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

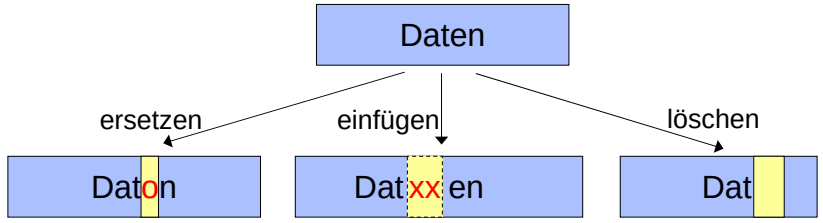
- Diskussion
  - → kann *Verfügbarkeit* sichergestellt werden?

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



•Folgende Manipulationen einer Dateneinheit sind z.B. möglich:

- •Ersetzen: Teile oder die gesamte Dateneinheit werden ersetzt. Im Beispiel auf der Folie wird das „e“ aus „Daten“ durch ein „o“ ersetzt.
- •Einfügen: In die Dateneinheit werden weitere Teile eingesetzt. Im Beispiel auf der Folie werden die Zeichen „xx“ eingefügt
- •Löschen: Teile oder die gesamte Dateneinheit werden gelöscht. Im Beispiel auf der Folie wird ein Teil der Dateneinheit (das „e“) gelöscht.

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)

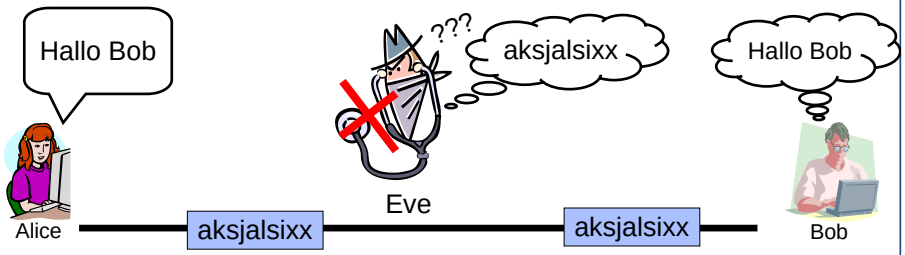
  → Message Authentication Codes (MAC)

•CRC schützt die Datenintegrität NICHT, da ein Angreifer sowohl die Daten als auch die CRC-Prüfsumme ändern kann. Bei Message Authentication Codes wird dies z.B. dadurch verhindert, dass ein Schlüssel in die Berechnung des MAC einfließt (siehe später).

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*



•Im Rahmen dieser Vorlesung werden wir oft „Alice“ und „Bob“ als Synonym für zwei miteinander kommunizierende Instanzen verwenden. Außerdem werden wir im Folgenden einen Angreifer oft mit den Namen „Eve“ oder „Mallory“ bezeichnen.

•Im Beispiel auf der Folie sind Alice und Bob autorisiert, die Daten ihrer Kommunikation zu kennen, während Mallory als Angreifer dies nicht dürfen soll. Alice und Bob setzten ein Verfahren ein, um Vertraulichkeit zu gewährleisten, weshalb nach obiger Definition Mallory keinen Informationsgewinn über die Daten der Kommunikation erhalten kann.

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?
  - …

•Die Vertraulichkeit von Daten bezieht sich wie hier definiert eventuell nur auf die Nutzdaten, aber unter Umständen nicht auf die Steuerungsinformationen. Beim Design von Kommunikationssystemen muss darauf geachtet werden, welche Daten jeweils zu schützen sind.

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

•Im Beispiel ist Alice das Subjekt. Die Authentifizierung erfolgt mittels Benutzername und Passwort, wobei die charakterisierende Eigenschaft zum Nachweis der Identität beispielsweise die Kenntnis des entsprechenden Passworts ist.

•Anderes Beispiel

•Werden Daten über ein unsicheres Netzwerk (z.B. Internet) übertragen, so können kryptographische Verfahren eingesetzt werden, um die Echtheit der Daten zu überprüfen.

