Kryptographie - Wie funktioniert Verschlüsselung?

v12

- Historische Verfahren
- Stromverschlüsselung
- Blockverschlüsselung
- Schlüsselaustausch
- Public-Key Verschlüsselung
- Signatur

Terminologie

- **Verschlüsselung** Vorgang, bei dem ein klar lesbarer Text in eine nicht einfach interpretierbare Zeichenfolge umgewandelt umgewandelt wird. (Chiffrierung)

- **Entschlüsselung** beschreibt im weiteren Sinne Deutung unbekannter Zeichen, Symbole, bzw. deren Umwandlung in bekannte Zeichen. (Dechiffrierung)

- **Entzifferung** eine kryptanalytische Methode, die aus einem Geheimtext ohne vorherige Kenntnis des Schlüssels den Klartext gewinnt. (Brechen, Knacken)

Historische Verschlüsselungen

- Transposition
- Caesar Verschlüsselung
- Polyalphabetische Verschlüsselung
- Vigenére-Quadrat
- ADFGVX
- Enigma

Transpositionschiffre

Skytale

Klartext: GEHEIMNISSESINDSICHER

1		G				I				S				I				I				R
2			E		E			M		I		S		S		N		S		C		E	
3				H					N						E				D				H	

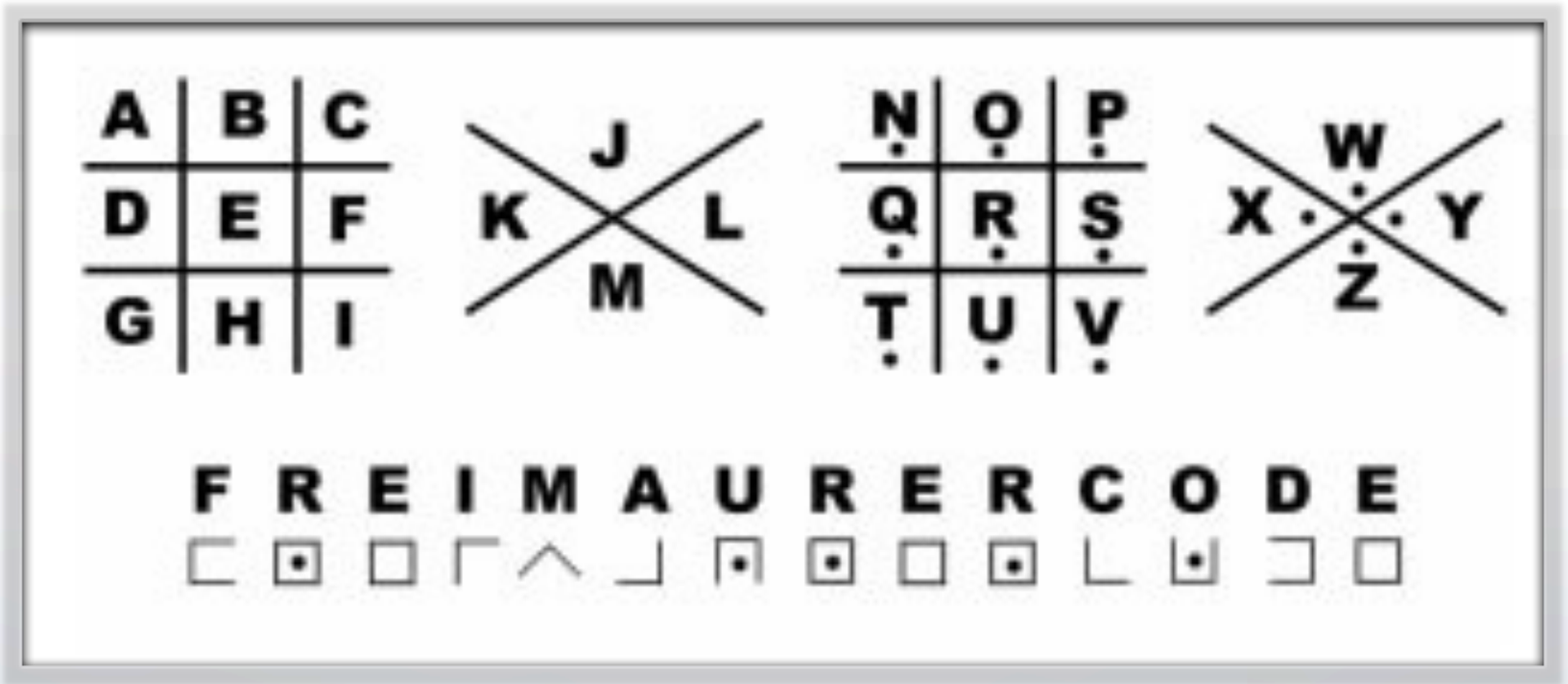
Chiffretext: GISIIREEMISSNSCEHNEDH

Leiter- oder Gartenzaunchiffre

Caesarverschlüsselung

- Monoalphabetische Verschlüsselung
  - Verschlüsselungsscheibe
  - jeder Buchstabe wird um 3 Stellen verschoben
  - A ➜ D, L ➜ O, Y ➜ B

Freimaurercode



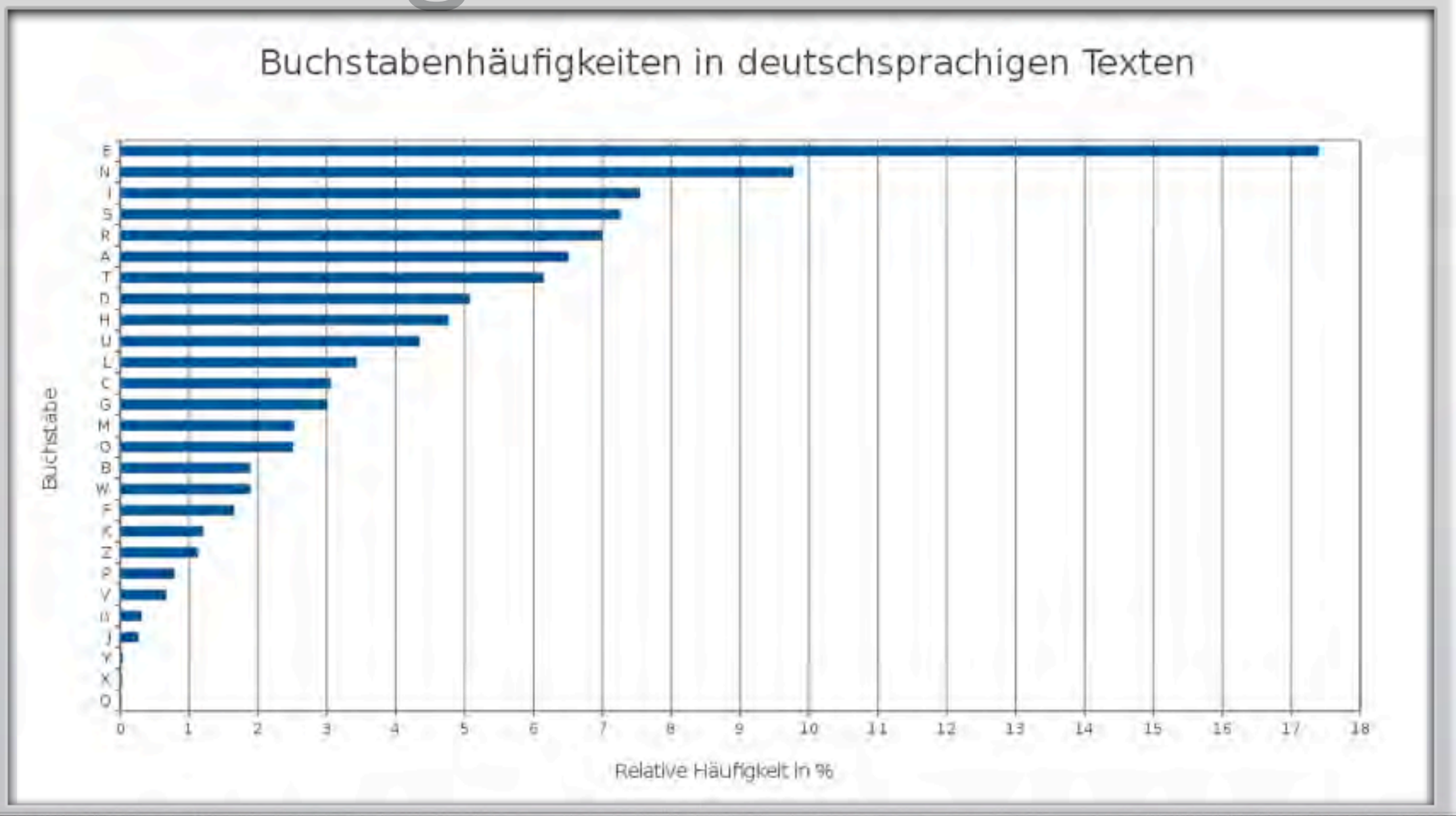
Monoalphabetische Substitution

- jedem Buchstaben wird ein anderer zugegeordnet
- Eher Codierung als Verschlüsselung
- breite Auswahl ähnlicher Verfahren: Freimaurercode, ROT13, ASCII, ...

