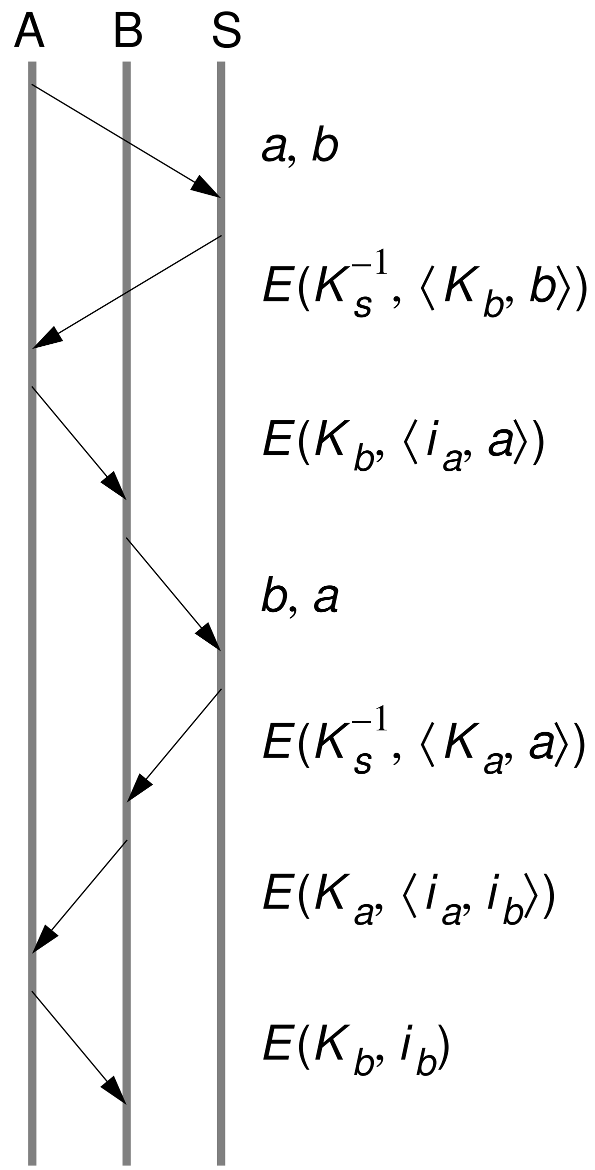
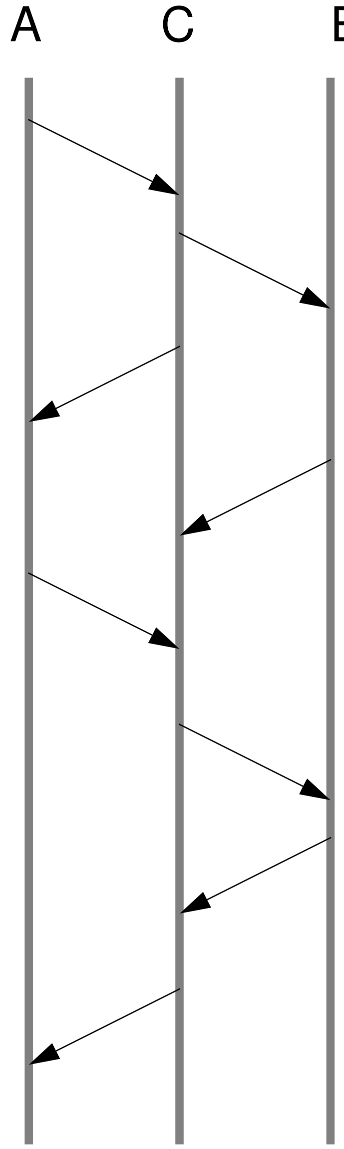
- Austausch über Authentifizierungsserver



K_x	öffentlicher Schlüssel von X
K_x^{-1}	geheimer Schlüssel von X
i_x	Einmal-Wort, erzeugt von X

- ’replay’-Probleme ähnlich wie beim symmetrischen Verfahren; Lösung durch Hinzufügen von Zeitstempel und Lebensdauer