Authentizität

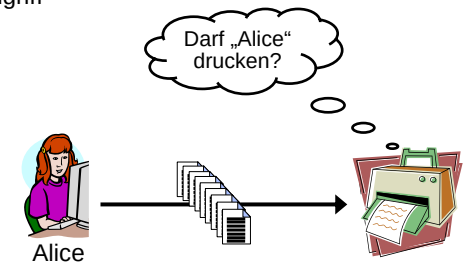
- Methoden zur Realisierung
  - Passwörter
  - Passwort Hashes
  - Einmalpasswörter
  - Signaturen
  - … oft sehr protokollspezifisch

→ viele Möglichkeiten die Authentizität von Daten und Subjekten zu realisieren

•Bei dem Schutzziel Authentizität unterscheidet man den Identitätsnachweis (=Authentizität einer Identität) von der Authentizität von Daten.

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*



•Aus dem Beispiel wird deutlich, dass oft für Autorisierung eine Authentifikation notwendig ist.

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

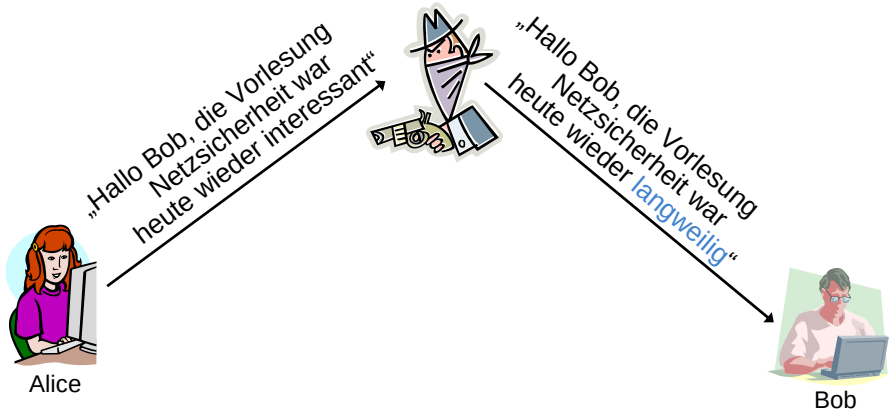
- Generelle Unterscheidung von Angreifern in
  - *aktiv* und *passiv*
    - aktiv: manipulieren, unterdrücken, einfügen, denial-of-service, …
    - passiv: abhören
  - *intelligent* und *blind*
    - intelligent: reagiert, passt sich an, kann sich verstecken, …
    - blind: stupides durchprobieren (brute-force), …
- *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



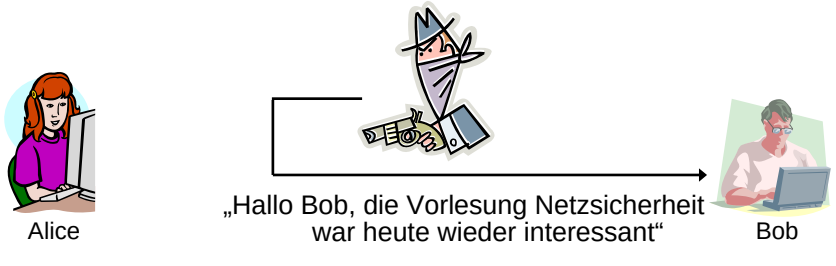
•Als Man-in-the-Middle-Angriff bezeichnet man einen Angriff, bei dem sich ein Angreifer, Mallory, zwischen die beiden Kommunikationspartner schaltet. Gegenüber Alice gibt Mallory vor Bob zu sein, wohingegen Mallory gegenüber Bob sich als Alice ausgibt.

•Über einen Man-in-the-Middle Angriff können die nötigen Gegebenheiten geschaffen werden, um Daten zu Manipulieren, soweit nicht spezielle Vorkehrungen gegen Man in the Middle Angriffe getroffen wurden.