Problem

- Es werden nur andere Buchstaben oder Symbole verwendet
- die eigentliche Information ist noch vorhanden
- sehr leicht zu entziffern mit einfacher Häufigkeitsanalyse

Häufigkeit: deutsch



Häufigkeitsanalyse

- Vorkommen von Symbolen zählen
- der häufigste ist wahrscheinlich “E” oder “N”
- die häufigsten Bigramme sind “ER”,“EN”, “NN”,“CH”
- Doppelbuchstaben
- nach “Q” kommt fast immer “U”

Problem

- Alle monoalphabetischen Verschlüsselungen haben sich als nicht sicher erwiesen
- Wenn das Verfahren bekannt wird, kann jede Nachricht entschlüsselt und Identität angenommen werden
- Es ist schwer, das Verfahren zu wechseln
- Für n Parteien braucht man n(n-1)/2 Verfahren
- Man muss das Verfahren **vorher** absprechen

Polyalphabetische Verschlüsselung

- Idee: verschiedene Positionen im Klartext werden mit verschiedenen Alphabeten verschlüsselt
- z.B.
  - 1,4,7,10, ... mit A ➜ C (+3)
  - 2,5,8,11, ... mit A ➜ K (+11)
  - 3,6,9,12, ... mit A ➜ H (+8)
- GEHEIMNISS ➜ JPPHTUQTAV

Beobachtung

- GEHEIMNISS ➜ JPPHTUQTAV
- Gleiche Buchstaben werden im Chiffretext nun mit verschiedenen Buchstaben dargestellt
- je mehr Alphabete man benutzt, umso gleichmässiger wird die Verteilung
- Häufigkeitsanalyse funktioniert leider immer noch
  - Man muss den Chiffretext dazu in mehrere Teile aufteilen

Vigenèrequadrat

- nach Blaise de Vingenère, um 1580
- 26 Alphabete, die durch ein Codewort ausgewählt werden
  - Der Buchstabe im Codewort wählt ein Alphabet aus
  - Der Buchstabe im Klartext wird mit dem gewählten Alphabet substituiert

Viegenérequadrat

Klartext	F	R	E	I	H	E	I	T																	
Schlüsselwort	G	E	H	E	I	M	N	I	S	S															
Chiffretext	L	V	L	M	P	Q	V	B					=ZEICHEN(REST(CODE(B1)-65+CODE(B2)-65; 26)+65)												
	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

Beobachtung

- Polyalphabetische Verschlüsselung
- je länger das Codewort ist, umso schwerer wird eine Häufigkeitsanalyse
- Das Codewort zu wechseln ist leicht
- Der Chiffretext hängt nun vom Klartext, den Alphabeten und vom Schlüsselwort ab, statt nur vom Verfahren

Angriff auf Vigenère

- einen wahrscheinlichen Klartextteil raten ("Crib", Spickzettel)
- den Crib über den Chiffretext legen und rückwärts entschlüsseln
- Wenn etwas sinnvolles rauskommt, könnte das ein Teil des Schlüssels sein

Folgerungen

- Je länger der Schlüssel, umso besser
- Je mehr Alphabete benutzt werden, umso besser
- Je zufälliger der Schlüssel ist, umso besser

Perfekte Verschlüsselung

- One Time Pad
  - Ein Schlüssel, der absolut zufällig und genauso lang wie die Nachricht ist, wird benötigt
  - Die Nachricht wird codiert, jeweils ein Zeichen des Klartexts mit einem Zeichen des Schlüssels
  - Der Chiffretext weist keine der bisherigen Probleme auf, da der Schlüssel zufällig ist, ist jeder Klartext gleich wahrscheinlich

OTP: In der Praxis untauglich

- Spontane Kommunikation ist nicht möglich
- "n" Parteien brauchen n(n-1)/2 OTPs
- Der OTP muss **vorher verteilt** werden und muss absolut geheim bleiben
- Guten Zufall in grossen Mengen herzustellen ist schwer
- Der OTP darf nur einmal benutzt werden

Alternative?

- OTP sicher, aber in der Praxis nicht brauchbar
- Polyalphabetische Substitution ist nicht sicher
- Transposition ist nicht sicher
- ➜ Kombination aus Substitution und Transposition?

ADFGVX

- Wurde im 1. Weltkrieg verwendet
- Die Chiffrierung besteht aus 2 Stufen
  - eine Substitutionsstufe, die Buchstaben des Klartexts in Bigramme verwandelt
  - eine Transpositionsstufe, die die Zeilenweise aufgeschriebenen Bigramme Spaltenweise umsortiert
- Die umsortierten Buchstaben werden durch Morsezeichen übertragen

Substitutionsstufe

	A	D	F	G	V	X	FREIHEIT
A	W	7	J	D	1	G	XV GX VV DV GD VV DV GF
D	L	A	V	U	I	P	
F	B	2	K	C	0	Z	
G	M	H	T	6	O	R	
V	X	3	9	N	E	8	
X	G	4	S	Y	F	5	

- Die Substitutionstabelle ist gegeben oder wird durch ein Codewort in der ADFGVX-Matrix erstellt
- Für jeden Buchstaben des Klartexts wird ein Bigramm abgelesen
- “FREIHEIT” ➜ “XV GX VV DV GD VV DV GF”

Transpositionsstufe

- Die erste Stufe wird zeilenweise aufgeschrieben
- Das Codewort (CODE) wird in den Kopf geschrieben
- Die Spalten werden alphabetisch sortiert (CDEO)
- Der Chiffretext wird zeilenweisen notiert

XV GX VV DV GD VV DV GF

Transpositionsschlüssel: CODE

C	***O***	***D***	***E***			C1	O4	D2	E3
X	V	G	X						
V	V	D	V						
G	D	V	V						
D	V	G	F						
C	***D***	***E***	***O***						
X	G	X	V						
V	D	V	V						
G	V	V	D						
D	G	F	V						

XGXV VDVV GVVD DGFV

- Die eigentliche Sicherheit liegt hier in der Transposition, die Substitution bietet nur geringe Sicherheit

- Bemerkenswert:

  - Kombination von Substitution und Transposition

    - **Ableitung der Schlüssel** aus Codeworten

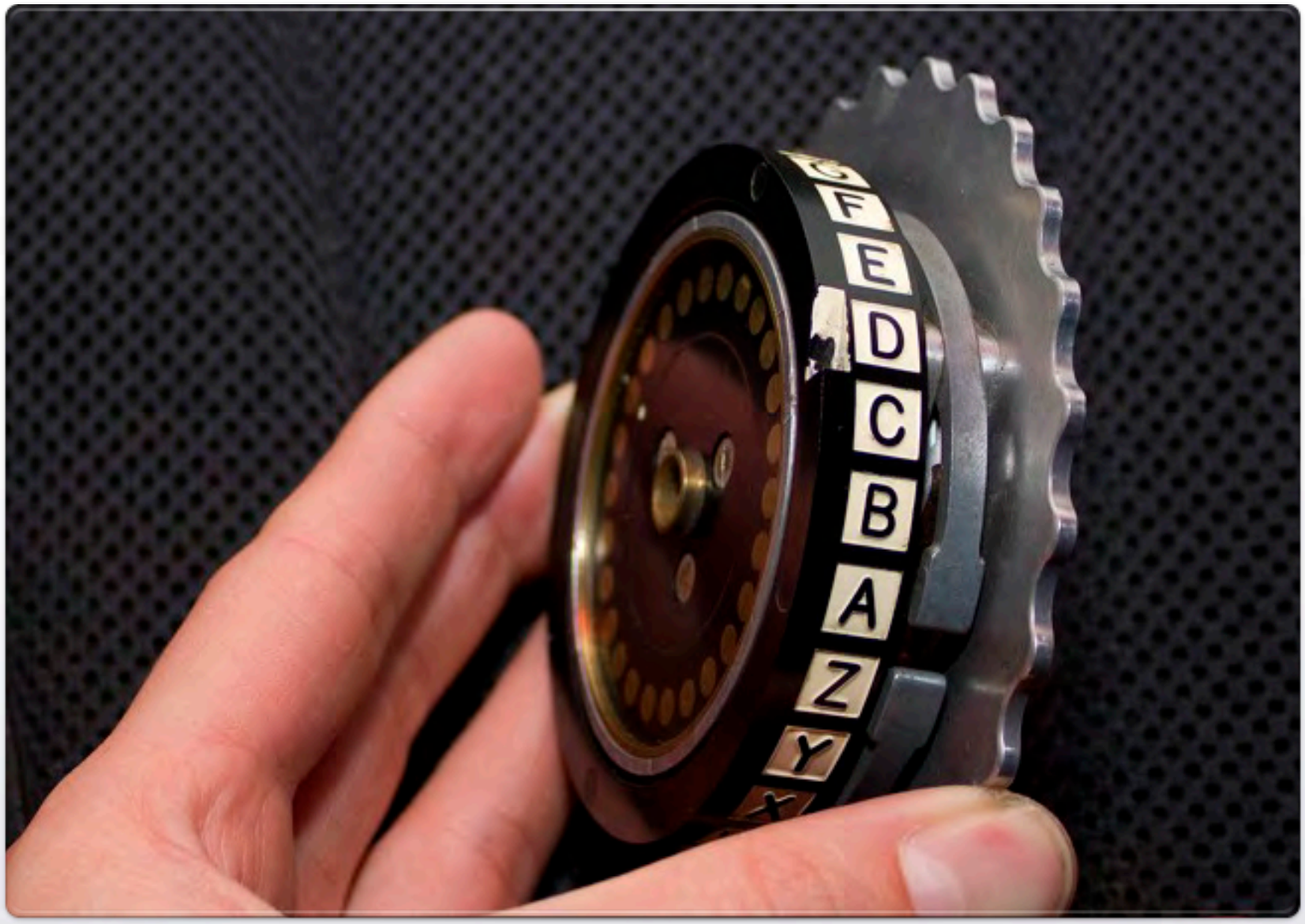
    - Schwäche: Die **Länge des Schlüssels** ist begrenzt

