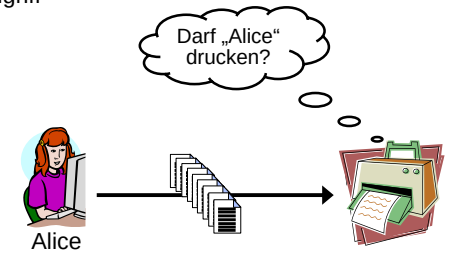
Authentizität

- Methoden zur Realisierung
  - Passwörter
  - Passwort Hashes
  - Einmalpasswörter
  - Signaturen
  - … oft sehr protokollspezifisch

→ viele Möglichkeiten die Authentizität von Daten und Subjekten zu realisieren

Autorisierung

- Es sollen nur autorisierte Instanzen Zugriff auf bestimmte Dienste oder Daten erhalten
  - weitere Einschränkungen möglich, z.B.
    - nur lesender Zugriff
    - lesender und schreibender Zugriff
    - ...
- Methoden zur Realisierung
  - Access Control Lists
  - Autorisierungszertifikate
  - ...
- Diskussion
  - → im welchem Zusammenhang stehen *Autorisierung zu Authentifikation*



Übersicht

- Schutzziele
  - Welche Schutzziele will ich? Wie sind diese definierbar?
- Angriffe
  - Was kann ein Angreifer tun? Wie sieht ein Angreifermodell aus?
- Kryptographische Bausteine
  - Welche Bausteinen habe ich an der Hand um sichere Protokolle zu entwickeln?
- Schlüsselaustausch
  - Wie kann ich Schlüssel über einen unsicheren Kanal aushandeln?
- Perfect Secrecy Properties
  - Welche allgemeinen Prinzipien sind bei Schlüsselprotokollen zu beachten?

Angriffe

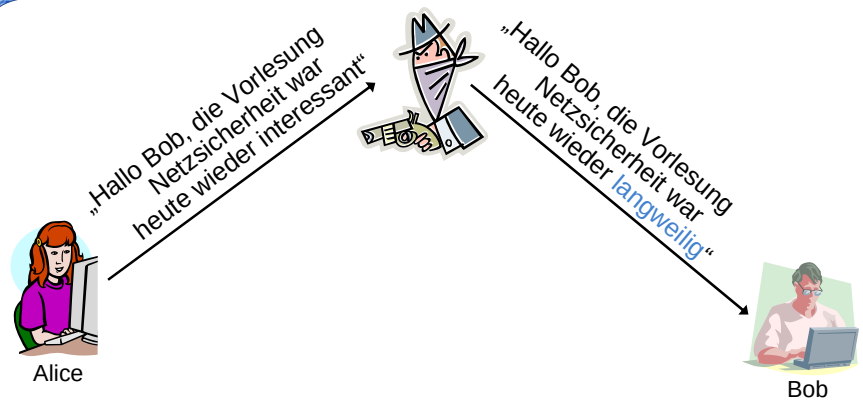
- Generelle Unterscheidung von Angreifern in
  - *aktiv* und *passiv*
    - aktiv: manipulieren, unterdrücken, einfügen, denial-of-service, …
    - passiv: abhören
  - *intelligent* und *blind*
    - intelligent: reagiert, passt sich an, kann sich verstecken, …
    - blind: stupides durchprobieren (brute-force), …
- *Dolev-Yao Angreifermodell* (bekanntestes Angreifermodell)
  - Angreifer kann abhören, unterdrücken, einfügen (an jeder Stelle im Netz!)
  - Angreifer kann aktiv und passiv agieren, ist intelligent
  - Angreifer ist nur durch kryptographische Berechnungen limitiert
  - *network is the attacker*, sehr starkes Modell → wird meist vereinfacht verwendet