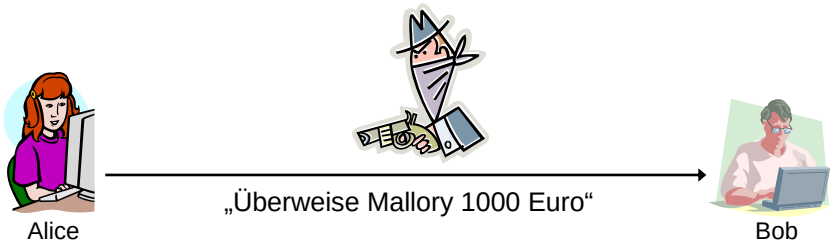
Unterdrücken

•Beim Unterdrücken von Nachrichten verhindert der Angreifer, dass eine Nachricht, die Alice losgeschickt hat, bei Bob ankommt. Z.B. über Man-in-the-Middle realisierbar

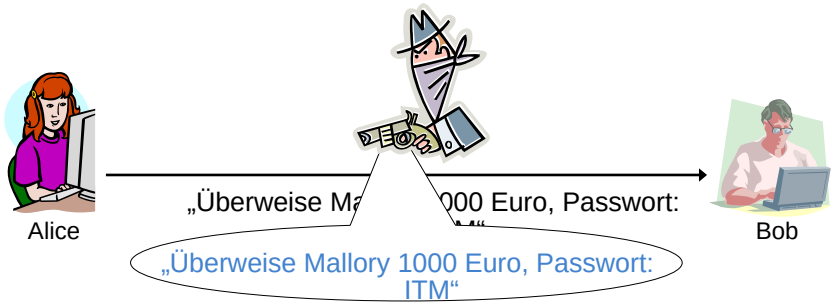
Einfügen



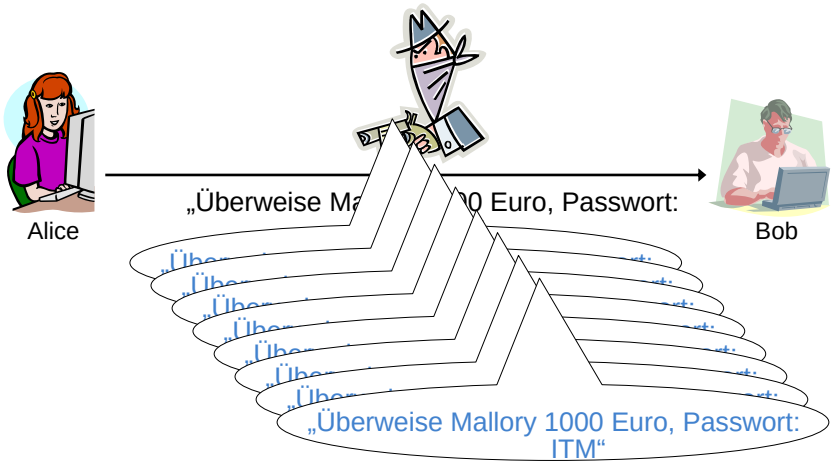
Replay Attacke



Replay Attacke



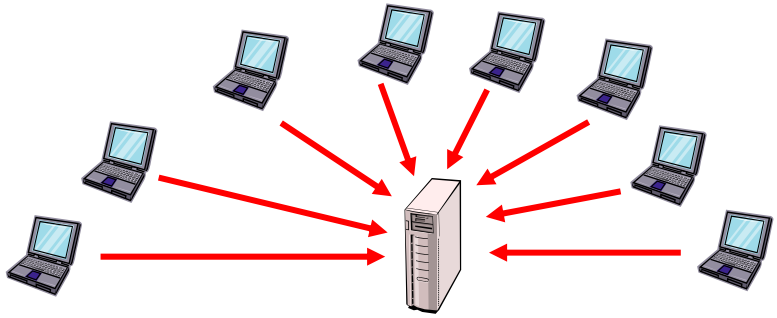
Replay Attacke



•Bei einer Replay-Attacke sendet ein Angreifer Daten erneut, die er zuvor abgehört hat.