- Ab 1. März 1918 im Einsatz, im April 1918 bereits geknackt

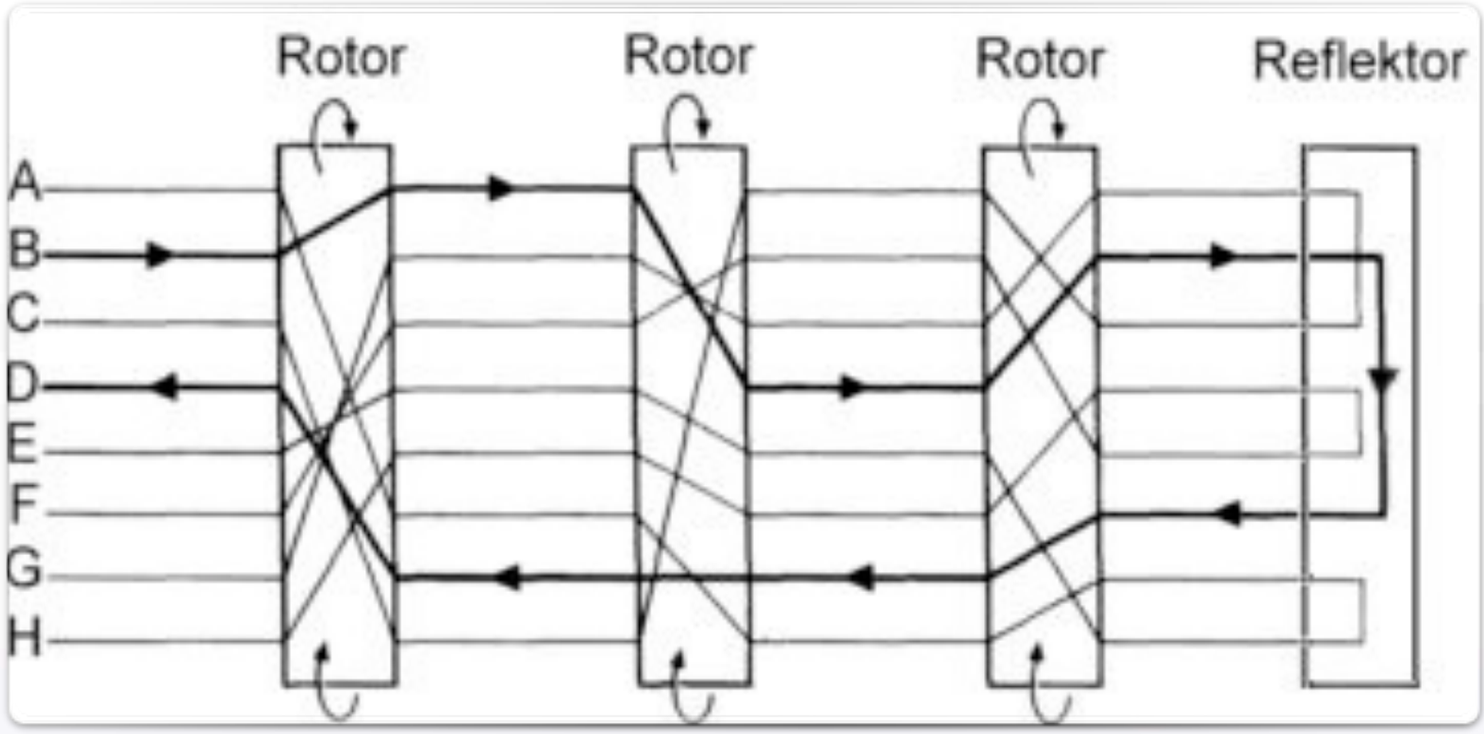
Der nächste Schritt: Enigma

- Idee:
  - Ein Rotor mit 26 Kontakten auf jeder Seite
  - Im Rotor sind die Kontakte zufällig aber bekannt durchkontaktiert
  - Eine Maschine mit Tastern, Lämpchen und mehreren Rotoren
  - Wenn man auf einen Taster drückt, geht ein Lämpchen an
  - Die Rotoren bewegen sich nach jedem Zeichen weiter

Enigmarotor / Walze



Enigma



- 3 Rotoren und ein Reflektor
- durch den Reflektor wird die Dechiffrierung mit demselben Gerät ermöglicht

Alphabetgenerator

- ein Rotor erzeugt eine monoalphabetische Substitution
- Das Alphabet bildet sich durch die Hintereinanderschaltung der Rotoren
- Damit hat man 26*26*26 Alphabete (17576)
- Der “Schlüssel” wird durch die Anfangsstellung und die Reihenfolge der Rotoren gebildet

Enigma

- Die Kombinatorik ist eigentlich viel größer, zu erkennen:
  - Tastenfeld
  - Lampenfeld
  - Steckfeld mit 6 Kabeln
  - 3 Rotoren = 6 Lagen
  - “Ringlage”
- etwa 10^16 mögliche Schlüssel

Verschlüsselung

- im Codebuch wird der Tagesschlüssel nachgeschlagen und eingestellt
  - Walzenlage, Walzenposition, Ringlage, Steckfeld
- Der Funker denkt sich einen Nachrichtenschlüssel aus, z.B. EXU
- Der Nachrichtenschlüssel wird ZWEIMAL eingegeben
- Der Nachrichtenschlüssel wird als Walzenposition eingestellt
- Die Nachricht wird eingegeben

Geheim! **Sonder-Maschinenschlüssel BGS** JANUAR 1939

Datum	Walzenlage	Rinstellung	Steckerverbingerdungen	Kenugruppen
31	IV II I	18 16 14	AC BL DN EF GJ IS PX QV RZ WY	VLB BGC OSK UHH
30	III II I	03 10 02	AD BE CM FQ IR JK LN OP UZ WY	SXX WYU PYO SQO
29	V II I	08 19 09	AC BH DV EI FM GY KR LW NU QZ	POS QTV SNO LIQ
28	III V II	07 20 11	BU CY ET FW GH IR JK LS OP VZ	IAH HDO DVQ ABK
27	I IV II	11 11 11	AT BX CN DY FL HU JP KR QV SZ	ZHG SMG NVR KZD

- Durch die Rotoren entsteht ein gewaltig langer Zyklus verschiedener Alphabete

- Ohne Kenntniss des Schlüssels ist eine Vorhersage des nächsten Alphabets aus der Historie faktisch unmöglich (PRNG!)

- während einer Nachricht tritt praktisch keine Wiederholung eines Alphabets auf

- Die Enigma bietet damit im Prinzip eine hervorragende Sicherheit

Enigma: Erkenntnisse

- viel Raum für Kombinatorik
- der “Tagesschlüssel” wird nicht zum Verschlüsseln benutzt, sondern initialisiert die Maschine
- Für die Nachricht wird ein Sessionkey verwendet (“Nachrichtenschlüssel”)
- Trotz bekannter Funktionsweise hielt die Maschine der Analyse lange Stand (Kerckhoff-Prinzip)

- Leider sammeln sich viele kleine Fehler
  - technische Einschränkungen die den Schlüsselraum verkleinern
  - Angriffsmöglichkeiten durch die Art der Benutzung (Cribs, doppelter Nachrichtenschlüssel, vorhersagbare Nachrichten, gestohlene Codebücher, Whistleblowing)
- Durch große Anstrengung und viele kluge Leute wurde die Enigma im Krieg geknackt. Ganz wesentlich verbunden mit der Entzifferung der Enigma ist Alan Turing