- Problem: aktiver Mithörer C



$$K_a$$

$$K_c$$

$$K_c$$

$$K_b$$

$$\overline{M}_1 = E(K_c, E(K_a^{-1}, M_1))$$

$$M_1 = D(K_a, D(K_c^{-1}, \overline{M}_1))$$

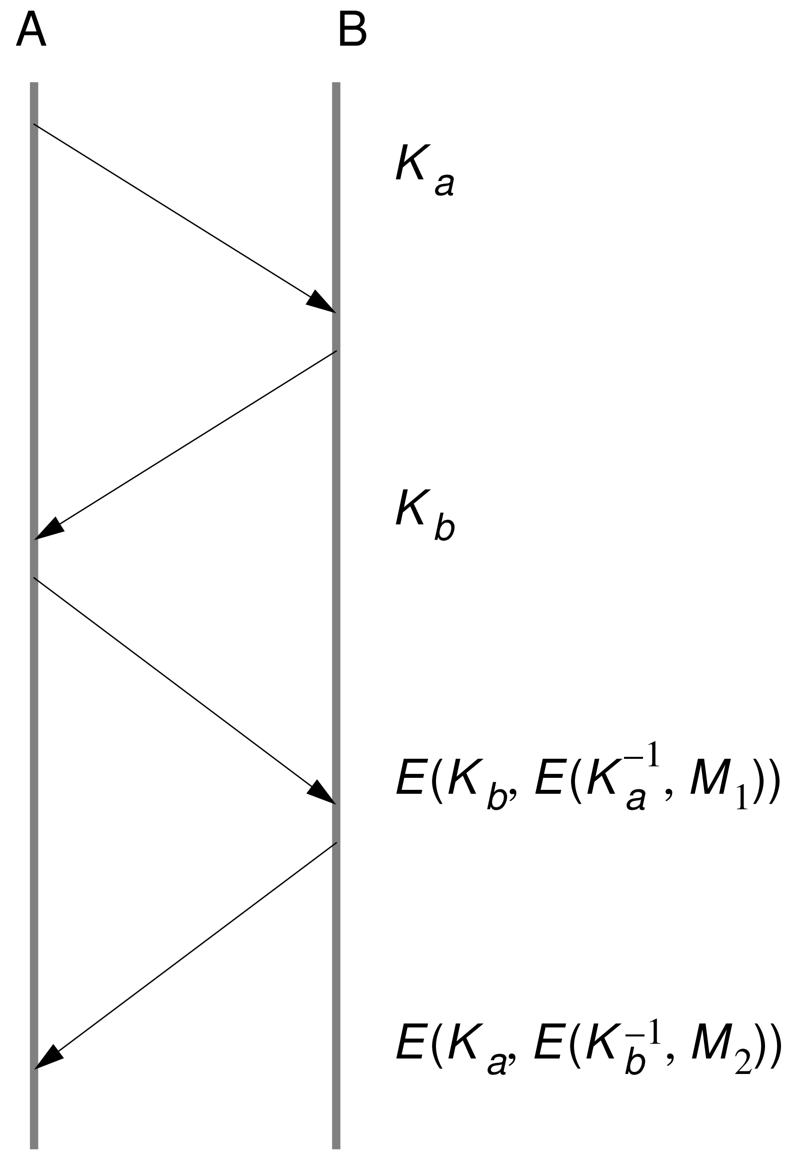
$$E(K_b, E(K_c^{-1}, M_1))$$

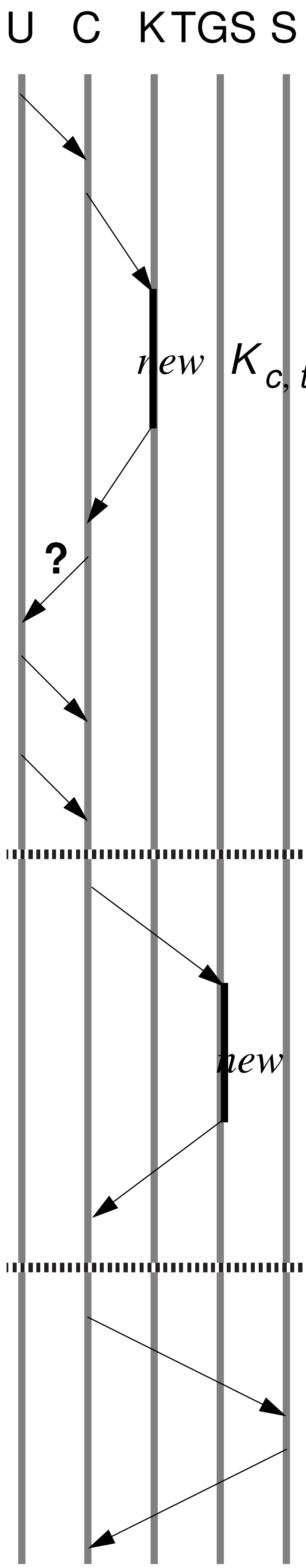
$$\overline{M}_2 = E(K_c, E(K_b^{-1}, M_2))$$

$$M_2 = D(K_a, D(K_c^{-1}, \overline{M}_2))$$

$$E(K_a, E(K_c^{-1}, M_2))$$

- Sicherer Austausch öffentlicher Schlüssel
  - Idee





U	user
C	client
K	Kerberos
TGS	ticket granting server
S	server

u *(userid)*

c, u, tgs, T_1

new $K_{c,\,tgs}$; $\quad T_{c,\,tgs} = E(K_{tgs}, \langle c, u, tgs, K_{c,\,tgs}, T, L \rangle)$

$E(K_c, \langle T_{c,\,tgs}, K_{c,\,tgs}, tgs, T_1 \rangle)$

u

s

$s, T_{c,\,tgs}, E(K_{c,\,tgs}, \langle c, T_2 \rangle)$

new $K_{c,\,s}$; $\quad T_{c,\,s} = E(K_s, \langle c, s, K_{c,\,s}, T, L \rangle)$

$E(K_{c,\,tgs}, \langle T_{c,\,s}, K_{c,\,s}, s, T_2 \rangle)$

$T_{c,\,s}, E(K_{c,\,s}, \langle c, T_3 \rangle)$

$E(K_{c,\,s}, T_3 + 1)$

- Konfiguration:

U	Benutzer (user)
C	Prozeß, der den Benutzer im System vertritt (client)
K	Kerberos-Server
TGS	Ticket Granting Server
S	Dienstleister (server)

- Vorausetzungen:

  Jedes Paar (U, K) verfügt über einen eigenen geheimen Schlüssel K_u.

  K und TGS verfügen über einen geheimen Schlüssel K_{tgs}.

  $K_{x,y}$ bezeichnet einen zwischen X und Y geheimen Schlüssel.

  Die Funktion f ist eine Einweg-Funktion, d. h. sie ist praktisch nicht invertierbar und es ist
  $K_c = f(user\ password) = f(u)$

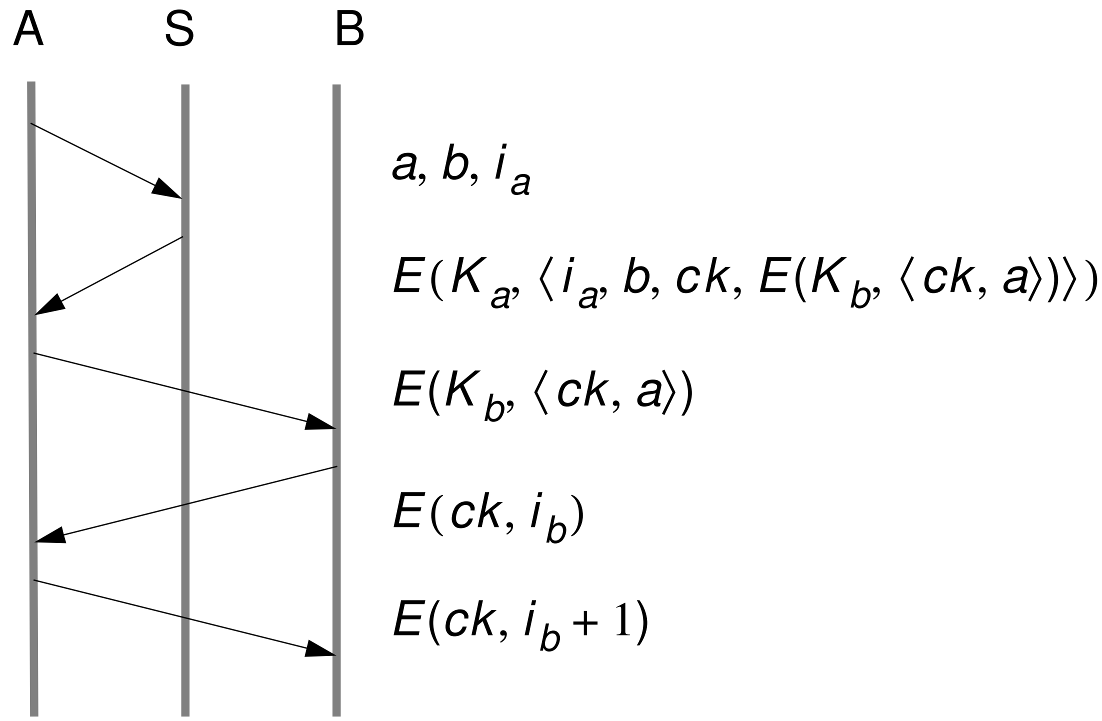
  n ist ein Einmal-Wort.

  Die Paare $\langle tgs, T_1\rangle$ und $\langle s, T_2\rangle$ dienen als Einmal-Worte.

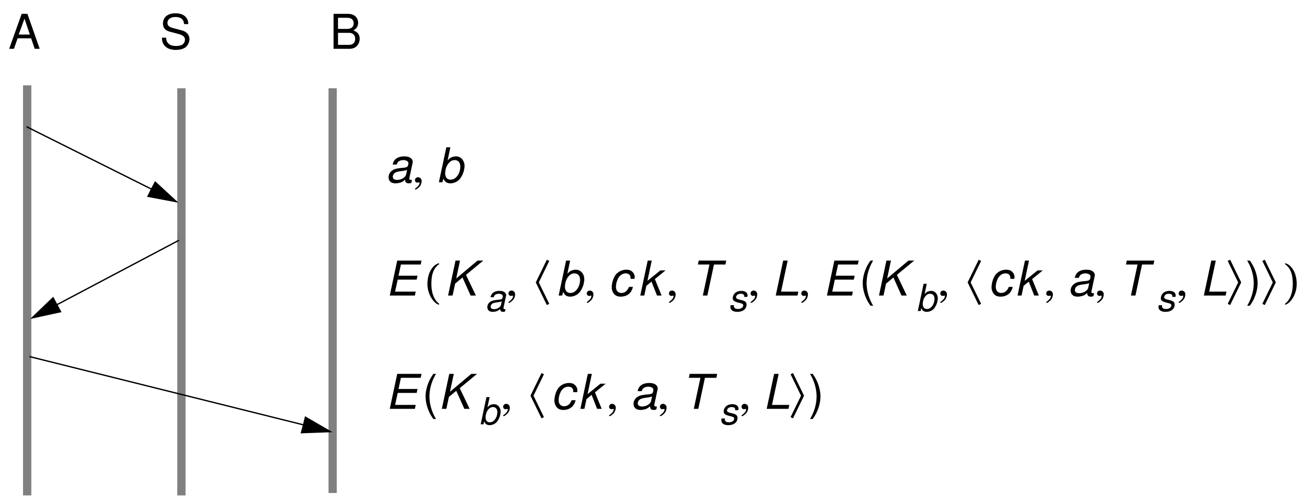
  Protokollablauf

- ❏ Anwendungsbeispiel Kerberos (V4, Weiterentwicklung des Needham-Schröder-Protokolls)
  - Entwickelt am MIT 1986
  - Ziel
    - Authentifizierung und Erzeugung eines gemeinsamen Schlüssels durch vertrauenswürdigen Server Kerberos
    - Zur Reduktion der Gelegenheiten, bei denen Identifikation oder Schlüssel eines Benutzers zum Kerberos übertragen werden müssen, nimmt Kerberos die Authentifizierung vor und stellt nur ein spezielles Ticket aus. Dieses Ticket ermöglicht durch Vorlage beim ’ticket granting server’ die Einrichtung eines gemeinsamen geheimen Schlüssels mit einem Partner.