(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

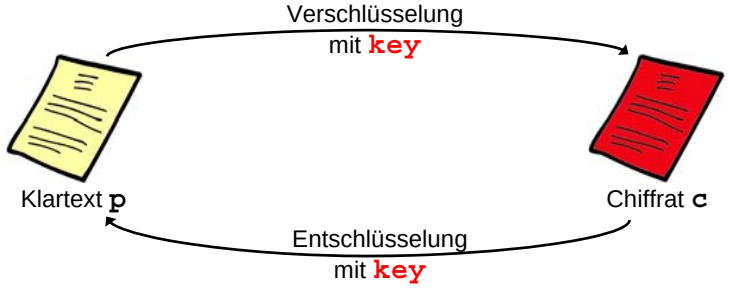


Übersicht

- 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: $p = D_{key}(c)$



•Mit Verschlüsselung kann z.B. Vertraulichkeit von Daten erreicht werden.

•Schreibweise: c=Ekey(p) bzw. m=Dkey(c). Dabei steht E für Encryption (Verschlüsselung) und D für Decryption (Entschlüsselung). p steht für plaintext (Klartext) während c für ciphertext (Chiffrat) steht.

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?

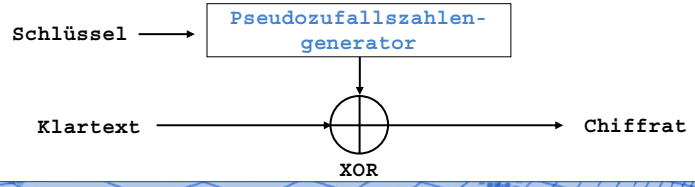
•Stromchiffren arbeiten auf einzelnen Zeichen, während Blockchiffren eine Menge von Zeichen, einen so genannten Block, als Eingabe haben und diesen in einem Durchgang verarbeiten. Gängige Chiffren sind AES, DES und 3DES („Tripple-DES“).

•Eines der Probleme der symmetrischen Verschlüsselung ist, dass pro Kommunikationspartner ein (geheimer) Schlüssel benötigt wird. Sind in einem Netz n Instanzen vorhanden, die alle miteinander kommunizieren wollen, so benötigt jeder Knoten n-1 Schlüssel. Die Verteilung dieser n-1 Schlüssel kann unter Umständen ein Problem darstellen (siehe später).

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



•Pseudozufallszahlengeneratoren generieren für den gleichen Initialisierungswert (Seed) immer die gleiche Folge von Zahlen. Daher ist der Prozess deterministisch und beide Parteien erzeugen die gleiche Folge von Zufallszahlen.

•Da es bei Stromchiffren zu keinerlei Übertrag kommt, ist die Operation sehr gut parallelisierbar.

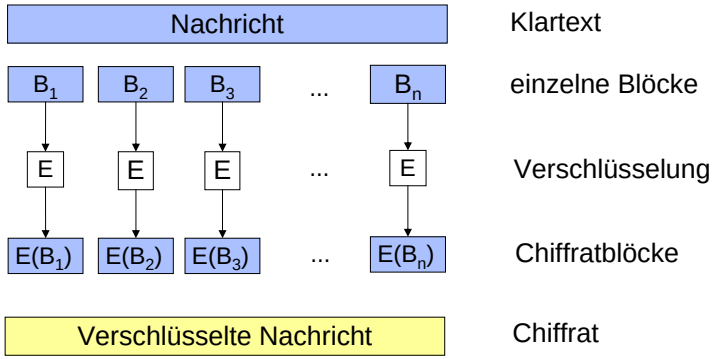
•Auf Grund der Eigenschaften des XOR-Operators kann auf der Empfängerseite durch die gleiche Zufallszahlenfolge der Plaintext rekonstruiert werden.