Kryptographie im Computerzeitalter

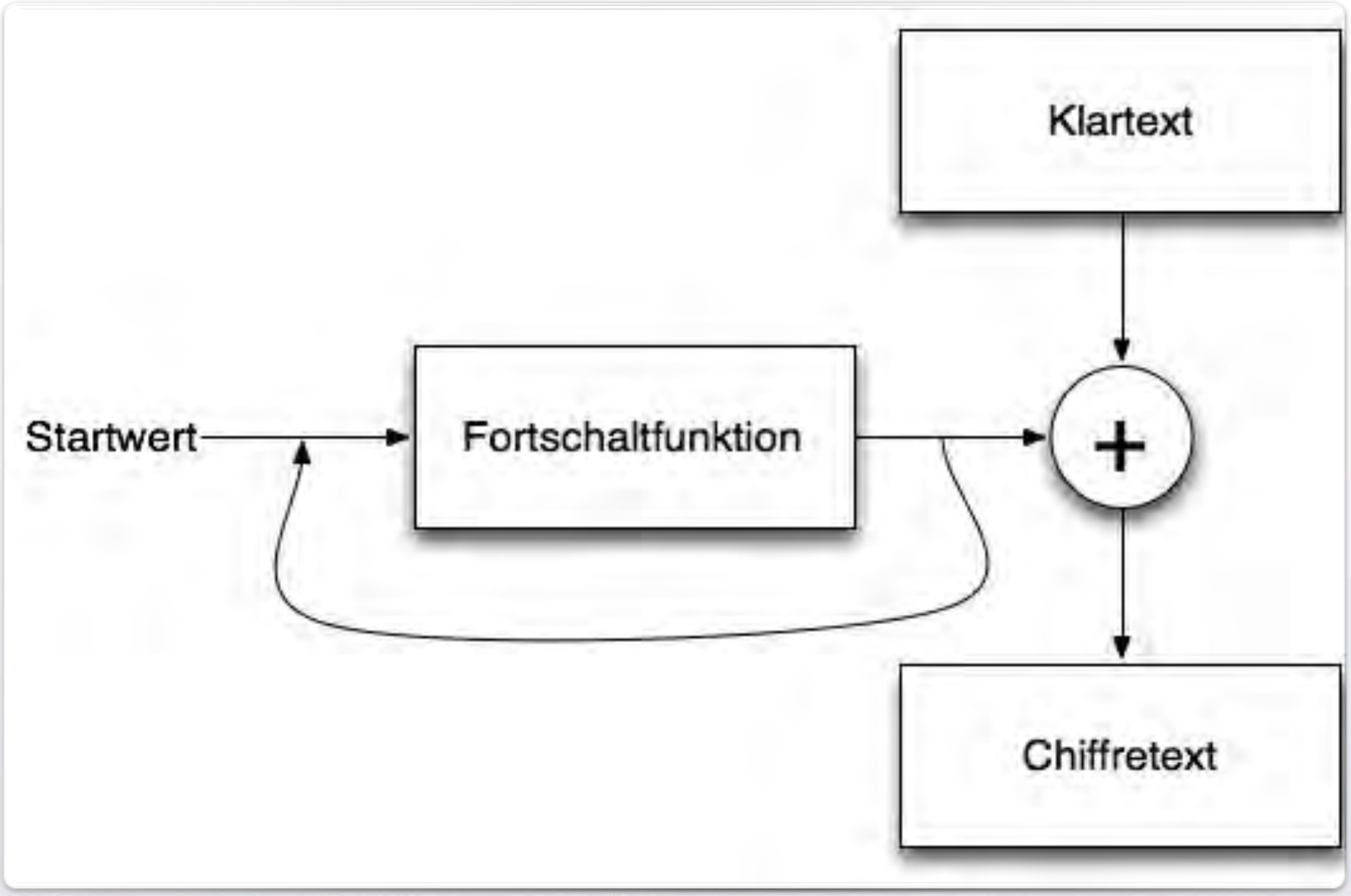
Computerkryptographie

- Elektronische Rechenmaschinen
  - sind schneller und flexiber als mechanische
  - können auch mathematisch schwierige Operationen durchführen
  - können zum Brechen von Kryptographie benutzt werden
- eine "elektronische Enigma" könnte problemlos Hunderte Rotoren haben, die sich nach kompliziertesten Schemen getauscht und vorwärts, rückwärts oder gar nicht bewegen

Das Beste aus Enigma und OTP

- Ein Pseudozufallszahlengenerator wird mit einem Schlüssel initialisiert (Enigma)
- Die Fortschaltfunktion erzeugt einen Schlüsselstrom (Enigma), ohne sich wiederholende Sequenzen (OTP)
- Der Schlüsselstrom wird mit dem Klartext zum Chiffretext verknüpft (OTP)
- zum Entschlüsseln wird derselbe Schlüsselstrom erzeugt und die Verschlüsselung umgekehrt

Stromchiffre



Modernes Verfahren: RC4

- Ron Rivest 1987
- Rivest Cipher 4 oder Ron's Code 4 :-) oder ARCFOUR
- 1987 entwickelt und geheim gehalten, 1994 anonym publiziert, seitdem "inoffiziell Open Source"
- verwendet in WEP (Wired Equivalence Privacy)

RC4 Initialisierung

- Es wird eine 255 Byte große S-Box benutzt (Substitutionsalphabet)
- Die S-Box wird aufsteigend mit s[i] = i belegt
- jedes Zelle der S-Box wird mit einer anderen vertauscht
- welche das ist hängt vom Schlüssel ab ➜

RC4 Verschlüsselung

- Für jedes zu verschlüsselnde Zeichen:
  - eine Schlüsselvertauschung wird vorgenommen
  - nach einem festen Muster wird ein (Pseudo-)Zufallszeichen aus der S-Box entnommen
  - Klartextzeichen wird damit XOR verknüpft

Sourcecode RC4

```
for (i=0; i<256; i++) {
        s[i] = i;
}
for (i=0, j=0; i<256; i++) {
        j = (j + s[i] + key[i % l]) % 256;
        SWAP(s[i], s[j], tmp)
}
i = 0; j = 0;
while (!feof(stdin)) {
        p = getc(stdin);
        i = (i + 1) % 256;
        j = (j + s[i]) % 256;
        SWAP(s[i], s[j], tmp)

        rnd = s[ (s[i] + s[j]) % 256 ];
        putc(p ^ rnd, stdout);
}
```

```
rene — rene@mail: ~ — bash — 53×11
kallisti:~ rene$ echo "Nachricht" | ./rc4 "secretkey"
 > cipher
kallisti:~ rene$ od -x cipher
0000000      c9eb    74bf    d83e    20f6    a65c
00b4
0000013
kallisti:~ rene$ cat cipher | ./rc4 "secretkey"
Nachricht
?{kallisti:~ rene$
```

RC4 Entschlüsselung

- Genauso wie die Verschlüsselung
- Schlüssel zur Verschlüsselung und Schlüssel zur Entschlüsselung sind **identisch**
- Verfahren dieser Art nennt man daher "**symmetrisch**"
- Der PRNG erzeugt den gleichen Schlüsselstrom und XOR rekonstruiert den Plaintext

Ansatz zum Brechen

```
kallisti:~ rene$ echo "Wichtige Nachricht" | ./rc4 "secretkey" | od -x
0000000      c1f2    74bf    d838    2df2    e208    e72a    55ea    485c
0000020      231c    9f03
0000024
kallisti:~ rene$ echo "Wichtige Nachricht" | ./rc4 "Secretkey" | od -x
0000000      8034    311d    f007    92f5    3a4f    a380    eec1    fc28
0000020      340c    01a3
0000024
kallisti:~ rene$ echo "Wichtige Nachricht" | ./rc4 "Xecretkey" | od -x
0000000      9b88    7834    a6a3    2d60    b47e    c886    7bd4    af0e
0000020      fd48    c796
0000024
kallisti:~ rene$
kallisti:~ rene$ echo "WichTige Nachricht" | ./rc4 "Xecretkey" | od -x
0000000      9b88    7834    a683    2d60    b47e    c886    7bd4    af0e
0000020      fd48    c796
0000024
kallisti:~ rene$ echo "Wichtige NAchricht" | ./rc4 "Xecretkey" | od -x
0000000      9b88    7834    a6a3    2d60    b47e    c8a6    7bd4    af0e
0000020      fd48    c796
```

Änderung am Schlüssel ändert den ganzen Strom (gut)
Änderung am Klartext ändert genau den betreffenden Teil im Chiffretext (schlecht)

Stromchiffren

- Jedes Zeichen wird einzeln verschlüsselt
- Ein Klartextzeichen wirkt sich auf genau ein Chiffretextzeichen aus
- Es besteht Verbesserungsbedarf
  - Rückwirkung des Klartextstroms auf den Schlüsselgenerator
  - Schritt zu Blockchiffren