Abhören

Manipulieren

Manipulieren

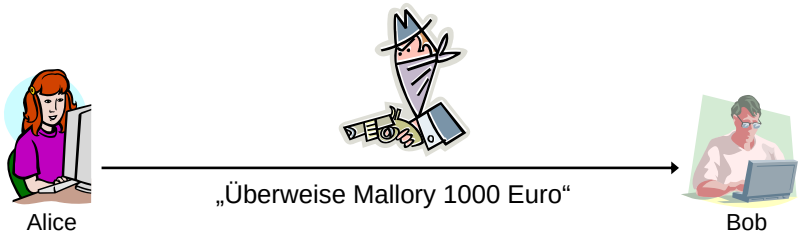
Man in the middle



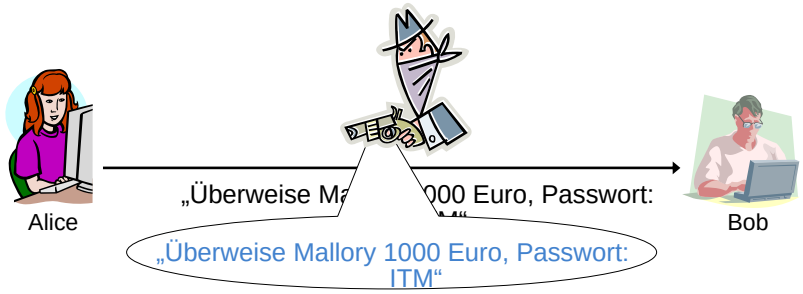
Unterdrücken

Einfügen

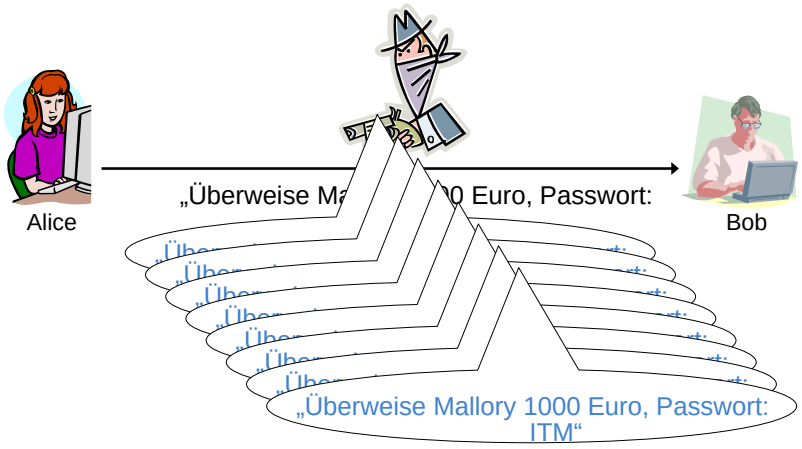
Replay Attacke



Replay Attacke

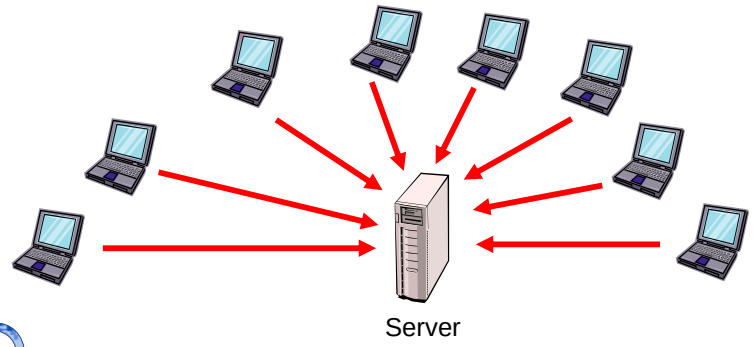


Replay Attacke



(Distributed) Denial of Service

- *Denial-of-Service* (DoS) Angriff
  - Einschränkung der Verfügbarkeit eines Dienstes
- *Distributed-Denial-of-Service* Angriff (DDoS)
  - DoS-Angriff durch verteilte Angreifer