•Die Qualität der Zufallszahlen ist für die Sicherheit des Verfahrens entscheidend.

•Eine spezielle Form der Stromchiffre ist der One-Time-Pad. Hierbei ist die Folge der Zufallszahlen genauso lange wie die Nutzdatenfolge. Wenn mit echten Zufallszahlen gearbeitet wird, so ist dies die einzige für den Kryptoanalytiker echt beweisbar sichere Verschlüsselung. Diese Sicherheit ist jedoch nur theoretisch und praktische werden One-Time-Pads wenig eingesetzt. Einmal ist es sehr schwer echte Zufallszahlen zu erzeugen, weiterhin wird nicht nur ein Schlüssel benötigt, sondern der komplette Pad, welcher genauso lang ist wie der Klartext, muss Sender und Empfänger bekannt sein.

•Die Stromchiffren RC4 wurden in WEP eingesetzt.

Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren



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

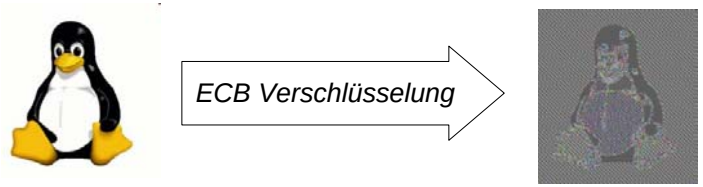
•Da Blockchiffren bei gleichem Schlüssel und gleichem Klartext auch immer denselben Chiffretext erzeugen, kann

- •ein aktiver Angreifer Replay Angriffe durchführen indem er chiffrierte Nachrichtenblöcke aufzeichnet und diese später in eine andere Übertragung einspielt
- •ein passiver Angreifer Wiederholungen erkennen und hieraus Schlussfolgerungen durch Analyse ziehen. Die “Struktur” des Ciphertext ist dem des Plaintext sehr ähnlich.

•Der Dargestellte Modus wird Electronic Codebook Mode (ECB) genannt.

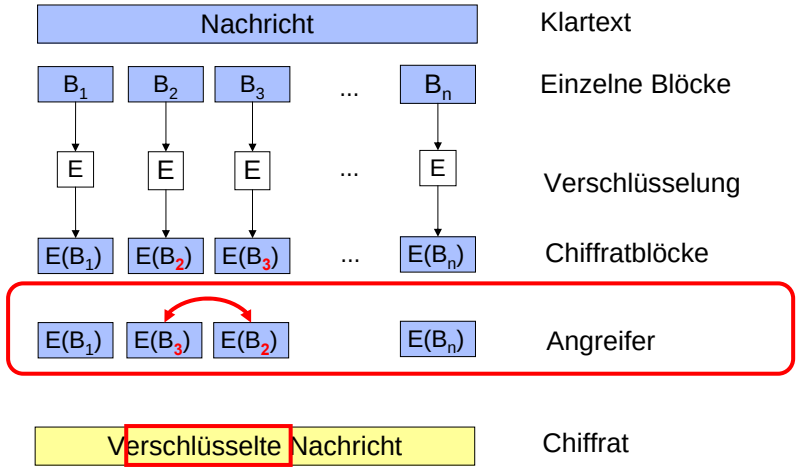
Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren

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

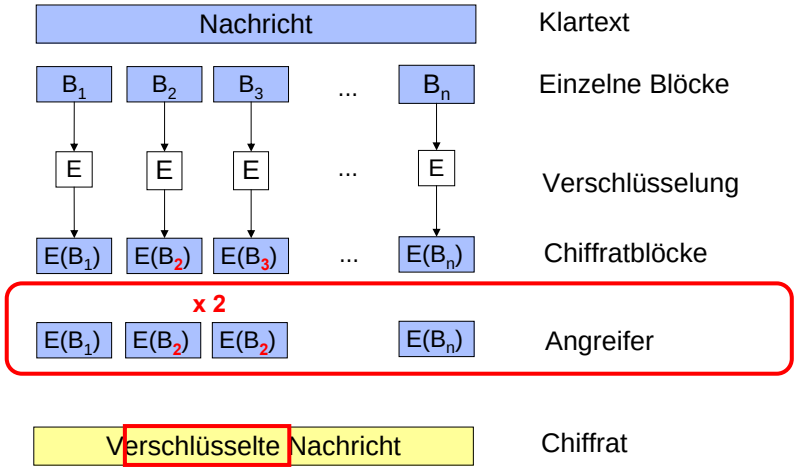


•Problem des ECB Modus ist, dass die Struktur des Plaintext im Cyphertext erhalten bleibt. Hierdurch kann ein Angreifer auf Grund der Struktur Aussagen über den Plaintext machen und Angriffe des Mustererkennung durchführen.

Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren



Symmetrische Verschlüsselung: Betriebsmodus bei Blockchiffren

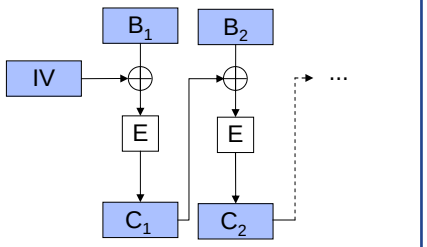


→ *Manipulationen können nicht erkannt werden!*

Symetrische Verschlüsselung : Betriebsmodus

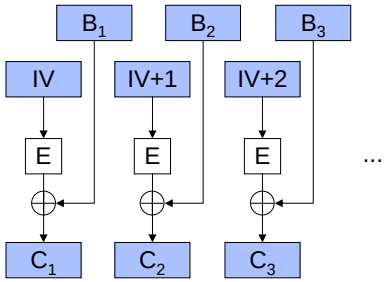
- Lösung: Blöcke abhängig voneinander machen
  - verschiedene Möglichkeiten realisiert als *Betriebsmodi*

Cipher Block Chaining (CBC)



Verschlüsselung

Counter Mode (CTR)



Verschlüsselung

Diskussion: wie sieht die Entschlüsselung bei CBC aus?

•Der Kreis mit dem Pluszeichen steht für die logische Operation „bitweises XOR“.

•Im CBC Mode wird der Ciphertext C1 durch XOR mit dem Plaintext Block B2 kombimiert. Hierdurch entsteht eine Abhängigkeit zwischen den Blöcken.

- •Verschlüsseung: Ci = E(Bi XOR Ci-1)
- •Entschlüsselung: Bi = Ci-1 XOR D(Ci)
- •Da immernoch die Möglichkeit besteht, dass gleiche Plaintext Blöcke auf gleiche Ciphertext Blöcke abgebildet werden, wird ein zufälliger Initialisierungsvektor IV zur Initialisierung verwendet.

Symetrische Verschlüsselung : Betriebsmodus

- Struktur durch verschiedene Betriebsmodi

Plaintext

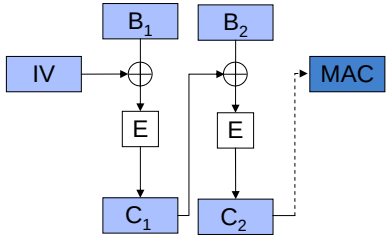
Electronic Codebook Mode (ECB)

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, …

•Counter Mode mit CBC-MAC ist ein Betriebsmodue zur Authentifizierung und Verschlüsselung

•Counter-Mode ist für Geheimhaltung verantwortlich

•CBC-MAC für Integrität und Authentifizierung (s. Abbildung). Hier werden die Ciphertext Blöcke Ci nicht weiterverwendet, sondern nur die Verkettung verwendet. Sie ergibt den MAC Wert.

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?