Blockchiffren

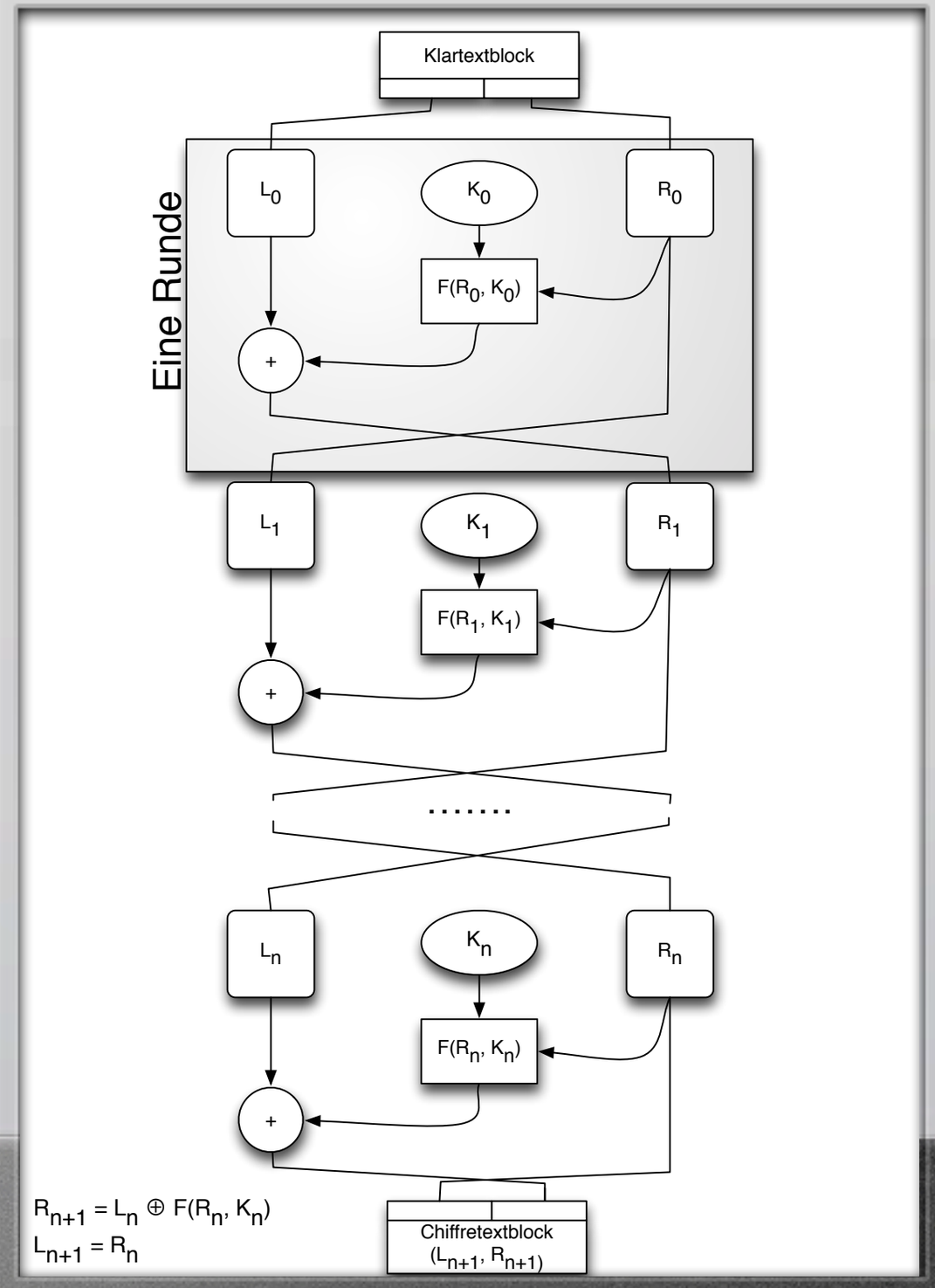
- Die Nachricht wird in Blöcke aufgeteilt (z.B. 64 Bit)
- Die Verschlüsselung wird auf den gesamten Block angewendet
- Muster im Klartext können besser verwischt werden
- Blockchiffren lassen sich mathematisch schlechter analysieren (gut für die Sicherheit)

Feistelchiffren

- Klartextblock wird in zwei Teile geteilt
- In jeder Runde werden beide Teile kombiniert und mir dem Schlüssel verrechnet
- Am Ende wird der Chiffretextblock aus den beiden Teilblöcken zusammengesetzt
- DES hat 64 Bit und 16 Runden, Twofish hat 128 Bit und 16 Runden

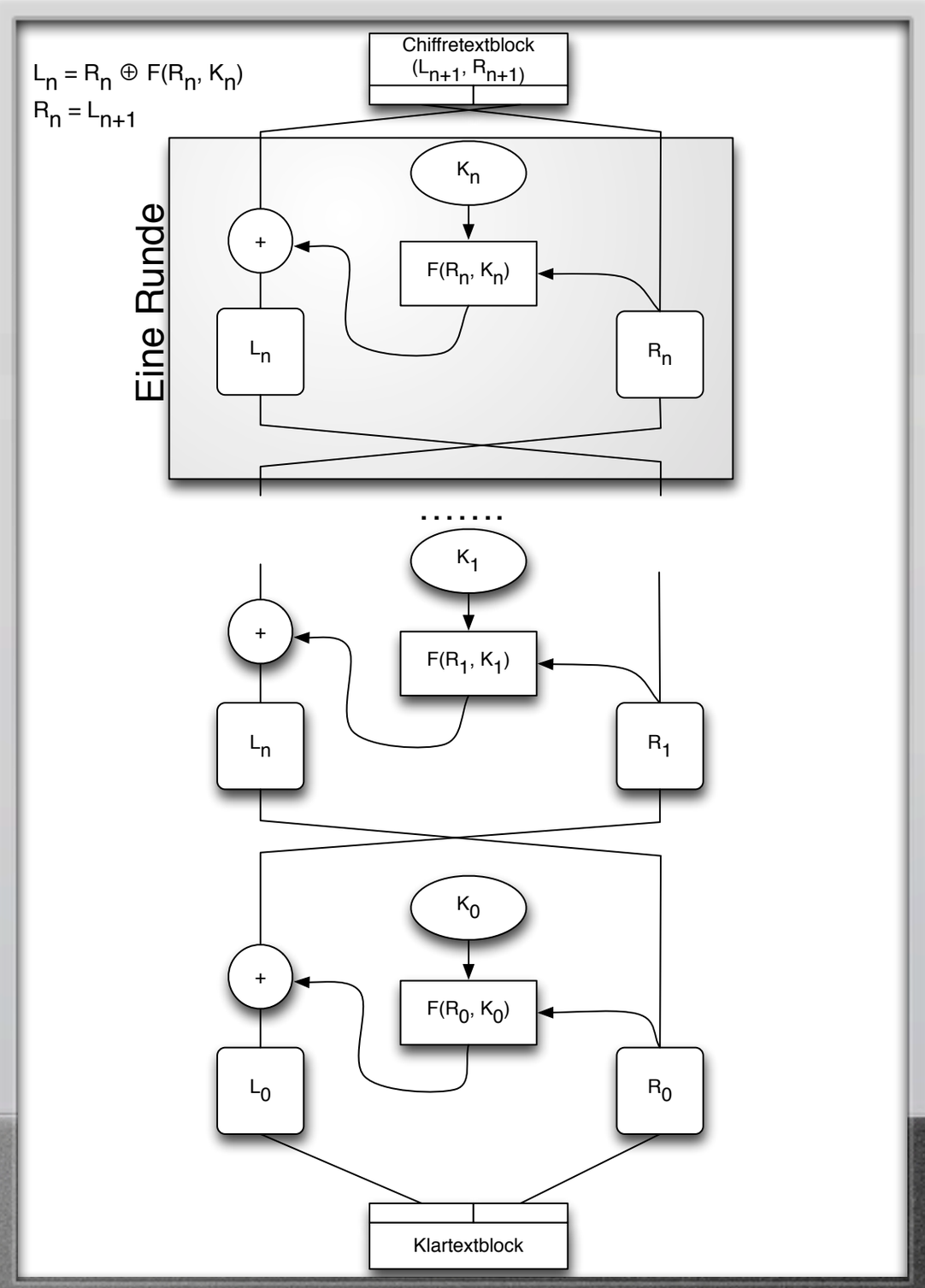
Prinzip: Verschlüsseln

- jede Runde berechnet eine neue “rechte” Seite aus “linker” Seite ver-X-odert mit dem Teilschlüssel
- die neue “linke” Seite ist die alte “rechte” Seite

