Methoden der Kryptographie

- Geheime Schlüssel sind die Sicherheitsgrundlage
  - Der Algorithmus ist bekannt
    - außer bei „security by obscurity“
- Public-Key-Cryptography
  - verwendet zwei Schlüssel
  - ein geheimer und ein öffentlicher Schlüssel
- Symmetrische Kryptographie
  - beide Seiten verwenden den selben geheimen Schlüssel
- Hash-Funktion
  - Ohne Schlüssel und ohne Geheimnis

Chiffrierungstypen

- Stromchiffrierer (stream cipher)
  - verschlüsselt bitweise
- Blockchiffre, Blockverschlüsselung (block ciphers)
  - Originaltext wird in gleichgroße Blöcke unterteilt
  - Jeder Block wird einzeln kodiert

Stream Ciphers

- Kombiniere jedes Bit eines Schlüsselstroms (key stream) mit dem Original bit
  - $m(i)$ = i-tes Bit der Nachricht
  - $ks(i)$ = i-tes Bit des Key Streams
  - $c(i)$ = i-tes bit des verschlüsselten Texts
- Verschlüsselung
  - $c(i) = ks(i) + m(i) \pmod 2$
    $= ks(i) \oplus m(i)$
- Entschlüsselung
  - $m(i) = ks(i) \oplus c(i)$

Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

RC4 Stream Cipher

- RC4 ist ein populärer Streamchiffrierer
  - ~~ausführlich analysiert und als sicher angesehen~~
  - ~~kann in SSL verwendet werden~~
    - 2.2015: RC4 im Rahmen von Transport Layer Security (TLS, früher: Secure Sockets Layer, SSL) verboten
  - Schlüssellänge: von 1 bis 256 Bytes
  - wird in SSH1, WEP/WPA für 802.11, … verwendet

Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

Quell-Code RC4

```
k[]: gegebene Schlüssel-Zeichenfolge der Länge 5 bis 256 Byte
L := Länge des Schlüssels in Byte
s[]: Byte-Vektor der Länge 256
 Für i = 0 bis 255
   s[i] := i
 j := 0
 Für i = 0 bis 255
   j := (j + s[i] + k[i mod L]) mod 256
   vertausche s[i] mit s[j]
```

```
klar[]: gegebene Klartext-Zeichenfolge der Länge X
schl[]: Vektor zum Abspeichern des Schlüsseltextes
 i := 0
 j := 0
 Für n = 0 bis X-1
   i := (i + 1) mod 256
   j := (j + s[i]) mod 256
   vertausche s[i] mit s[j]
   zufallszahl := s[(s[i] + s[j]) mod 256]
   schl[n] := zufallszahl XOR klar[n]
```

- Aus Wikipedia (http://de.wikipedia.org/wiki/Rc4)

Block-Chiffre

- Nachrichten werden in Blöcken von k bits verschlüsselt
  - z.B. 64-bit Blöcke
- Injektive Abbildung um den Quelltext in den k-bit verschlüsselten Text umzuwandeln
- Beispiel k=3:

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

Block-Chiffre

- Wie viele mögliche Abbildungen gibt es für k-Bit Block-Chiffre?