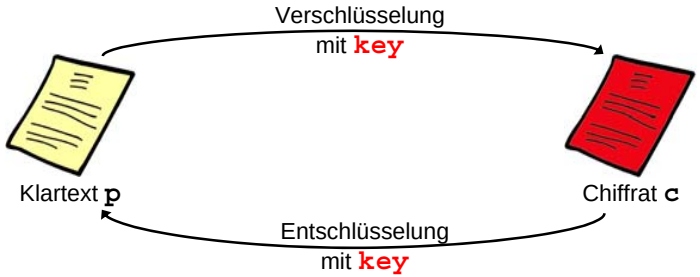


Übersicht

- Schutzziele
  - Welche Schutzziele will ich? Wie sind diese definierbar?
- Angriffe
  - Was kann ein Angreifer tun? Wie sieht ein Angreifermodell aus?
- Kryptographische Bausteine
  - Welche Bausteinen habe ich an der Hand um sichere Protokolle zu entwickeln?
- Schlüsselaustausch
  - Wie kann ich Schlüssel über einen unsicheren Kanal aushandeln?
- Perfect Secrecy Properties
  - Welche allgemeinen Prinzipien sind bei Schlüsselprotokollen zu beachten?

Symmetrische Verschlüsselung

- Gemeinsames Geheimnis der Kommunikations-partner → gemeinsamer Schlüssel **key**
- Gemeinsamer Schlüssel **key** zum
  - verschlüsseln: $c = E_{key}(p)$
  - entschlüsseln: $p = D_{key}(c)$



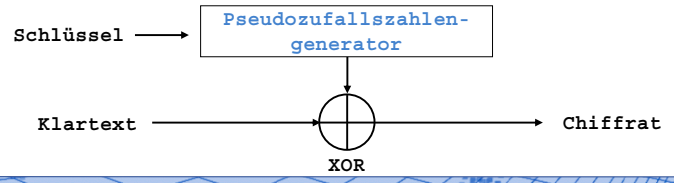
Symmetrische Verschlüsselung

- Grundsätzliche Arten von Chiffren
  - *Blockchiffren*
    - Blockweises Verschlüsseln der Daten
    - gängige Blockchiffren: AES, DES, 3DES, …
  - *Stromchiffren*
    - Bitweises (bzw. Zeichenweises) Verschlüsseln der Daten
    - muss nicht warten bis ein Block von Daten bereit steht (daher für Echtzeitübertragung geeignet, z.B. im Mobilfunk verwendet)
- Diskussion: welches Problem tritt auf, wenn man symmetrische Verschlüsselung in Netzen einsetzt?

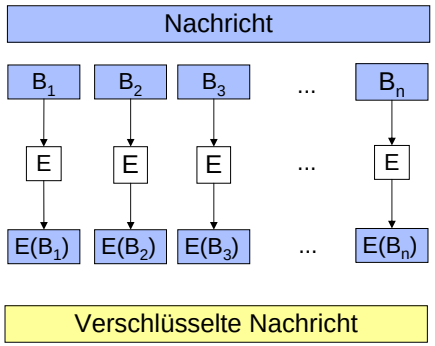
Kennen Sie Anwendungen von symmetrischer Kryptographie?

Symmetrische Verschlüsselung: Stromchiffren

- Stromchiffren operieren Zeichenweise
  - Strom von Schlüssel-Zeichen, von zu verschlüsselnden Zeichen
  - Funktion (z.B. **XOR**) verknüpft beide Ströme zeichenweise
  - Verschlüsselung: $c_i = p_i \text{ XOR } k_i$
  - Entschlüsselung: $p_i = c_i \text{ XOR } k_i$
- Verwendung von Pseudozufallszahlenfolge
  - Eingabe: kurzer Initialisierungswert ← *gemeinsamer Schlüssel*
  - Ausgabe: Folge von Zeichen, die
    - mittels *deterministischen* Prozesses gewonnen werden
    - gewisse Eigenschaften einer echt zufälligen Folge aufweisen



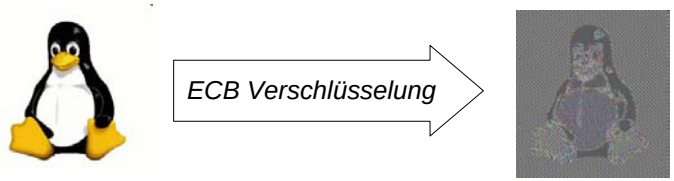
Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren



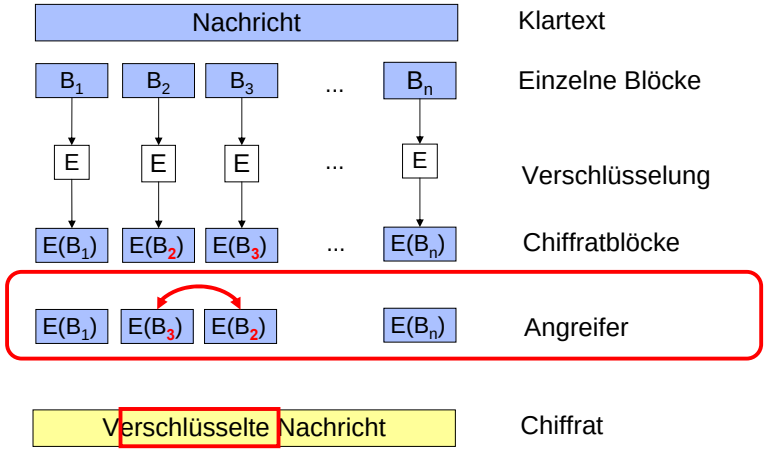
Welche Probleme können dadurch entstehen, dass die *Nachricht in Blöcke aufgeteilt* ist? Was kann ein Angreifer tun?

Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren

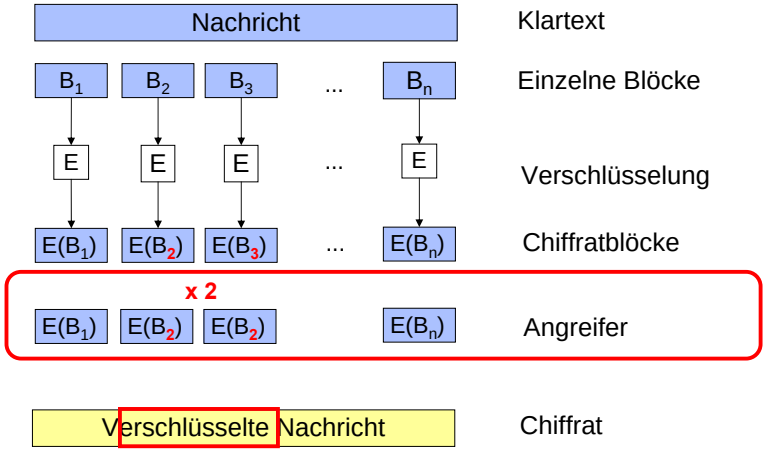
- *Electronic Codebook Mode* (ECB)
  - Problem
    - Blöcke werden einzeln, unabhängig voneinander verschlüsselt
    - *Struktur des Cyphertext ähnlich Struktur des Plaintext*



Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren



Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren

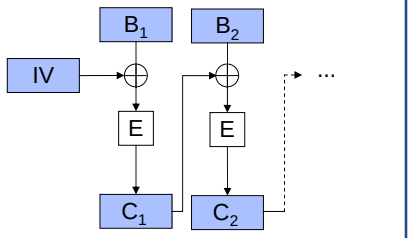


→ *Manipulationen können nicht erkannt werden!*

Symmetrische Verschlüsselung : Betriebsmodus

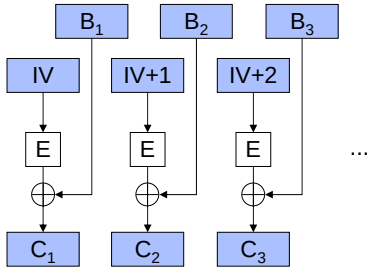
- Lösung: Blöcke abhängig voneinander machen
  - verschiedene Möglichkeiten realisiert als *Betriebsmodi*

Cipher Block Chaining (CBC)



Verschlüsselung

Counter Mode (CTR)



Verschlüsselung

Diskussion: wie sieht die Entschlüsselung bei CBC aus?