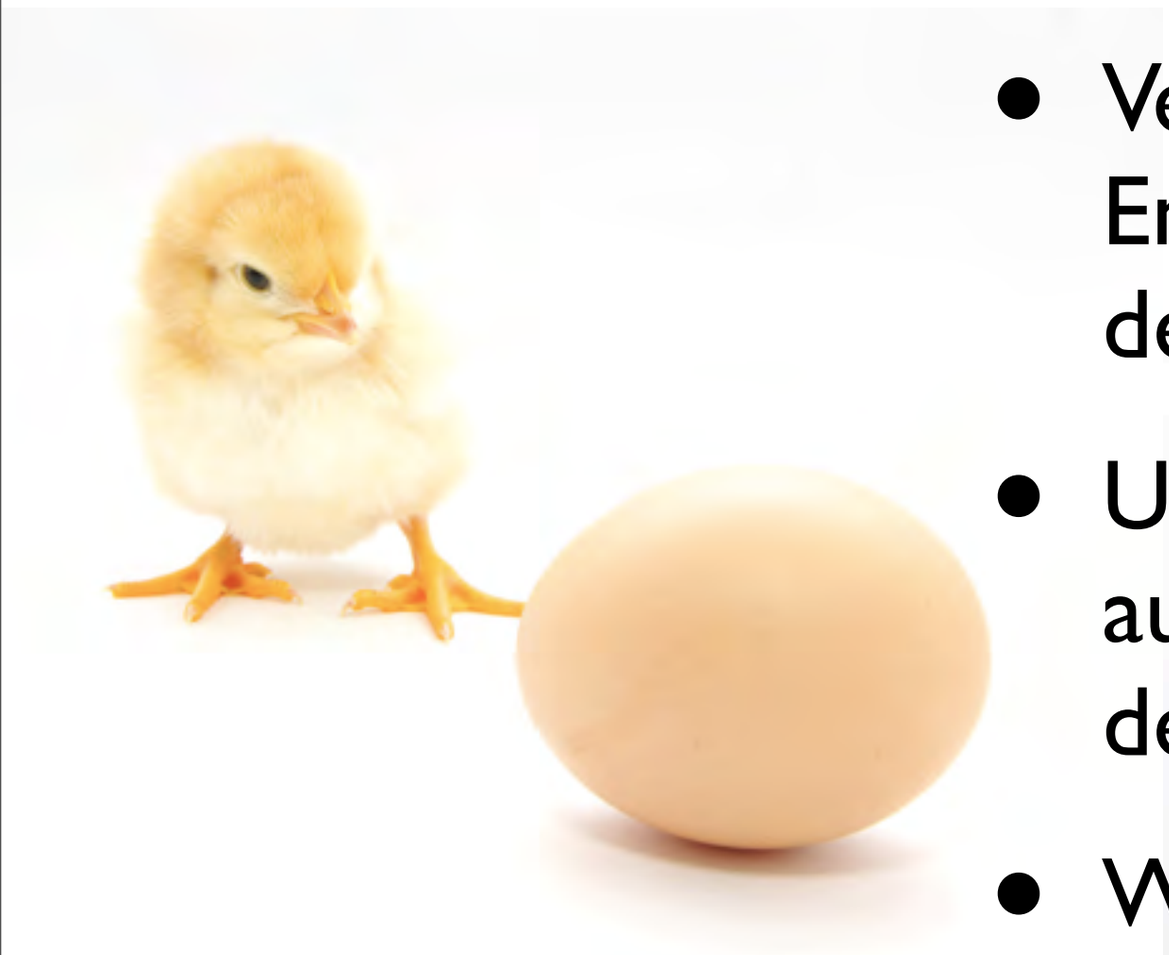


Prinzip: Entschlüsseln

- Jede Runde berechnet eine Iteration rückwärts
- Die Funktion “F” ist dieselbe wie beim Verschlüsseln
- Den Teilschlüssel $F(R_n,K_n)$ braucht man zuerst
- R_n kommt direkt aus dem Chiffretext, das ist L_{n+1}



- Nach einigen Runden haben sich praktisch alle Klartextbits auf alle Chiffretextbits ausgewirkt
- Das Feistelnetzwerk ist garantiert entschlüsselbar, da die Struktur umkehrbar ist
- Sogar dann, wenn die Funktion F nicht umkehrbar ist
- Die Stärke der Verschlüsselung liegt in einem guten Schlüsselgenerator, denn offensichtlich braucht man viel Schlüsselmaterial



- Problem ist bisher immer wieder:
  - Verschlüsselung und Entschlüsselung geschehen mit demselben Schlüssel
  - Um verschlüsselte Daten auszutauschen muss der Empfänger den Schlüssel bereits besitzen
  - Wie stimmen sich Sender und Empfänger über den gemeinsamen Schlüssel ab?

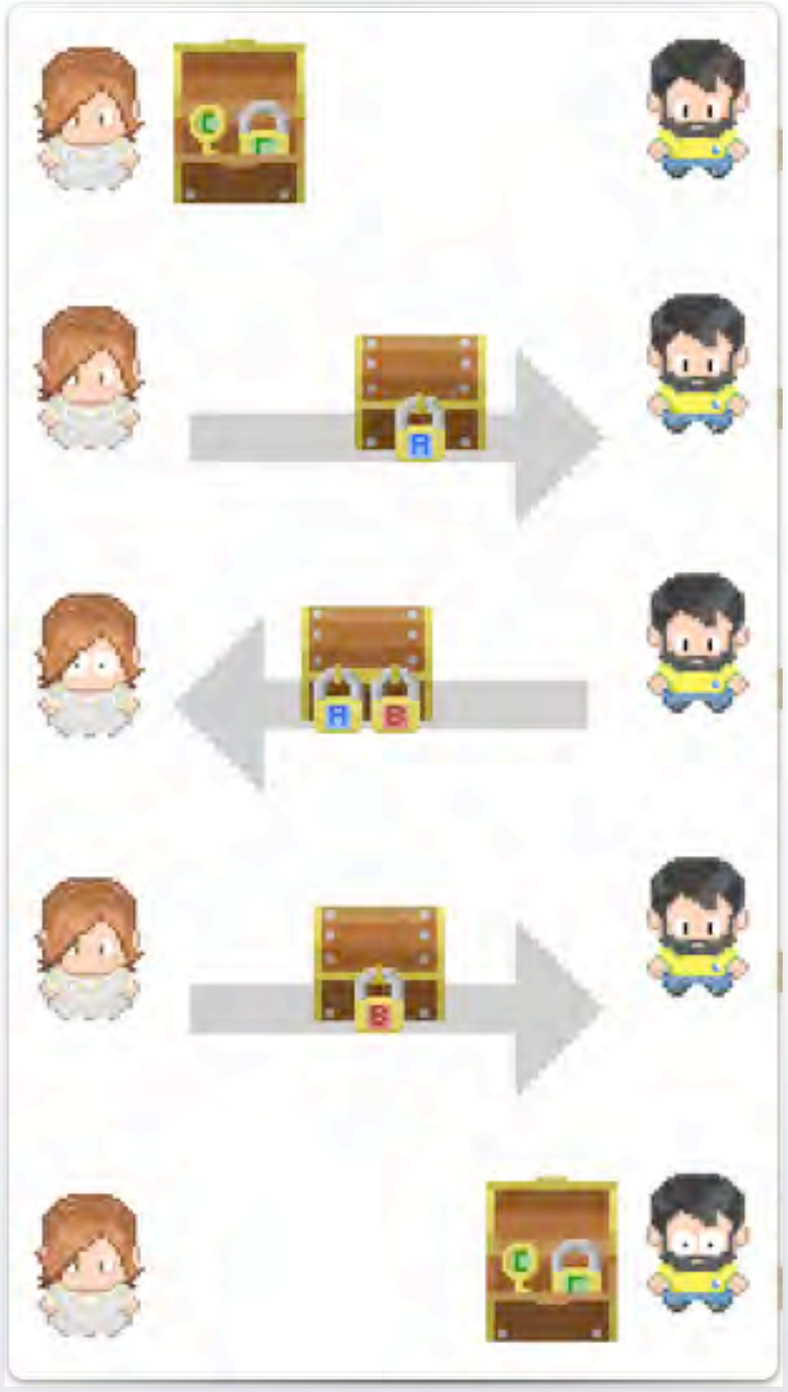
Schlüsselaustausch

Geheimnisse und unsichere Kanäle

- Allgemeinere Frage:
  "Können zwei Parteien ein Geheimniss (z.B. einen Schlüssel) über einen unsicheren Kanal übertragen, ohne vorher ein Geheimniss abzustimmen?"
  - (Der Angreifer kann alles sehen aber nichts manipulieren)

Witfield Diffies Trick:

- Alice hinterlegt ein Geheimniss in der Kiste
- Die Kiste wird mit Alices Schloss gesichert und zu Bob geschickt
- Die Kiste wird mit Bobs Schloss gesichert und zu Alice geschickt
- Alice entfernt ihr Schloss
- Bob entfernt sein Schloss und entnimmt das Geheimniss aus der Kiste