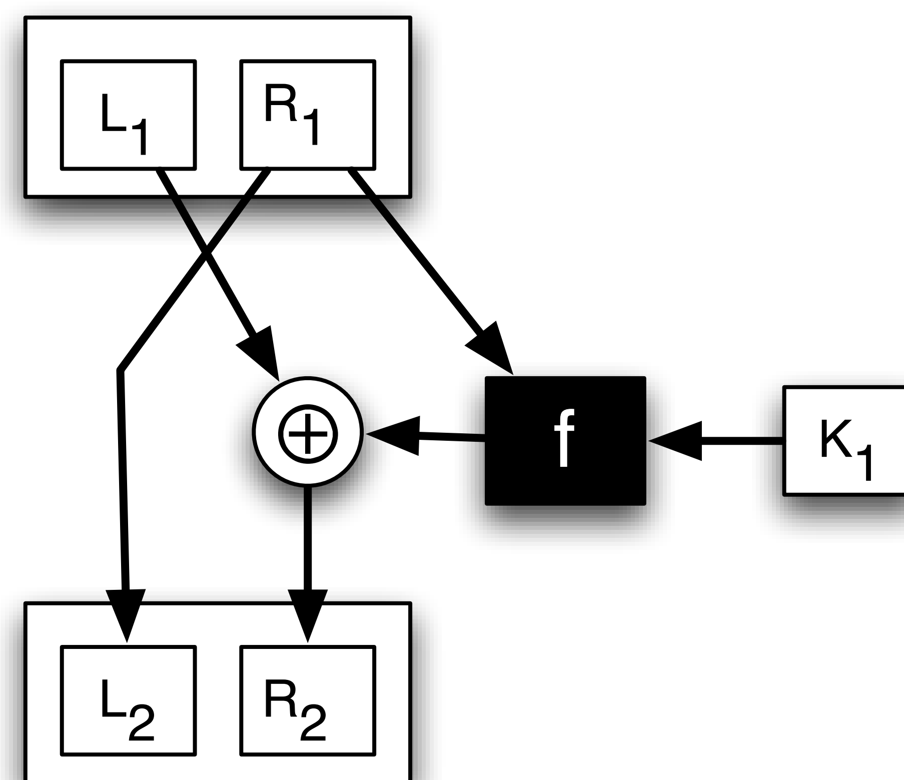
Block-Chiffre

- Wie viele mögliche Abbildungen gibt es für k-Bit Block-Chiffre?
  - Im allgemeinen: 2^k!
  - riesig für k=64
  - und absolut sicher, wenn man sie zufällig auswählt
- Problem:
  - Die meisten dieser Abbildungen benötigen große Tabellen um sie zu berechnen
- Lösung
  - Statt einer Tabelle, verwendet man eine Funktion, die diese Tabelle simuliert
  - Dadurch verliert man möglicherweise wieder die Sicherhei

Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

Feistel-Chiffre

- Aufteilung der Nachricht in zwei Hälften L_1, R_1
  - Schlüssel K_1, K_2, ...
  - Mehrere Runden: resultierender Code: L_n, R_n
- Verschlüsselung
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- Entschlüsselung
  - $R_{i-1} = L_i$
  - $L_{i-1} = R_i \oplus f(L_i, K_i)$
- f: beliebige, komplexe Funktion