- Korrektur mit Hilfe einer Einmal-Information i (nonce)



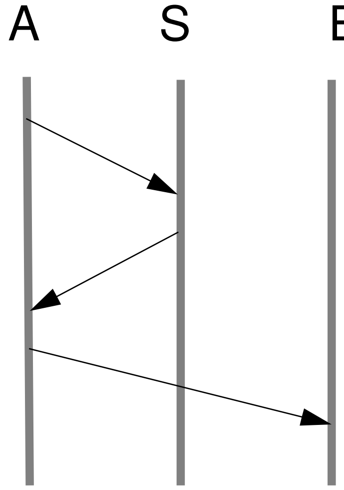
- Zukünftige Täuschung, wenn Schlüssel *ck* einmal gebrochen wurde und die Nachricht $E(K_b, \langle ck, a\rangle)$ aufgezeichnet wurde.
- Verbesserung durch Zeitstempel und Lebensdauer (Denning und Sacco)



Authentifizierung

❑ Sicherer Austausch eines symmetrischen Schlüssels

- Einführung eines Authentisierungsservers S (Needham-Schröder)



a, b, i_a

$E(K_a, \langle i_a, b, ck, E(K_b, \langle ck, a\rangle)\rangle)$

$E(K_b, \langle ck, a\rangle)$

K_x geheimer Schlüssel zwischen X und S

ck vom Server ad hoc gewählter 'communication key', der zur Sicherung der Kommunikation zwischen A und B dienen soll und nur A und B bekannt werden darf.

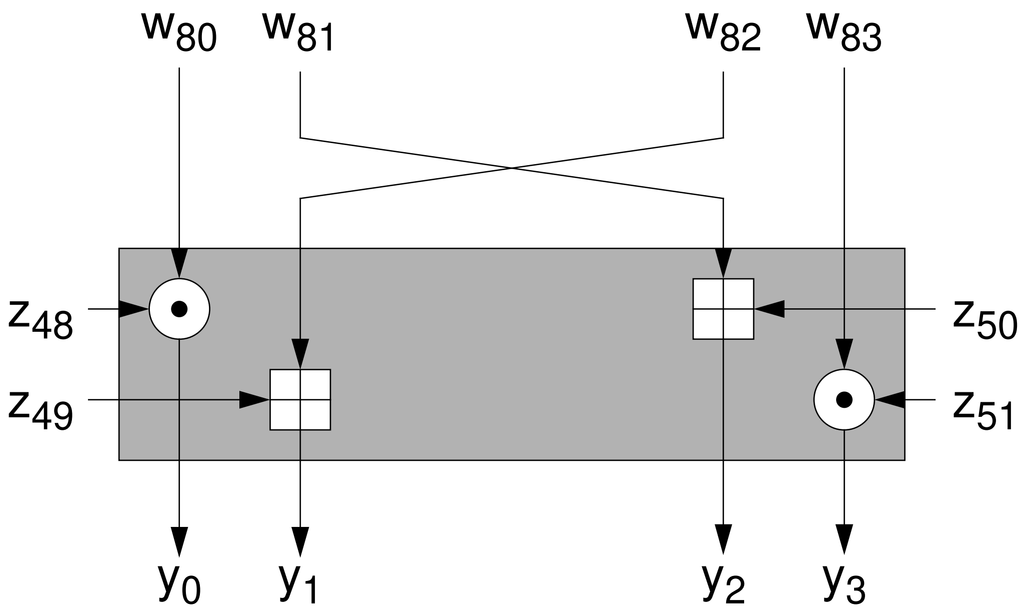
A weiß, daß (außer dem Server) nur B seine Nachrichten entschlüsseln kann und alle mit *ck* verschlüsselten Nachrichten, die er erhält nur von B kommen können.

B weiß nicht, ob die letzte Nachricht tatsächlich von A stammt oder ob es sich um ein 'replay' einer frühren Kommunikation handelt. Ein Fremder könnte so die Kommunikation zwischen A und B unterbinden.

❑ Verletzlichkeit asymmetrischer Verfahren

- Bekanntwerden des Paß-Satzes oder des geheimen Schlüssels
  - ermöglicht Entschlüsselung
  - ermöglicht Fälschung von Unterschriften
- Verfälschung öffentlicher Schlüssel
  - öffentlicher Schlüssel nur vertrauenswürdig, wenn vom Besitzer selbst oder einem Vertrauenswürdigen ausgehändigt
  - Führung der Schlüssel nicht auf einem System, zu dem Fremde Zugang haben!
- Backup-Kopien von beiden Schlüsseln
- "Löschen" von Dateien und Speicherbereichen
- Viren, Trojanische Pferde
  - Infektion von PGP
  - Bezug eines infizierten oder verfälschten PGP
  - PGP unterstellt, daß Rechner gegen Viren gesichert ist
- "Abhören" der elektromagnetischen Abstrahlung
  - Sichtgerät
  - Übertragungswege (Kabel zu peripheren Geräten)
- "Beobachtung" durch andere im Mehrbenutzerbetrieb
  - Beobachtung während der Entschlüsselung
  - Entschlüsselung nur an einem separaten, privaten System