Problem

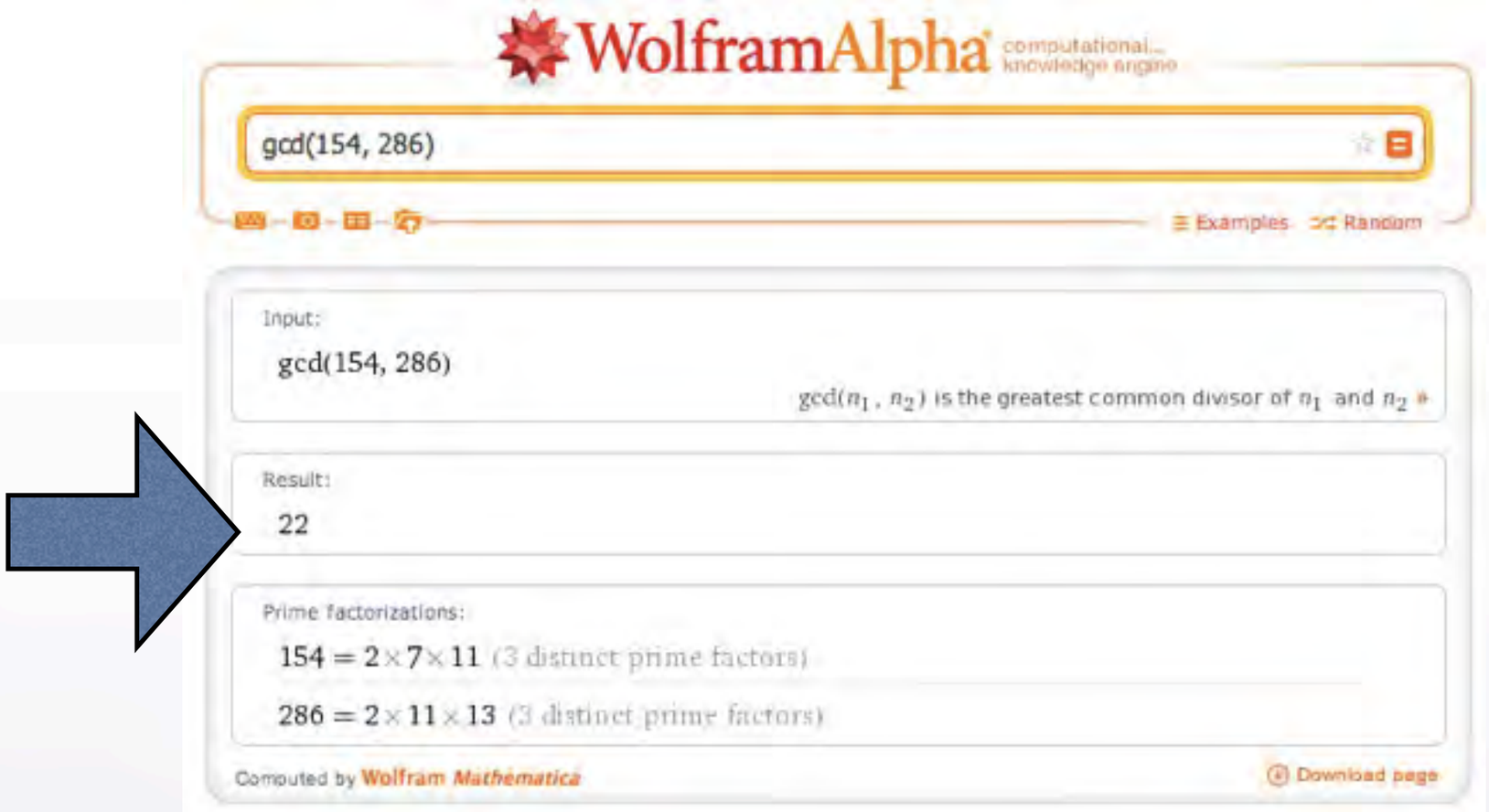
- Das Verfahren erfordert einen Verschlüsselung, bei der:
  - $DEC_B(DEC_A(ENC_B(ENC_A(M)))) = M$

    ENC_X: Encrypt Schlüssel X, DEC_X: Decrypt Schlüssel X, M: Message
  - “letztes-erstes”
- Leider ist das bei keiner der (damals) bekannten mathematischen Methoden gegeben

Theoretische Durchführung

- Alice hat ein Geheimniss: 22 und denkt sich eine Zahl aus: 7
- Alice multipliziert das Geheimniss mit der Zahl: 22*7=154
- Alice sendet Bob: 154
- Bob denkt sich eine Zahl aus und multipliziert: 13*154 = 2002
- Bob sendet Alice: 2002, Alice berechnet: 2002 / 7 = 286
- Alice sendet Bob: 286, Bob berechnet: 286 / 13 = 22

Leider nicht sicher ...



- Die Berechnung des kleinsten gemeinsamen Teiler (gcd) ist mit dem euklidischen Algorithmus auch für sehr große Zahlen effizient durchführbar

Einwegfunktion (Trap Door Function)

- Man braucht eine Funktion, die
  - vorwärts leicht berechenbar ist
  - rückwärts nicht trivial berechnet werden kann
- Nachdem Witfield Diffie, Martin Hellmann und Ralph Merkle über zwei (!) Jahre daran arbeiten findet Martin Hellman eine mathematische Lösung: den **diskreten Logarithmus**

Diffie-Hellman Schlüsselaustausch

- Man nehme:
  - p: eine prime Restklasse (Z/pZ)*
  - g: eine Primitivwurzel der Klasse (Erzeuger)
  - a und b: zwei geheime, zufällige Zahlen
- Alice wählt a und sendet p,g und $A = g^a \pmod p$ an Bob
- Bob wählt b, sendet $B = g^b \pmod p$ an Alice und berechnet: $K = A^b \pmod p$
- Alice berechnet aus Bobs B und a: $K = B^a \pmod p$
- Klar, denn $A^b = (g^a)^b = g^{a*b} = g^{b*a} = (g^b)^a = B^a$ (alles mod p)
- a und b lassen sich aus p, g, A und B nicht effizient bestimmen, denn dazu braucht man den diskreten Logarithmus

Exkurs: Restklassen

- Teilt man Zahlen mit Rest z.B. durch 3, so kommen Reste von 0, 1 oder 2 vor
- 5 : 3 = 1 Rest 2 (1*3 + 2)
- 31 : 3 = 10 Rest 1 (10*3 + 1)
- Alle Zahlen, die beim Teilen durch 3 den Rest 2 lassen (2, 5, 8, 11, ...), haben die Form x*3 + 2
- Man schreibt: **5 = 2 (mod 3)**

Exkurs: Restklassen

- 2 (mod 3) ist der Platzhalter (Repräsentant) für viele Zahlen, nämlich alle, die beim Teilen durch 3 den Rest 2 lassen
- beim Teilen durch 3 gibt es genau 3 Resteklassen: 0, 1 und 2
- 5 und 8 gehören zu derselben Klasse, nämlich 2 (mod 3)
- Unter bestimmten Bedingungen kann man in Restklassen "normal" addieren und multiplizieren:
  - 2 + 2 = 1 (mod 3), denn 2 + 2 = 4 (mod 3) = 1 (mod 3)
  - 2*3*4 = 4 (mod 5), denn 2*3*4 = 24 (mod 5) = 4*5+4 (mod 5) = 4 (mod 5)
  - 21+10 = 7 (mod 24), klar, denn 10h nach 21:00 Uhr ist ja 7:00 Uhr

Exkurs: Erzeuger

- Ein Erzeuger ist eine Zahl, deren Potenzen alle Elemente der Klasse bilden
- In primen Restklassen sind alle Elemente Erzeuger
- Potenzieren in primer Restklasse (Z/7Z)*:
  - $y = 5^3 \pmod 7 = 5^2*5 = 25*5 = 4*5 = 20 = 6 \pmod 7$
  - $25 = 3*7+4$, $20 = 2*7+6$
- Auch in Restklassen gilt:
  - $a^{m+n}=a^m*a^n \pmod p$
  - $a^{mn} = a^{m*n} = a^{n*m} = a^{nm} \pmod p$

Exkurs: Diskreter Logarithmus

- Der diskrete Logarithmus ist die Umkehrung der Potenz in einer Resteklasse
- Klar: $x = 5^3 \pmod 7$ auszurechnen ist leicht (eben gemacht)
- $5^x \pmod 7 = 1$ auszurechnen ist nicht so leicht
- “Ich habe den Wecker von 12:00 mehrmals um 7 Stunden weitergestellt. Jetzt steht er auf 6:00. Wie oft habe ich gedreht?”

Exkurs: Diskreter Logarithmus

- Das Potenzieren ist nicht umkehrbar.
- Einwegfunktion, wie beim Farbe mischen
  - Man kann Farbe nicht mehr entmischen (Logarithmus)
  - Man kann denselben Farbton nur erhalten, wenn man neu mischt und vergeicht (Brute Force)

Diffie-Hellman Schlüsselaustausch

- p: eine prime Restklasse $(Z/pZ)^*$
- g: eine Primitivwurzel der Klasse (Erzeuger)
- a und b: zwei geheime, zufällige Zahlen

- Alice wählt a und sendet p, g und $A = g^a \pmod p$ an Bob
- Bob wählt b, sendet $B = g^b \pmod p$ an Alice und berechnet $K = A^b$
- Alice berechnet $K = B^a$

Diffie-Hellmann

- $A = g^a \pmod p$, $B = g^b \pmod p$
- Alice kennt g, p, a und B: $B^a = (g^b)^a = g^{ba} = \mathbf{g^{ab} \pmod p}$
- Bob kennt g, p, b und A: $A^b = (g^a)^b = \mathbf{g^{ab} \pmod p}$
- Ein Angreifer kennt A, B, g und p und kann $g^a g^b = g^{a+b}$ oder B^A oder A^B berechnen, aber **nicht** $g^{ab} \pmod p$
- Um an a oder b zu kommen, braucht man den Diskreten Logarithmus. Das geht bei großen Zahlen nur mit exorbitant hohem Rechenaufwand.