Symmetrische Verschlüsselung : Betriebsmodus

- Struktur durch verschiedene Betriebsmodi

Plaintext

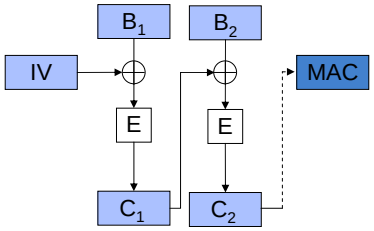
Electronic Codebook Mode (ECB)

Cipher Block Chaining Mode (CBC)

Symmetrische Verschlüsselung: Kombinierte Betriebsmodi

Verschiedene Arten von *Authentifizierter Verschlüsselung*

- Counter-Mode mit CBC-MAC
  - Verschlüsselung mit CTR
  - *Authentifizierung durch CBC-MAC*
  - 2 Runden, ineffizient



- Galois/Counter Mode
  - Verschlüsselung mit CTR
  - Authentifizierung mit Galois-Polynom GF(2^{128})
- Andere
  - OCB (patentiert), EAX, CWC, …

Asymmetrische Verschlüsselung

- Asymmetrische Verschlüsselung
  - *öffentlicher Schlüssel* (bekannt)
  - *privater Schlüssel* (geheim)
- Ver- und Entschlüsselung
  - Verschlüsselung mit dem öffentlichen Schlüssel des Empfängers durch Absender
  - Entschlüsselung mit dem privaten Schlüssel des Empfängers durch Empfänger

Was sind die Vorteile gegenüber symmetrischer Verschlüsselung?

Asymmetrische Verschlüsselung: Schreibweise

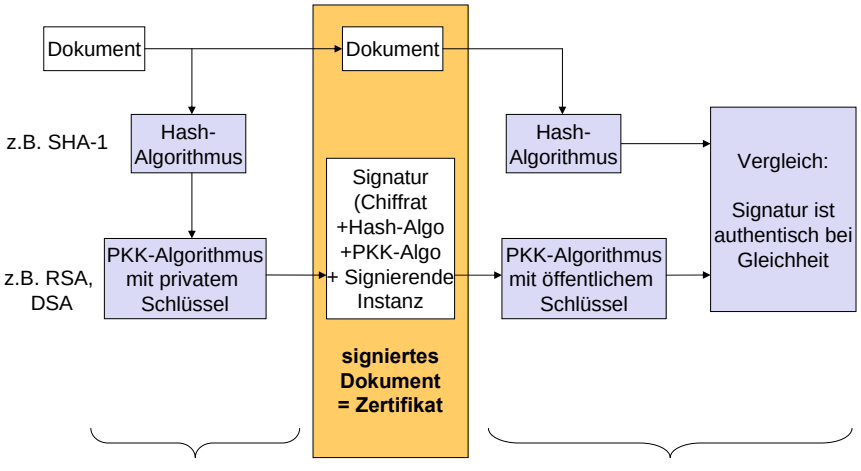
- Kennzeichnung als öffentlich oder privater Schlüssel
  - $pubKey_{Alice}$ oder pub_{Alice}
  - $privKey_{Alice}$ oder $priv_{Alice}$
- Operationen (Bob sendet Nachricht an Alica)
  - Verschlüsselung: $c = E_{pub_{Alice}}(p)$
  - Entschlüsselung: $p = D_{priv_{Alice}}(c)$
- Gebräuchliche Verfahren: RSA, El Gamal, …

Kennen Sie Anwendungen asymmetrischer Verschlüsselung?