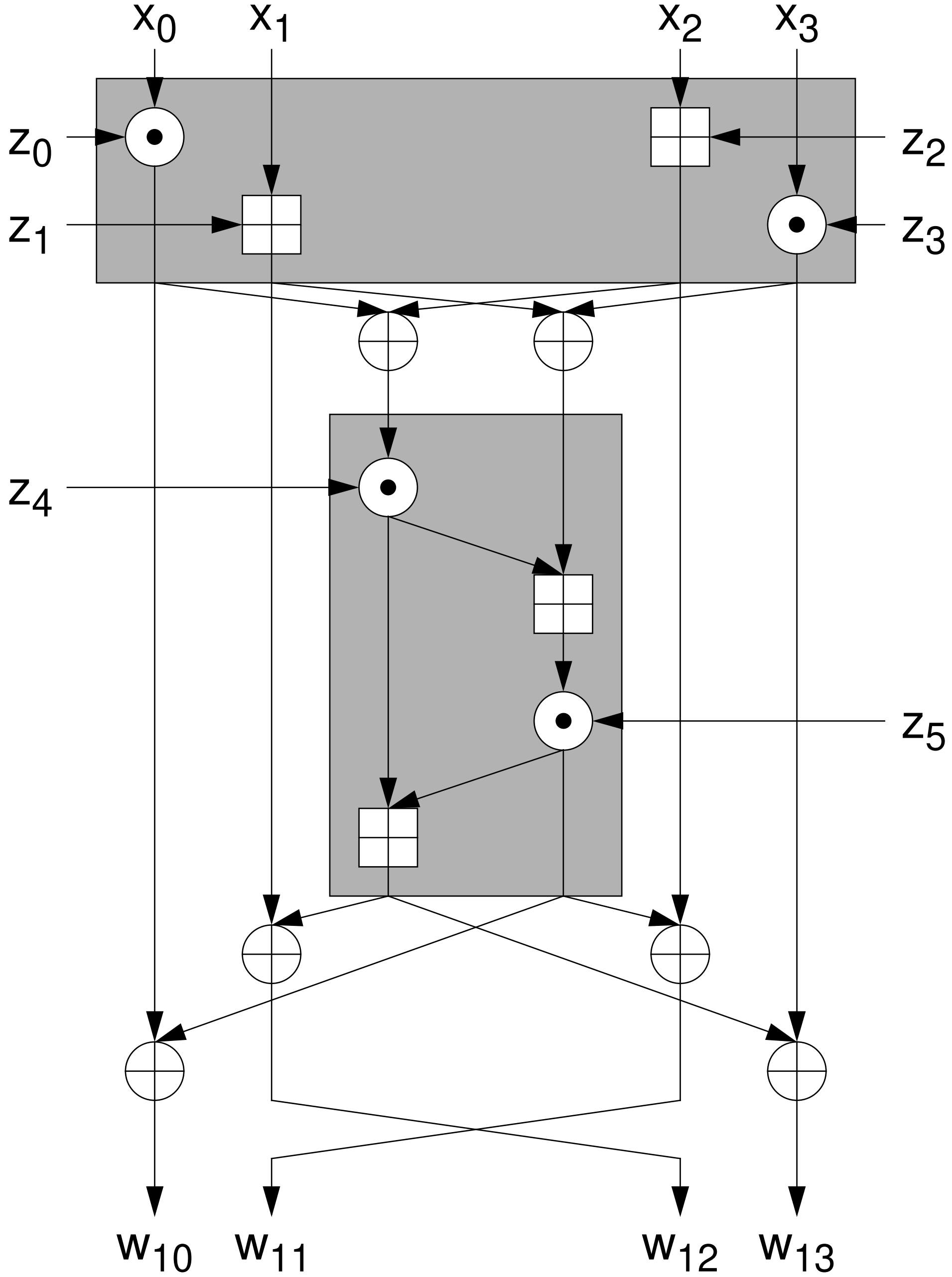
Ausgabeumwandlung



Elementare Operationen

- $\oplus$ bitweises XOR
- $\bullet$ Addition modulo 2^{16} (16 Bit unsigned)
- $+$ Multiplikation modulo $2^{16}+1$, wobei für die Multiplikation Faktoren mit dem Wert 0 durch 2^{16} ersetzt werden