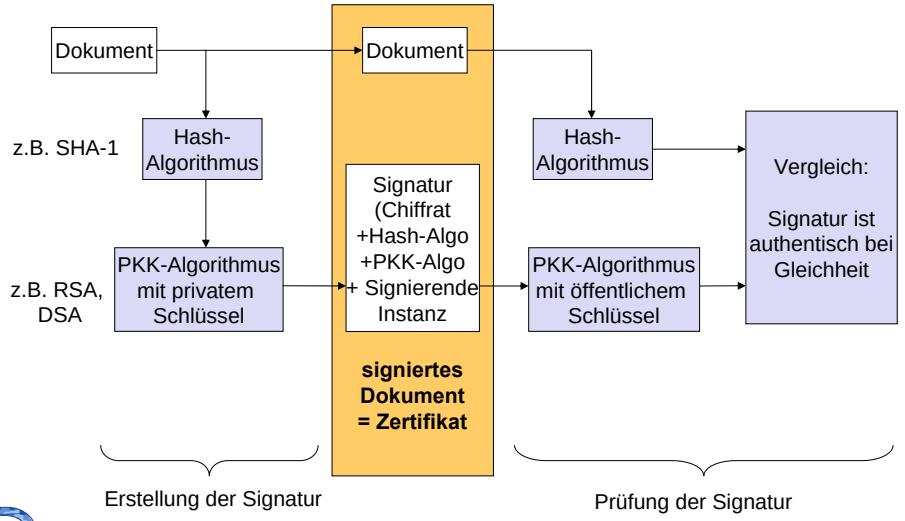
•E steht für Encryption und D für Decryption

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

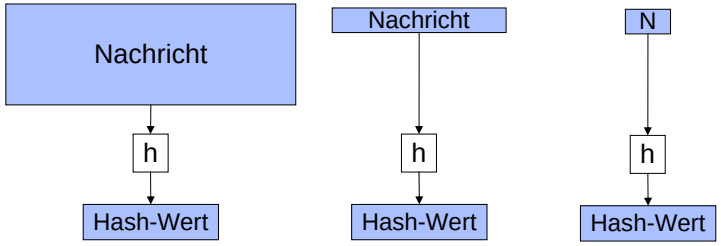
•Bitte beachten Sie, welcher Schlüssel (öffentlicht/privat) jeweils beim Signieren und Verifizieren von Signaturen bzw. beim Verschlüsseln und Entschlüsseln verwendet wird!

Asymmetrische Verschlüsselung: Digitale Signatur



Hash-Funktion

- Kryptographische Einweg-Funktion $\mathbf{n \rightarrow h(n)}$
  - Umkehrung schwierig
    - zu gegebenem $\mathbf{h(n)}$ ein $\mathbf{n}$ finden
  - Kollision schwierig
    - $\mathbf{n}_1$ und $\mathbf{n}_2$ finden, so dass $\mathbf{h(n_1) = h(n_2)}$
- bildet Daten beliebiger Länge auf Bitstring fester Länge ab



•Brechen Sie einen MD5-Hash hier: http://md5.rednoize.com/

Hash-Funktion

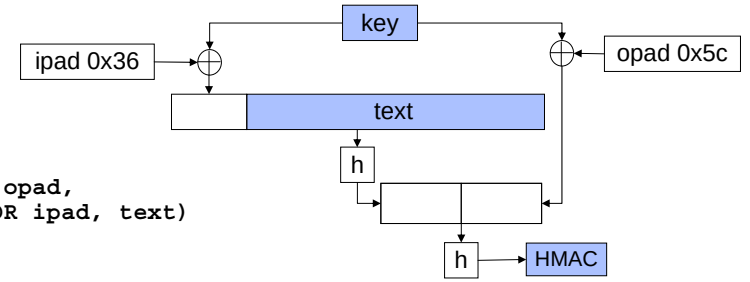
- Hash-Funktionen: SHA-1, MD5, …
  - `sha1('Netzsicherheit')=1e13980291f5a113817610ec8ef94858e0bf90b5` **(160bit)**
  - `md5('Netzsicherheit')=12fa7645a6cf63baceceaad2c844efaf` **(128bit)**
- Probleme
  - auf SHA-1 und MD5 gibt es (theoretische) Angriffe
    - SHA-2 ist ähnlich zu SHA-1, daher keine 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?