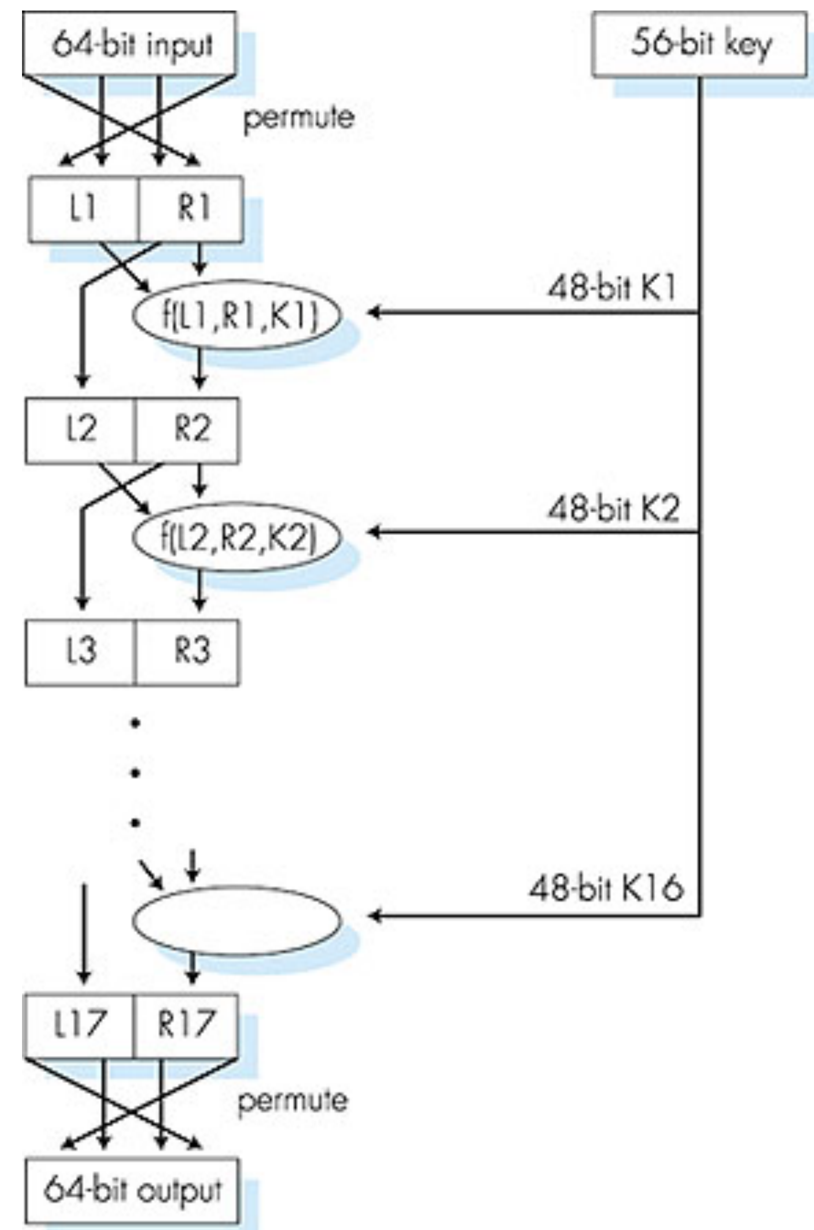


Weitere Symmetrische Codes

- Skipjack
  - 80-Bit symmetrischer Code
  - baut auf Feistel-Chiffre auf
  - wenig sicher
- RC5
  - Schlüssellänge 1-2048 Bits
  - Rivest Code 5 (1994)
  - Mehrere Runden der Feistel-Chiffre

Digital Encryption Standard

- Geschickt gewählte Kombination von
  - Xor-Operationen
  - Feistel-Chiffre
  - Permutationen
  - Table-Lookups
  - verwendet 56-Bit Schlüssel
- 1975 entwickelt von Wissenschaftlern von IBM
  - Mittlerweile nicht mehr sicher
    - leistungsfähigere Rechner
    - Erkenntnisse in Kryptologie
- Nachfolger: AES (2001)