Asymmetrische Verschlüsselung: Signatur

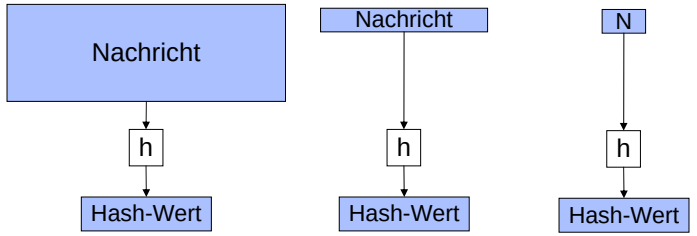
- Signatur von Nachrichten mit privatem Schlüssels
  - $Signatur = Sig_{priv_{Alice}}(Nachricht)$
- Verifizieren der Signatur mit öffentlichem Schlüssel
  - $Ver_{pub_{Alice}}(Nachricht, Signatur)$
- Performance
  - asymmetrische Kryptographie ist viel langsamer als symmetrische Kryptographie
  - wird selten ganze Nachricht signiert
    - → Message Digest bilden und diesen signieren
  - hybride Kryptosysteme
    - symmetrischen Sitzungsschlüssel über asymmetrische Kryptographie aushandeln, Datenfluss mit symmetrischem Sitzungsschlüssel schützen

Asymmetrische Verschlüsselung: Digitale Signatur



Hash-Funktion

- Kryptographische Einweg-Funktion $n \rightarrow h(n)$
  - Umkehrung schwierig
    - zu gegebenem $h(n)$ ein n finden
  - Kollision schwierig
    - n_1 und n_2 finden, so dass $h(n_1) = h(n_2)$
- bildet Daten beliebiger Länge auf Bitstring fester Länge ab



Hash-Funktion

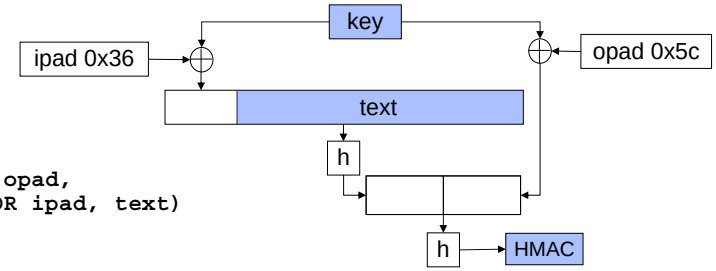
- Hash-Funktionen: SHA-1, MD5, …
  - `sha1('Netzsicherheit')=1e13980291f5a113817610ec8ef94858e0bf90b5` **(160bit)**
  - `md5('Netzsicherheit')=12fa7645a6cf63baceceaad2c844efaf` **(128bit)**
- Probleme
  - auf SHA-1 und MD5 gibt es (theoretische) Angriffe
    - SHA-2 ist ähnlich zu SHA-1, daher keine dauerhafte Alternative
    - aktuell Review Prozess im SHA-3 Contest von NIST (http://csrc.nist.gov/groups/ST/hash/sha-3)
  - Angriffe über Vorberechnungen (Rainbow Table)

Kennen Sie Anwendungen von Hash-Funktionen?

Message Authentication Code: HMAC

- HMAC: Keyed-Hashing for Message Authentication
  - verwendet zur Integritätssicherung
  - nur wer geheimen Schlüssel *key* kennt kann
    - authentische Nachrichten erzeugen
    - Authentizität von Nachrichten prüfen

```
HMAC = H(
    key XOR opad,
    H(key XOR ipad, text)
    )
```



Übersicht

- Schutzziele
  - Welche Schutzziele will ich? Wie sind diese definierbar?
- Angriffe
  - Was kann ein Angreifer tun? Wie sieht ein Angreifermodell aus?
- Kryptographische Bausteine
  - Welche Bausteinen habe ich an der Hand um sichere Protokolle zu entwickeln?
- Schlüsselaustausch
  - Wie kann ich Schlüssel über einen unsicheren Kanal aushandeln?
- Perfect Secrecy Properties
  - Welche allgemeinen Prinzipien sind bei Schlüsselprotokollen zu beachten?