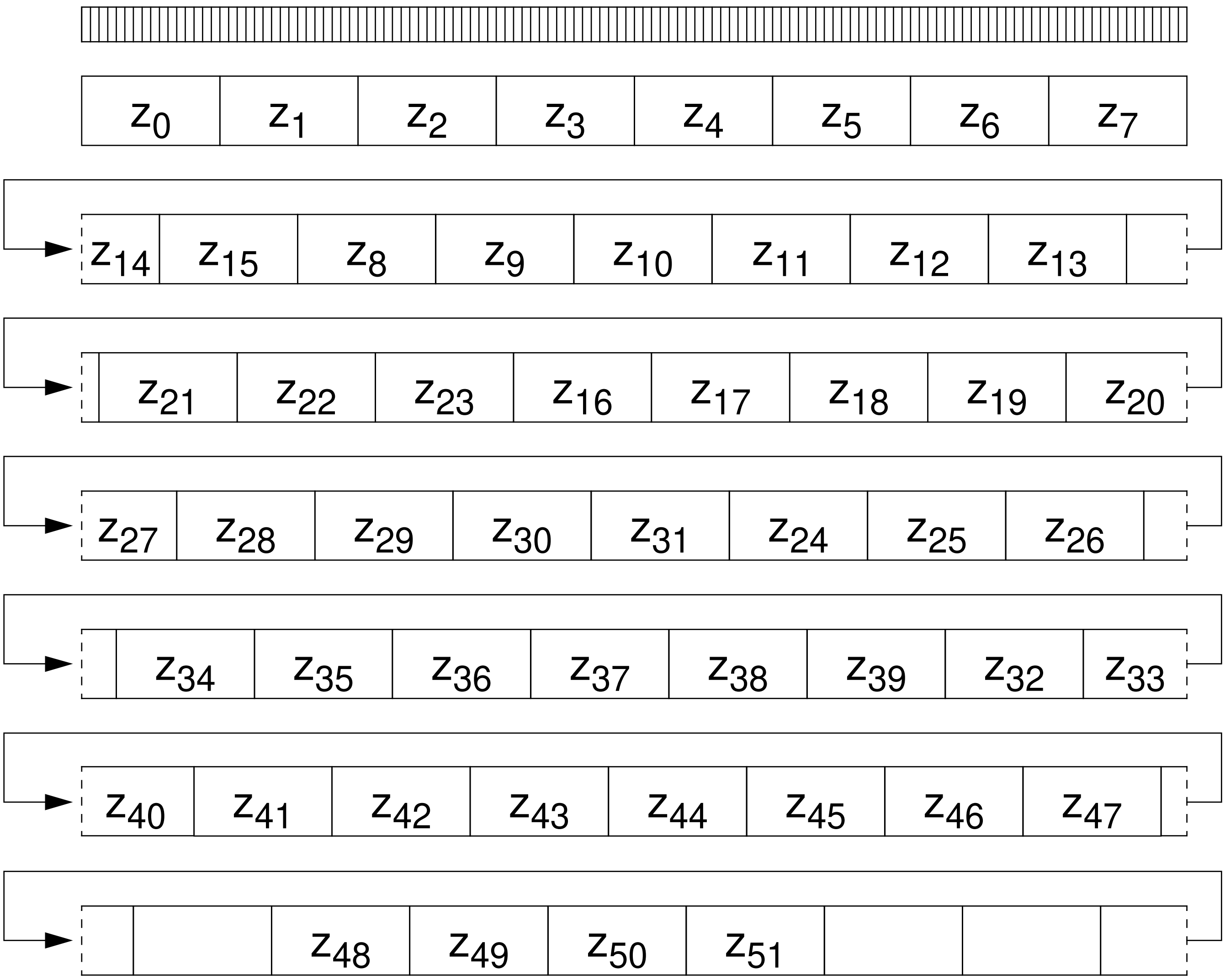
Iteration



Gewinnung der Unterschlüssel

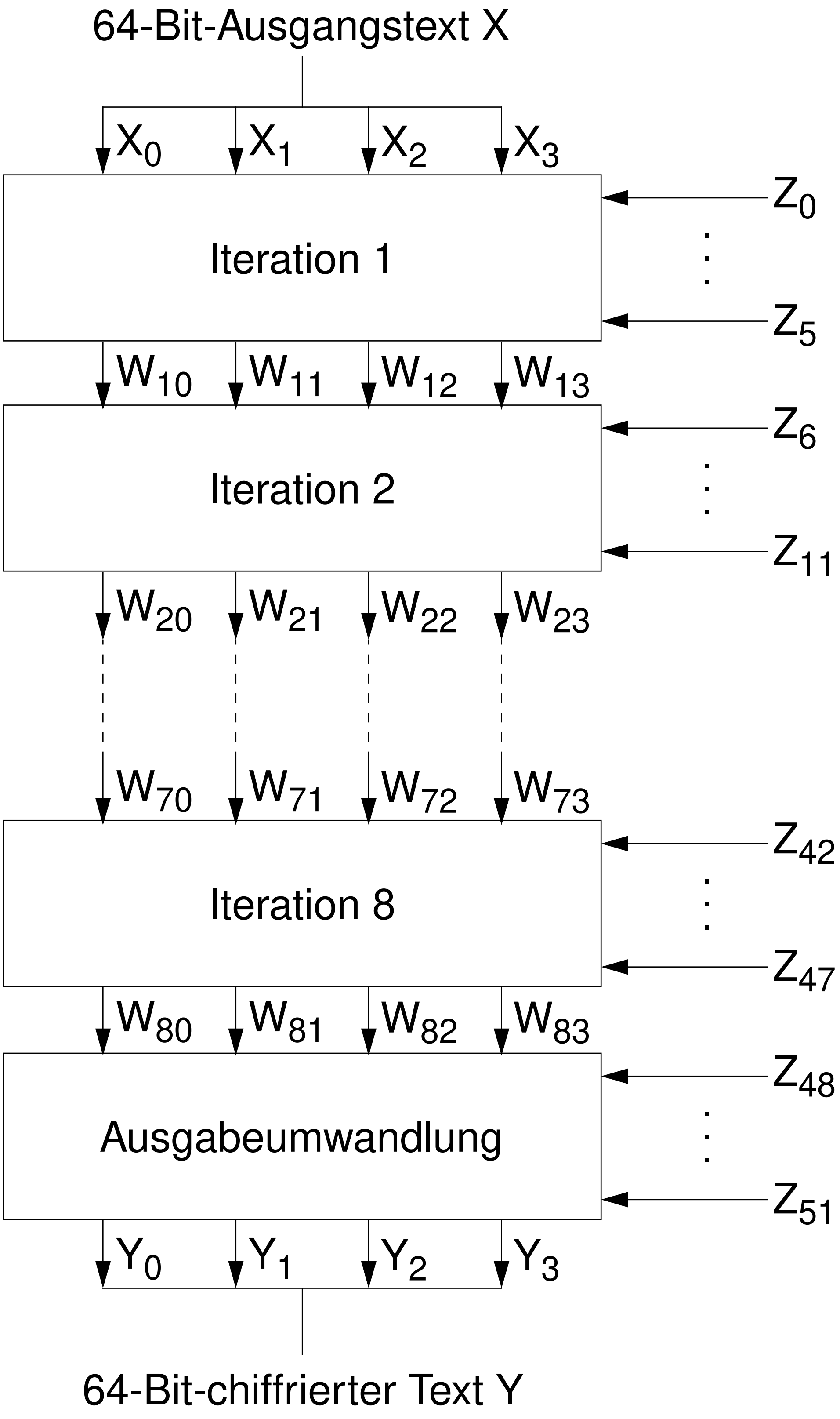
- Zur Gewinnung der ersten 8 Unterschlüssel Z_0 bis Z_7 wird der Schluessel in 8 Worte zu 16 Bit unterteilt.
- Anschließend wird er um 25 Bit zyklisch nach links verschoben und in weitere 8 Worte zu 16 Bit unterteilt.
- Dieser Vorgang wird solange wiederholt, bis alle benötigten 52 Unterschlüssel ermittelt sind.

Schlüssel: 128 Bit



❑ Symmetrische Verfahren

- DES (Data Encryption Standard, standardisiert 1977)
- IDEA (International Data Encryption Algorithm)



Zahlenbeispiel:

$p = 47$, $q = 59$, $n = pq = 2773$, $(p-1)(q-1) = 2668$ und $d = 157$.

Es ist festzustellen, ob d zu $(p-1)(q-1)$ teilerfremd ist und gegebenenfalls e zu berechnen. Beides wird mit dem Euklidischen Algorithmus erledigt, indem man $x_0 = (p-1)(q-1)$ setzt, $x_1 = d$ und $x_{i+1} = x_{i-1} mod x_i$.

Weiter bestimmt man a_i, b_i so, daß $x_i = a_i x_0 + b_i x_1$ ist.

Wenn k der kleinste Index ist mit $x_k = 1$, dann ist $e = b_k$.

i		x_i	a_i	b_i
0	2668	2668	1	0
1	157	157	0	1
2	2668 = 16x157+156	156	1	-16
3	157 = 1x156+1	1	-1	17 = e

Man wähle zwei Primzahlen p und q.

Man bilde das Produkt $n = pq$.

Man wähle zufällig eine große Zahl d die teilerfremd zu $(p-1)(q-1)$ ist.

Man berechne e aus der Bedingung
$ed = 1 mod((p-1)(q-1))$

Man kann zeigen, daß die so gewonnenen Zahlen die Eigenschaft $\forall M((M^e)^d = M)$ besitzen.

❑ Beispiel eines asymmetrischen Kryptosystems
Rivest, Shamir und Adleman

- Man zerlege den Text in Blöcke, die als Zahlen zwischen 0 und n aufgefaßt werden können.
- Der veröffentlichte Schlüssel besteht aus einem Zahlenpaar (e, n).
- Die Verschlüsselung eines Blockes M erfolgt durch Berechnung von

$$C = E((e, n), M) = M^e \mathit{mod}\, n.$$

- Die Entschlüsselung eines Blockes C erfolgt durch Berechnung von

$$M = D((d, n), C) = C^d \mathit{mod}\, n.$$

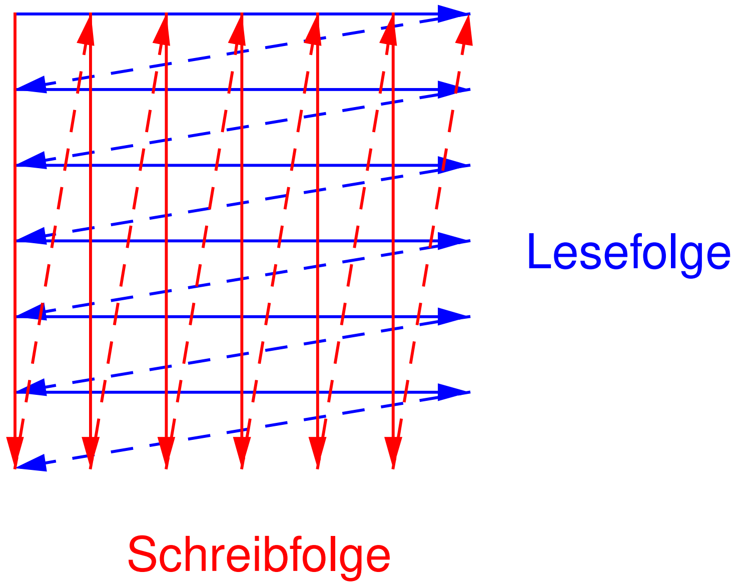
- Die Entschlüsselung funktioniert jedenfalls dann, wenn gilt:

$$\forall M((M^e)^d = M \mathit{mod}\ n)$$

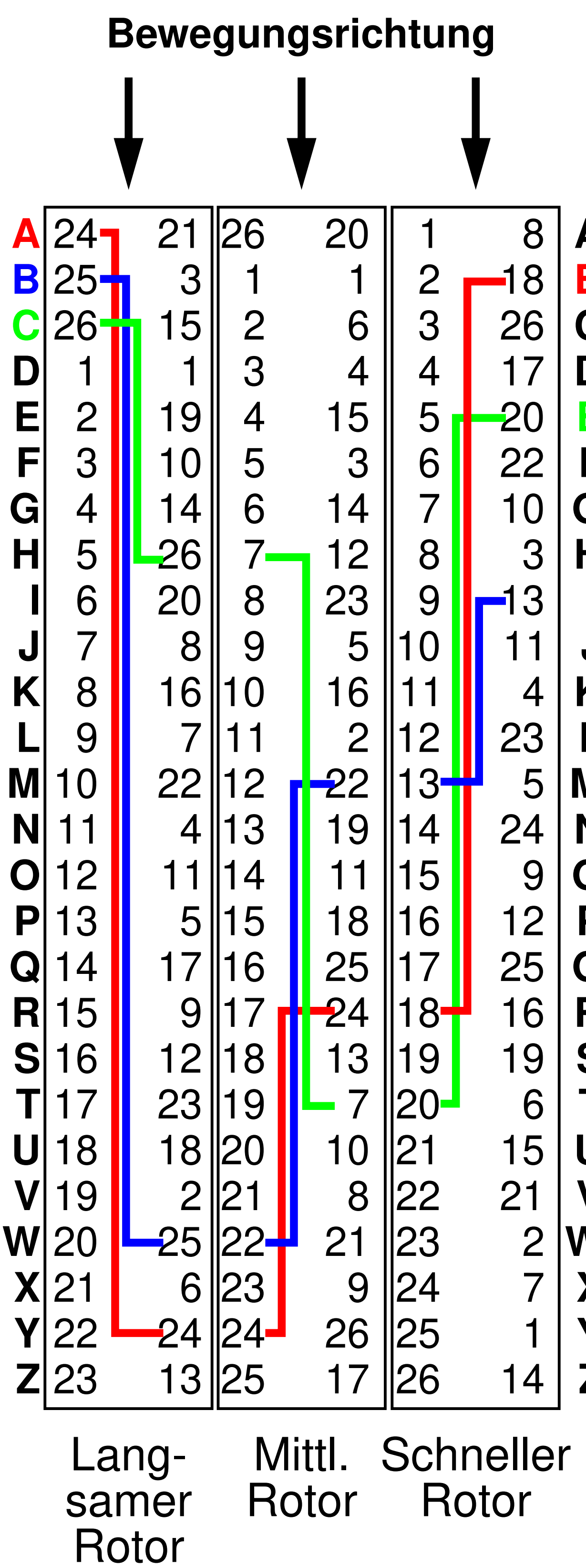
Wie findet man Zahlen d, e und n, die diese Eigenschaft besitzen, wobei es zudem sehr aufwendig ist, aus der Kenntnis von e und n die Zahl d zu ermitteln?

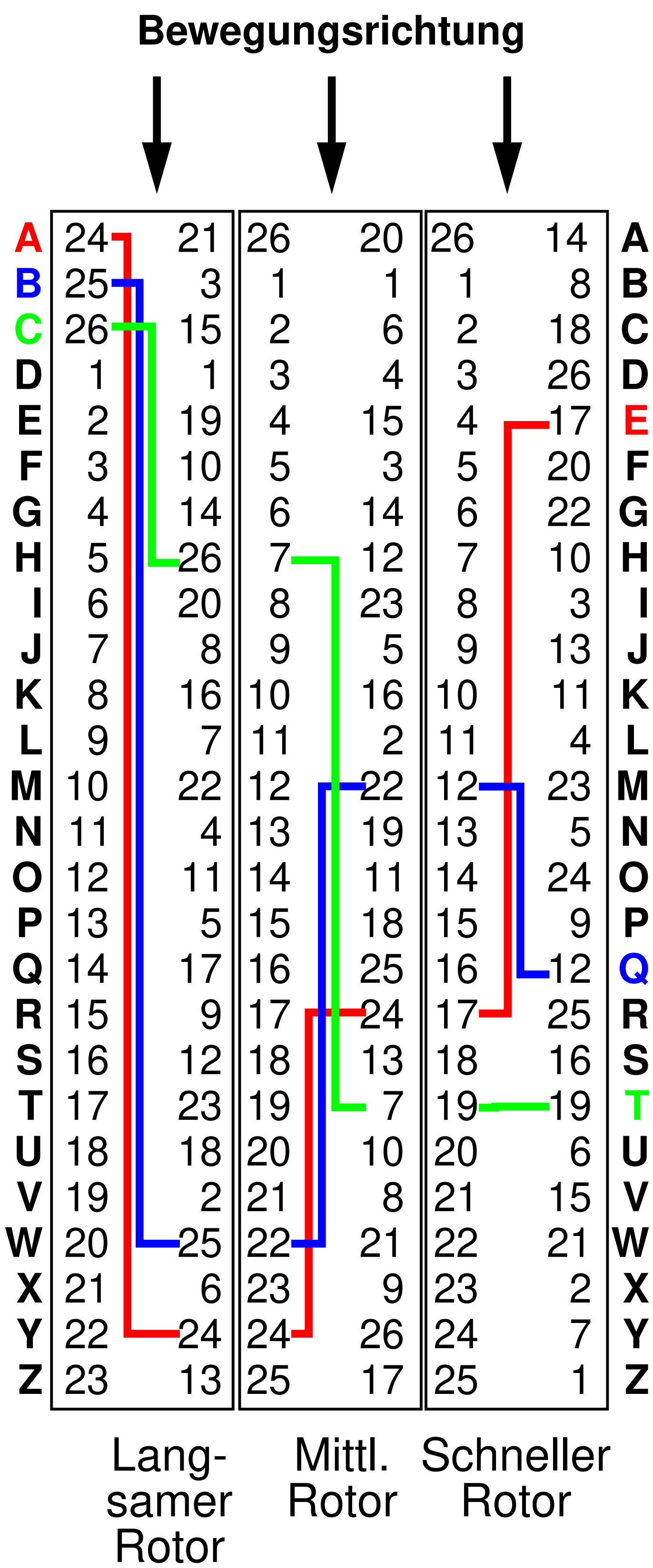
- Transposition

  Lesefolge und Verschlüsselungsfolge unterschiedlich



Beispiel: Rotormaschine





- Polyalphabetische Verschlüsselung

  Beispiel: Vigenére (Caesar-Verschlüsselungen mit einer zyklisch wiederholten Folge von Verschiebungswert)

  In folgender Tabelle dienen die Zeichen des geheimen Schlüsselwortes zur Zeilenwahlt