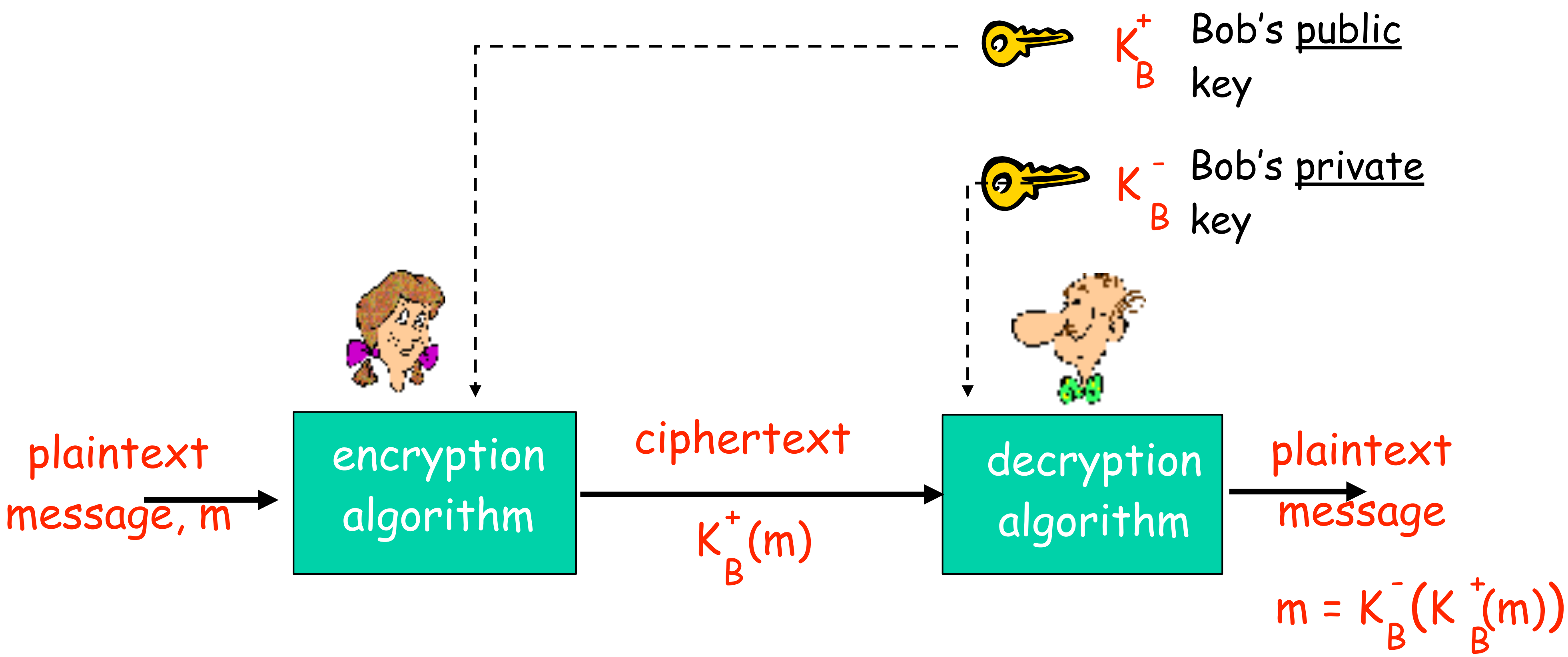


Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

Advanced Encryption Standard

- Geschickt gewählte Kombination von
  - Xor-Operationen
  - Permutationen
  - Table-Lookups
  - Multiplikation in $GF[2^8]$
  - symmetrische 128,192 oder 256-Bit Schlüssel
- Joan Daemen und Vincent Rijmen (Rijndael)
  - 2001 als AES unter vielen ausgewählt worden
  - bis heute als sicher erachtet

Public key cryptography



Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

Asymmetrische Verschlüsselungsmethoden

- z.B. RSA, Ronald Rivest, Adi Shamir, Lenard Adleman, 1977
  - Diffie-Hellman, PGP
- Geheimer Schlüssel privat: kennt nur der Empfänger der Nachricht
- Öffentlichen Schlüssel offen: Ist allen Teilnehmern bekannt
- Wird erzeugt durch Funktion
  - keygen(privat) = offen
- Verschlüsselungsfunktion f und Entschlüsselungsfunktion g
  - sind auch allen bekannt
- Verschlüsselung
  - f(offen,text) = code
  - kann jeder berechnen
- Entschlüsselung
  - g(privat,code) = text
  - nur vom Empfänger

RSA

- R. Rivest, A. Shamir, L. Adleman
  - On Digital Signatures and Public Key Cryptosystems, Communication of the ACM
- Verfahren beruht auf der Schwierigkeit der Primfaktorzerlegung
- 1. Beispiel:
  - 15 = ? * ?
  - 15 = 3 * 5
- 2. Beispiel:
  - 3865818645841127319129567277348359557444790410289933586483552047443 = 1234567890123456789012345678900209 * 31313131313131313131313131300227

Sicherheit RSA

- Bis heute ist kein effizientes Verfahren zur Primfaktorzerlegung bekannt
  - Aber das Produkt von Primzahlen kann effizient bestimmt werden
  - Primzahlen können effizient bestimmt werden
  - Primzahlen sehr häufig

RSA

- Erzeugung der Schlüssel
  - Wähle zufällig zwei Primzahlen p und q mit k bits ($k \geq 500$).
  - $n = p \cdot q$
  - e ist Zahl, die teilerfremd ist mit $(p - 1)\cdot(q - 1)$.
  - $d = 1/e \bmod (p - 1)(q - 1)$
    - es gilt $d \cdot e \equiv 1 \bmod (p - 1)(q - 1)$
- Public Key $P = (e, n)$
- Secret Key $S = (d, n)$

RSA

- Kodierung
  - Teile Nachricht in Blöcke der Größe 2^{2k} auf
  - Interpretiere Block M als Zahl $0 \leq M < 2^{2k}$
  - Chiffre: $P(M) = C = M^e \bmod n$
- Dekodierung
  - $S(C) = C^d \bmod n = M$
- Korrektheit gilt nach dem kleinen Satz von Fermat
  - Für Primzahl p und von p teilerfremde Zahl a gilt:

$$a^p \equiv a \pmod{p}$$

RSA Beispiel

- Bob wählt p=11, q=13
  - n =143, (p-1).(q-1)=120
  - e = 23
  - d = 47
    - e d = 1 mod 24
- Verschlüsselung von 8-Bit-Nachricht:

Bit pattern	m	m^e	$c=m^e \bmod n$
00000111	7	2.7368747e+19	2

- Entschlüsselung:

c	$c^d = 2^{47}$	$m=c^d \bmod n$
2	1.4073749e+14	7

Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

RSA Beispiel
Binäre Exponentiation

- Berechnung von 7^{23} mod 143,
- $23 = 10111_2$
  - $7^{23} = 7 \cdot (7^{11})^2$
  - $7^{11} = 7 \cdot (7^5)^2$
  - $7^5 = 7 \cdot (7^2)^2$
- Einsetzen:

$$7^{23} \bmod 143 = (((7^2)^2.7)^2.7)^2.7 \bmod 143 = 2$$

Zwischenergebnisse klein halten:

$$(a \bmod n) \bmod n = a \bmod n$$