•Brechen Sie einen MD5-Hash hier: http://md5.rednoize.com/

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)
    )
```


•ipad und opad sind Konstanten, die im Standard definiert werden.

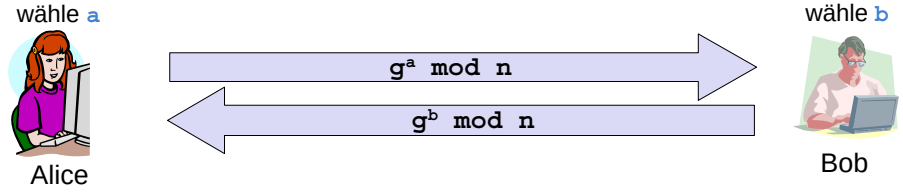
•h ist eine beliebige Hash-Funktion

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**



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

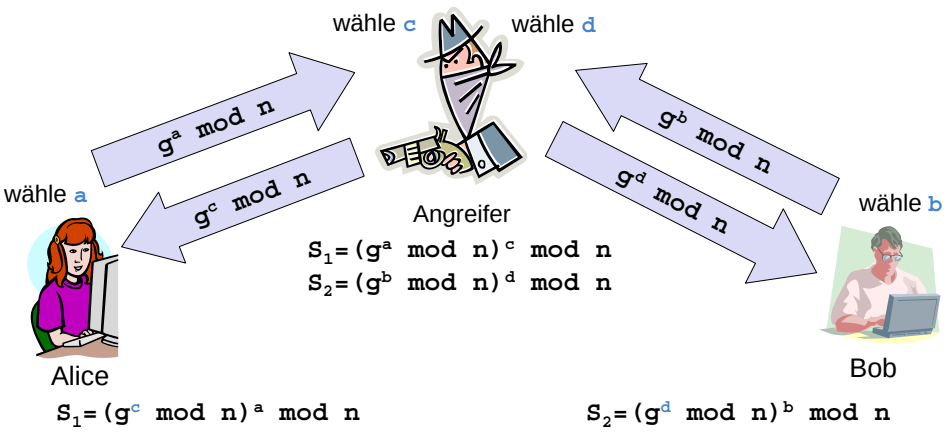
bekannt: **b**, $g^a \bmod n$, **n**
berechne: $S = (g^a \bmod n)^b \bmod n$

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

•Zusatzinfo
  •g nennt man Generator
  •n ist der Modulus, eine große Primzahl
•Die Sicherheit des Verfahrens beruht darauf, dass es sehr schwierig ist den diskreten Logarithmus in einem Primzahlkörper zu berechnen (d.h. es ist schwierig b=log gb mod n zu berechnen)
•Berechnung
  •Alice berechnet S = (g^b mod n)^a mod n = g^(ba) mod n = S
  •Bob berechnet S = (g^a mod n)^b mod n = g^(ab) mod n = S

Angriff auf Diffie-Hellman

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



Diskussion: Wie kann man diesen Angriff verhindern?

•Bei Authentifiziertem Diffie-Hellman ist dieses Problem nicht mehr vorhanden.

•Es gibt verschiedene Verfahren, wie der DH Austausch z.B. mit asymmetrischen Schlüsselpaaren und Signaturen arbeiten. Hierbei sind die Signaturen von einer höheren Instanz ausgestellt, der sowohl Alice als auch Bob vertrauen.

•Verfeinfacht gesagt, wird der DH Austausch signiert, und damit die Authentizität des Kommunikationspartners realisiert.

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/