Diffie-Hellman-Schlüsselaustausch

- Schlüsselaustausch über unsicheren Kanal
  - Alice und Bob wollen einen Schlüssel **S** austauschen
  - **g** und **n** müssen gewählt werden
  - Alica wählt Zufallszahl **a**, Bob wählt Zufallszahl **b**

wähle **a**

Alice

$g^a \bmod n$

$g^b \bmod n$

wähle **b**

Bob

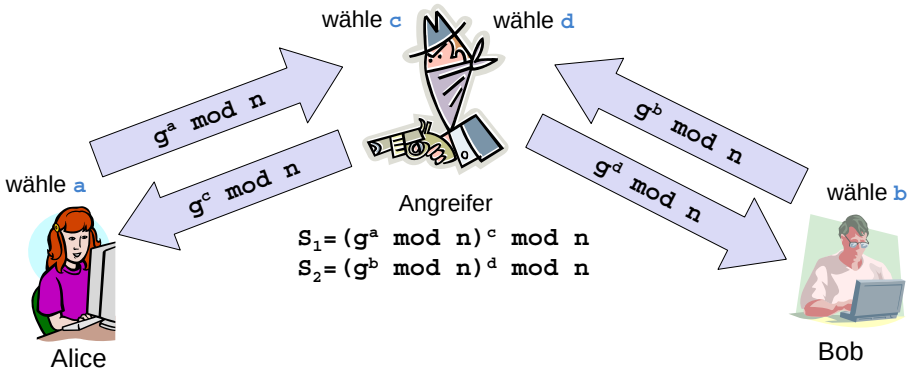
bekannt: **a**, $g^b \bmod n$, **n**
berechne: $S = (g^b \bmod n)^a \bmod n$

bekannt: **b**, $g^a \bmod n$, **n**
berechne: $S = (g^a \bmod n)^b \bmod n$

Angreifer: $g^a \bmod n, g^b \bmod n, g, n$
→ es fehlt **a** oder **b** zur Berechnung von S!

Angriff auf Diffie-Hellman

- Man-in-the-Middle Angriff auf Diffie-Hellman



Diskussion: Wie kann man diesen Angriff verhindern?

Übersicht

- Schutzziele
  - Welche Schutzziele will ich? Wie sind diese definierbar?
- Angriffe
  - Was kann ein Angreifer tun? Wie sieht ein Angreifermodell aus?
- Kryptographische Bausteine
  - Welche Bausteinen habe ich an der Hand um sichere Protokolle zu entwickeln?
- Schlüsselaustausch
  - Wie kann ich Schlüssel über einen unsicheren Kanal aushandeln?
- Perfect Secrecy Properties
  - Welche allgemeinen Prinzipien sind bei Schlüsselprotokollen zu beachten?

Perfect Secrecy

- Unabhängigkeit von Schlüsseln
  - durch langlebige Geheimnisse werden dynamische Sitzungsschlüssel erzeugt
  - Offenlegung eines langlebigen Geheimnis darf keine alten Sitzungsschlüssel verwundbar machen
  - aus einem Sitzungsschlüssel darf kein früherer oder zukünftiger Sitzungsschlüssel abgeleitet werden können
- Zwei Perfect Secrecy Eigenschaften
  - *Perfect Forward Secrecy*
    - Angreifer kann keine zukünftigen Nachrichten von Session *n+1* lesen wenn er in Besitz des Sitzungsschlüssel für Session *n* kommt
  - *Perfect Backward Secrecy*
    - Angreifer kann keine alten Nachrichten lesen von Session *n-1* lesen wenn er in Besitz des Sitzungsschlüssel für Session n kommt
- Methode zur Realisierung
  - Aushandlung von Sitzungsschlüsseln z.B. über Diffie-Hellman und Authentifizierung über langlebiges Geheimnis

Literatur

- *Sichere Netzwerkkommunikation*, Bless et al., Springer, 2005.
- *Encyclopedia of Cryptography and Security*, Tilborg, Springer, 2005.
- *IT-Sicherheit, Konzepte, Verfahren, Protokolle*, Eckert, Oldenbourg Verlag, 2003.
- *Applied Cryptography*, Schneier, Wiley, 1995.
- *Practical Cryptography*, Schneier, Wiley, 2003.
- *Handbook of Applied Cryptography*, CRC, 1996. http://www.cacr.math.uwaterloo.ca/hac/