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

- CAESAR

  $E: M \rightarrow (M + k_1)\ mod\ 26$

  Nachteil: Leicht durch vollständiges Probieren zu brechen

- Substitution

  26! Permutationen

  Nachteil: Brechbar durch Häufigkeitsanalysen

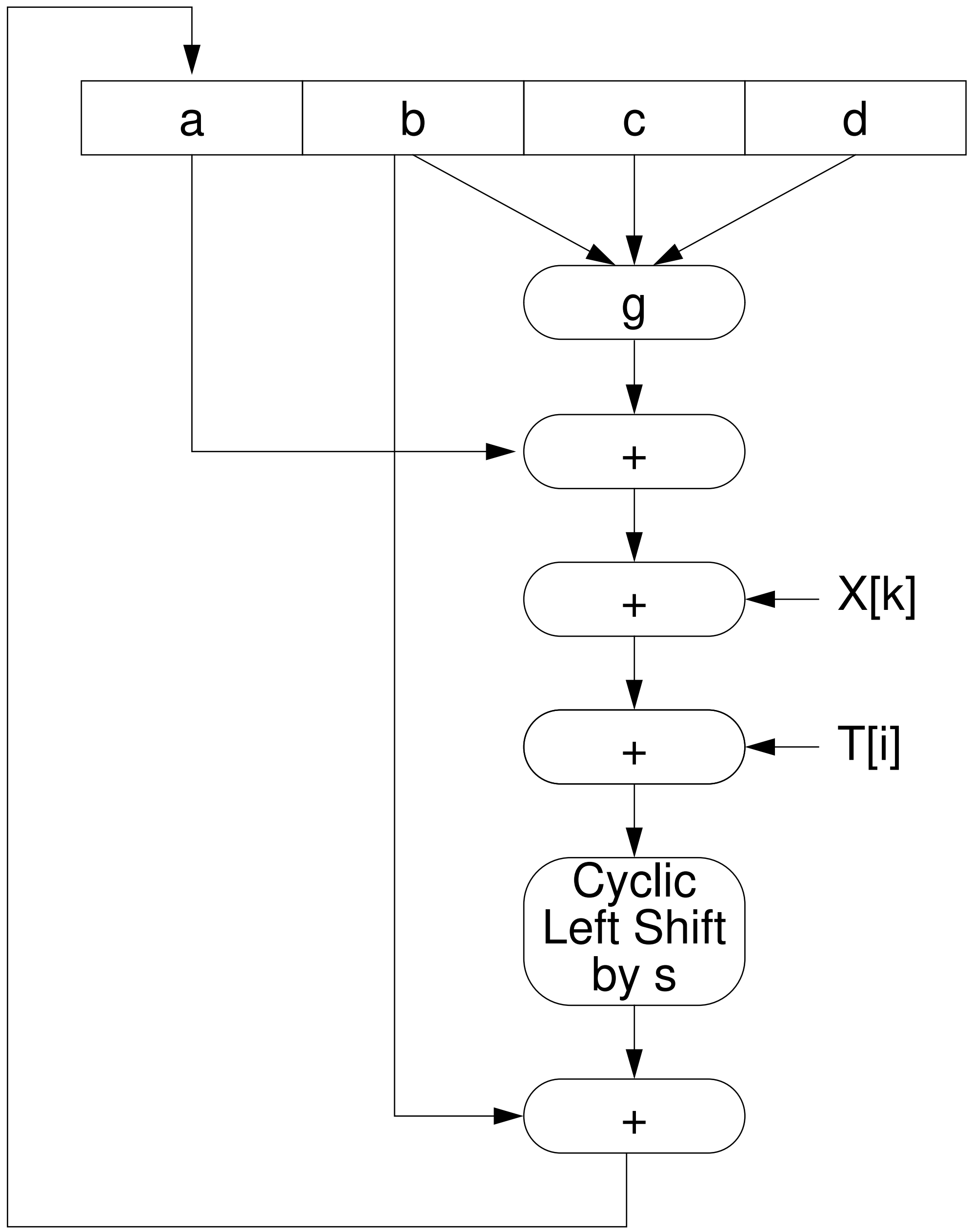
```
T[ 1] = D76AA478
T[ 2] = E8C7B756
T[ 3] = 242070DB
T[ 4] = C1BDCEEE
T[ 5] = F57C0FAF
T[ 6] = 4787C62A
T[ 7] = A8304613
T[ 8] = FD469501
T[ 9] = 698098D8
T[10] = 8B44F7AF
T[11] = FFFF5BB1
T[12] = 895CD7BE
T[13] = 6B901122
T[14] = FD987193
T[15] = A679438E
T[16] = 49B40821
T[17] = F61E2562
T[18] = C040B340
T[19] = 265E5A51
T[20] = E9B6C7AA
T[21] = D62F105D
T[22] = 02441453
T[23] = D8A1E681
T[24] = E7D3FBC8
T[25] = 21E1CDE6
T[26] = C33707D6
T[27] = F4D50D87
T[28] = 455A14ED
T[29] = A9E3E905
T[30] = FCEFA3F8
T[31] = 676F02D9
T[32] = 8D2A4C8A

T[33] = FFFA3942
T[34] = 8771F681
T[35] = 6D9D6122
T[36] = FDE5380C
T[37] = A4BEEA44
T[38] = 4BDECFA9
T[39] = F6BB4B60
T[40] = BEBFBC70
T[41] = 289B7EC6
T[42] = EAA127FA
T[43] = D4EF3085
T[44] = 04881D05
T[45] = D9D4D039
T[46] = E6DB99E5
T[47] = 1FA27CF8
T[48] = C4AC5665
T[49] = F4292244
T[50] = 432AFF97
T[51] = AB9423A7
T[52] = FC93A039
T[53] = 655B59C3
T[54] = 8F0CCC92
T[55] = FFEFF47D
T[56] = 85845DD1
T[57] = 6FA87E4F
T[58] = FE2CE6E0
T[59] = A3014314
T[60] = 4E0811A1
T[61] = F7537E82
T[62] = BD3AF235
T[63] = 2AD7D2BB
T[64] = EB86D391
```

❑ Konventionelle Kryptosysteme

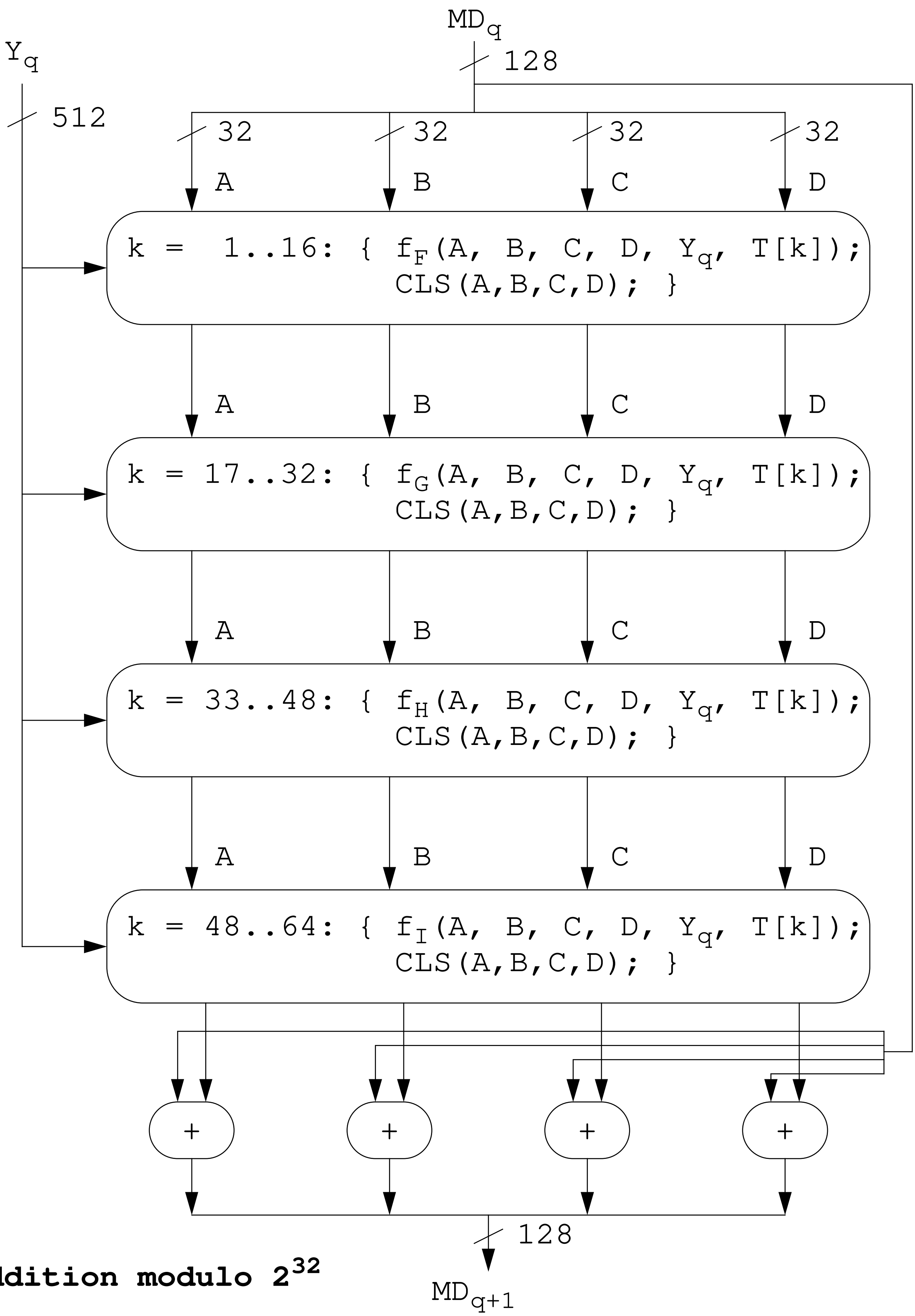
$$T[i] = \lfloor \sin(i) \cdot 2^{32} \rfloor$$

❑ In MD5 verwendete Basis-Funktionen

- $F(b, c, d) = (b \wedge c) \vee (\neg b \wedge d)$
- $G(b, c, d) = (b \wedge d) \vee (c \wedge \neg d)$
- $H(b, c, d) = b \oplus c \oplus d$
- $I(b, c, d) = c \oplus (c \wedge \neg d)$
- $f_g(a, b, c, d, k, s, i)$



X[k] k-tes 32-Bit-Wort im aktuellen 512-Bit-Eingabeblock



Addition modulo 2^{32}

- ❏ Aus welchen Gründen kann *B* wissen, daß die erhaltene Nachricht *M* "in Ordnung" ist?
  - Weil *B* die Nachricht *M* schon vorher kennt.
  - Weil *B* Teile der Nachricht *M* schon vorher kennt und weiß, daß der Rest davon abhängig ist.
  - Weil *B* weiß, daß *M* strukturelle Redundanz enthält, die nicht zufällig entsteht (Beispiel MD5).

- ❏ Signatur und Vertraulichkeit
  - Unterschriften werden mit dem eigenen, geheimen Schlüssel verschlüsselt.
  - Nachrichten werden mit dem öffentlichen Schlüssel des Adressaten verschlüsselt.

In dieser Reihenfolge!

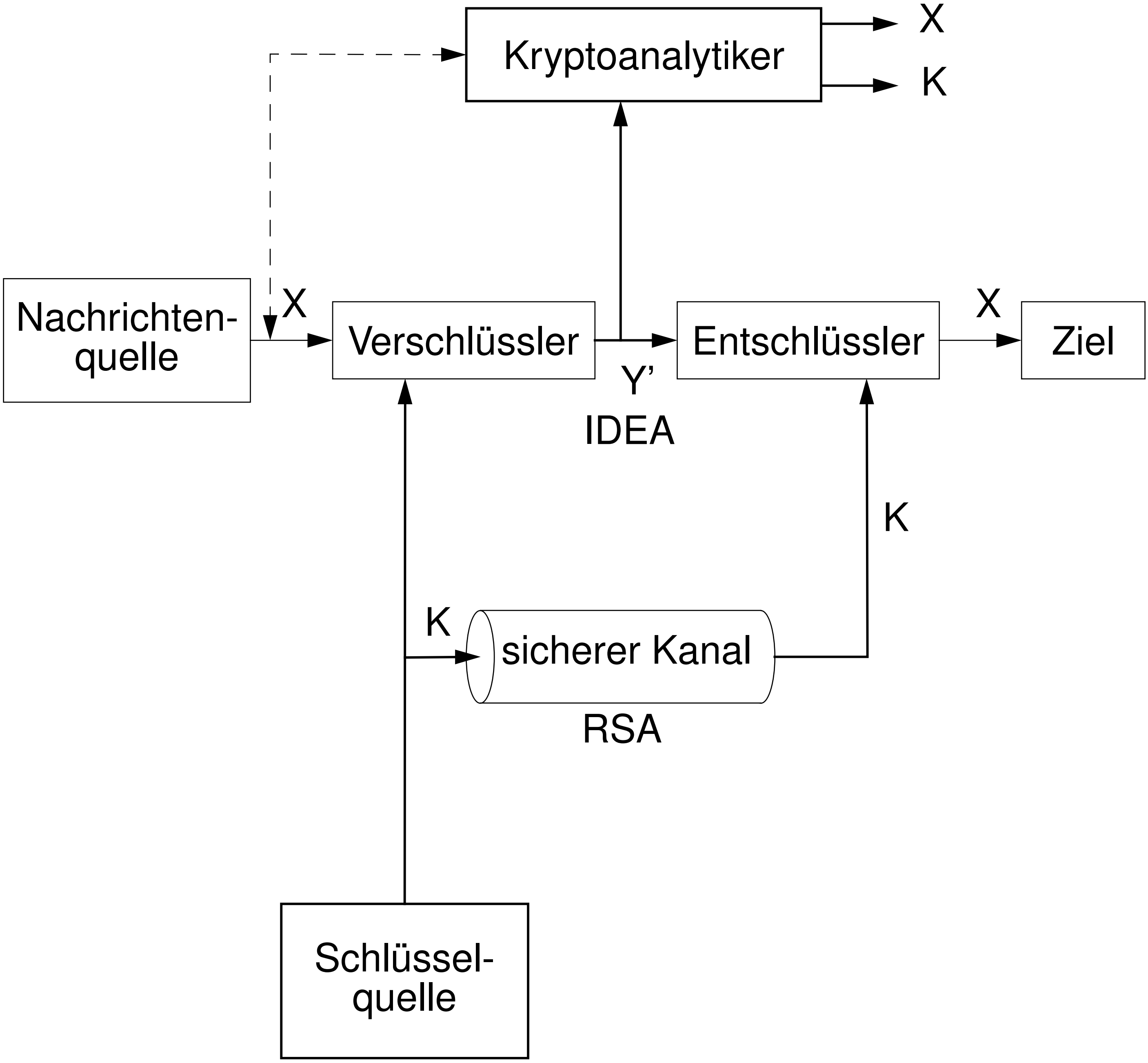
  - Man signiere nichts, was man nicht entschlüsseln kann.

- Forderung
  - Wenn K_2 unbekannt ist, soll es sehr aufwendig sein, T aus der Kenntnis von $E(K_1, T)$ zu ermitteln.
  - Es soll sehr aufwendig sein aus T und $E(K_1, T)$ den Schlüssel K_1 zu ermitteln.
  - In asymmetrischen Kryptosystemen soll es sehr aufwendig sein, aus K_1 den Schlüssel K_2 zu ermitteln und umgekehrt.

- Asymmetrische Systeme mit öffentlichem Schlüssel (public key systems)
  - A hält K_1 geheim und veröffentlicht K_2.
  - Warum kann B darauf vertrauen, daß die Information $D(K_2, X)$ von A stammt?

    Weil nur A den Schlüssel K_1 kennt.
  - Warum kann B darauf vertrauen, daß K_1 und K_2 verwandte Schlüssel sind?

    Weil jemand, dem B vertraut, es ihm zusichert.
  - Letztlich bedeutet dies, daß man einigen sehr weit verbreiteten Tatsachen vertraut.


Kryptoanalytiker
X
K
Nachrichten-
quelle
X
Verschlüssler
Y'
IDEA
Entschlüssler
X
Ziel
K
K
sicherer Kanal
RSA
Schlüssel-
quelle

8.5 Kryptographische Maßnahmen

❑ Literatur

Stallings. W.: Protect Your Privacy: The PGP user's Guide. Englewood Cliffs, NJ: Prentice Hall, 1995.

❑ Sichere Kanäle

- Was ankommt ist das Gleiche wie das, was gesendet wurde.
- Nur der beabsichtigte Empfänger kann die Information lesen.
- Der Empfänger kann nicht über den Urheber getäuscht werden.

❑ Kryptographie

- Es gibt eine Verschlüsselungsfunktion E (encrypt) und eine Entschlüsselungsfunktion D (decrypt). Beides sind Funktionen von zwei Variablen, nämlich einem Schlüssel und einem Text.
- Verschlüsselung wird dargestellt durch $E(K, T)$ und Entschlüsselung durch $D(K, T)$. Ein Paar K_1 und K_2 von Schlüsseln heißt verwandt, wenn gilt:

$$D(K_2, E(K_1, T)) = T$$

- Ein Kryptosystem heißt symmetrisch, wenn $K_1 = K_2$ ist, sonst heißt es asymmetrisch. In den gängigen asymmetrischen Algorithmen gilt

$$D(K_2, E(K_1, T)) = T$$

und

$$E(K_2, D(K_1, T)) = T.$$