Ergebnis

- Mehrere nicht-geheime Nachrichten erzeugen einen gemeinsamen Schlüssel
- Ungeeignet für Mails und spontane Kommunikation, da Alice zum Senden der Nachricht eine direkte Antwort von Bob benötigt

Public Key Kryptographie

Asymmetrisch Verfahren

- Es gibt verschiedene Schlüssel
  - Mit einem Schlüssel wird verschlüsselt.
  - Mit dem anderen wird entschlüsselt
  - Der Dechiffrierschlüssel kann nicht aus dem Chiffrierschlüssel berechnet werden

Asymmetrisch Verfahren

- Der Chiffrierschlüssel
  - ist nicht weiter schützenswert
  - wird deswegen als “Public Key” bezeichnet
- Der Dechiffrierschlüssel
  - sollte nur dem Empfänger bekannt sein
  - wird dewegen als “Private Key” bezeichnet

Public Key Kryptographie

- Anschauliches Beispiel:
  - Ein öffentlicher Schlüssel: Telefonbuch von Berlin, sortiert nach Nachnamen. Etwa 3 Millionen Einträge, 8000 Seiten
  - Ein privater Schlüssel, Telefonbuch von Berlin, sortiert nach Telefonnummern. Ebenfalls 3 Millionen Einträge, 8000 Seiten

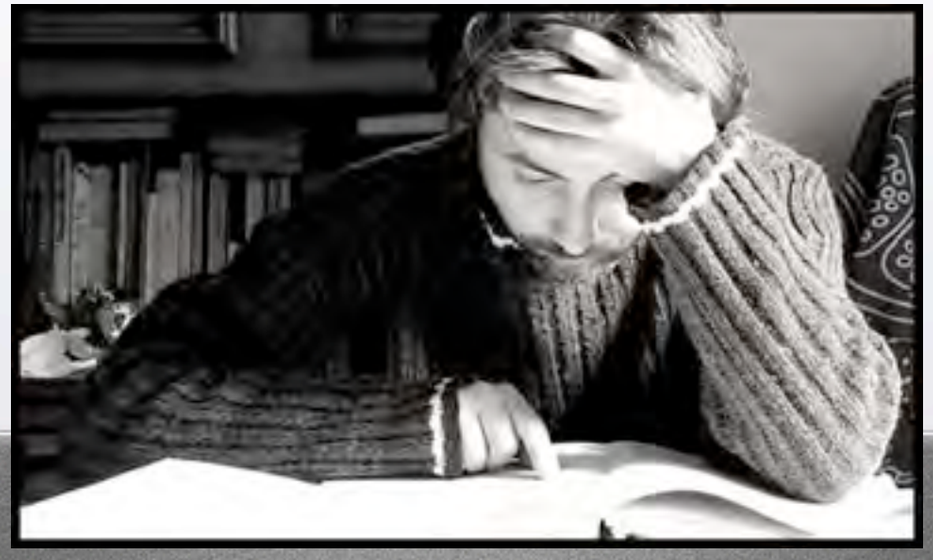
Telefonbuchverschlüsselung

- Verschlüsseln von “RSA” mit dem öffentlichen Schlüssel kann jeder ohne weiteres, es dauert keine 10 Sekunden:
  - **Rabe**, Heinz: 030-563097
  - **Schwalbe**, Gerda: 030-774239
  - **Adler**, Günter: 030-922386
- Chiffretext: 563097 774239 922386

Telefonbuchentschlüsselung

- Dechiffrierschlüssel: Ein Spezialtelefonbuch, welches nach Telefonnummern sortiert ist
- Entschlüsselung: **563097** 774239 922386 210563
- 563095: Meise, Bernd
- 563096: Huhn, Bert
- **563097**: **Rabe**, Heinz
- 563098: Spatz, Helge

- Der Angreifer hat nur das normale Telefonbuch
- Annahme: Ohne exorbitant viel Arbeit kann man aus ein normalen Telefonbuch keines für die Rückwärtssuche machen
- Ohne den “privaten Schlüssel” muss für jeden Buchstaben von vorn angefangen und jede Nummer verglichen werden



Extraspaß: Nichtexistente Nummern

RSA Verfahren

- Handwerkszeug:
  - Faktorisierung
  - Eulersche φ-Funktion
  - Kleiner Fermatscher Satz

RSA

- Man wähle p und q prim und berechnet n=p*q und nennt es den Modulus
- Man berechne φ(n) einfach als (p-1)(q-1)
- Man wähle eine Zahl e teilerfremd zu φ(n), Encryptionkey
- Man berechne das multiplikative Inverse zu e genannt d, Decryptionkey, so dass e*d = 1 (mod n)
- mit Wissen von φ(n) ist das kein Problem.

RSA

- Man **vernichtet** p, q, und φ(n)
- Verschlüsseln: Sei M der Klartext, dann ist C = M^e (mod n) der Chiffretext
- Entschlüsseln: $C^d = (M^e)^d$ mod n = M^{ed} mod n = M, denn e*d = 1 (mod n)
- um p zu q berechnen, muss man n **faktorisieren** und mit φ(n) und e könnte man d berechnen

Nachteil asymmetrischer Kryptographie

- RSA ist sehr rechenaufwändig, weil viel potenziert wird
- Public Key Verfahren eignen sich nur für kleine Datenmengen
- DES ist bei gleicher Schlüssellänge etwa 1000 mal so schnell wie RSA

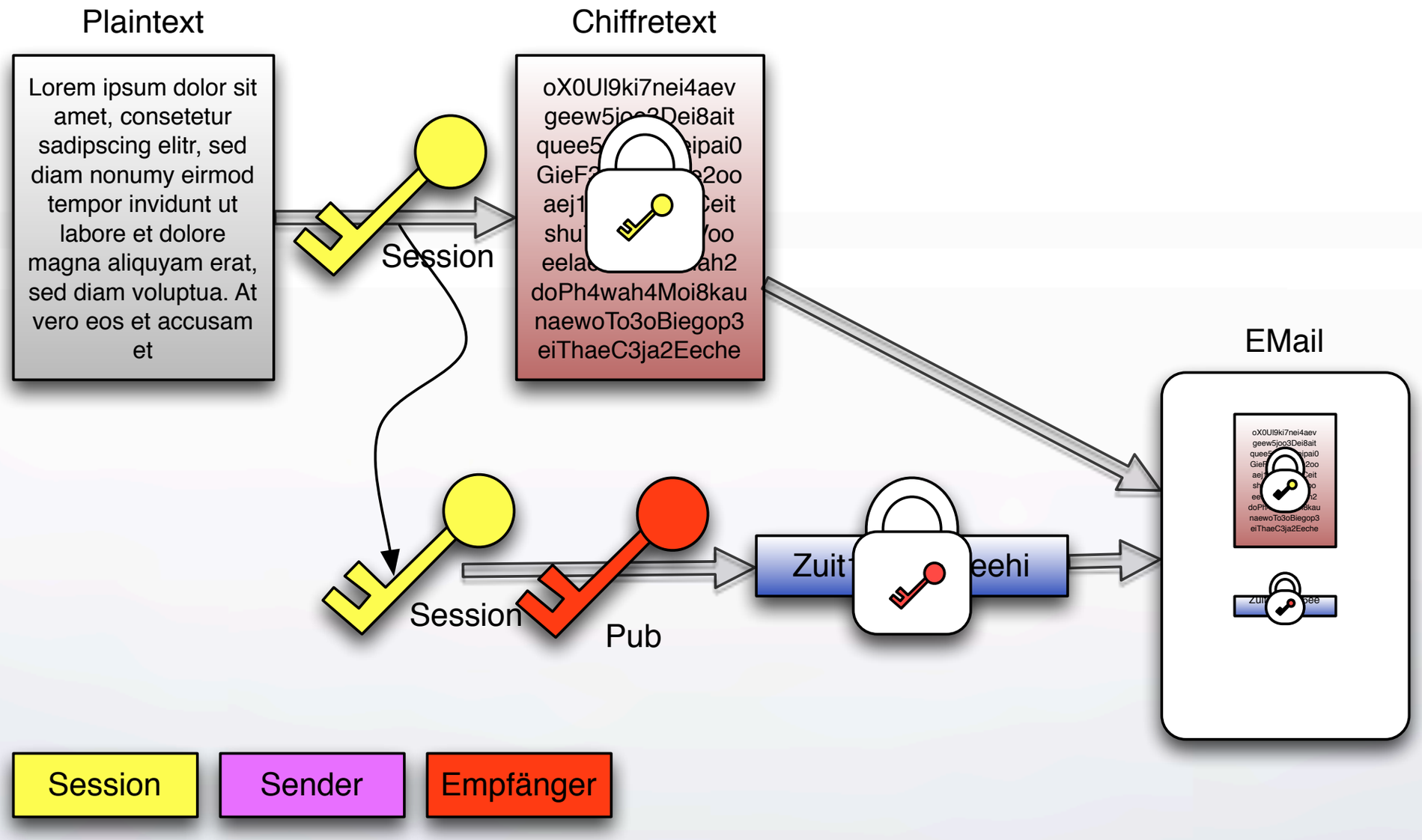
Nachteil asymmetrischer Kryptographie

- Jeder kann den öffentlichen Schlüssel benutzen
- Der Besitz des Schlüssels war immer auch der Nachweis, das die Nachricht vom richtigen Absender stammt
- Bei Public Key Verfahren kann man davon nicht ohne weiteres ausgehen!

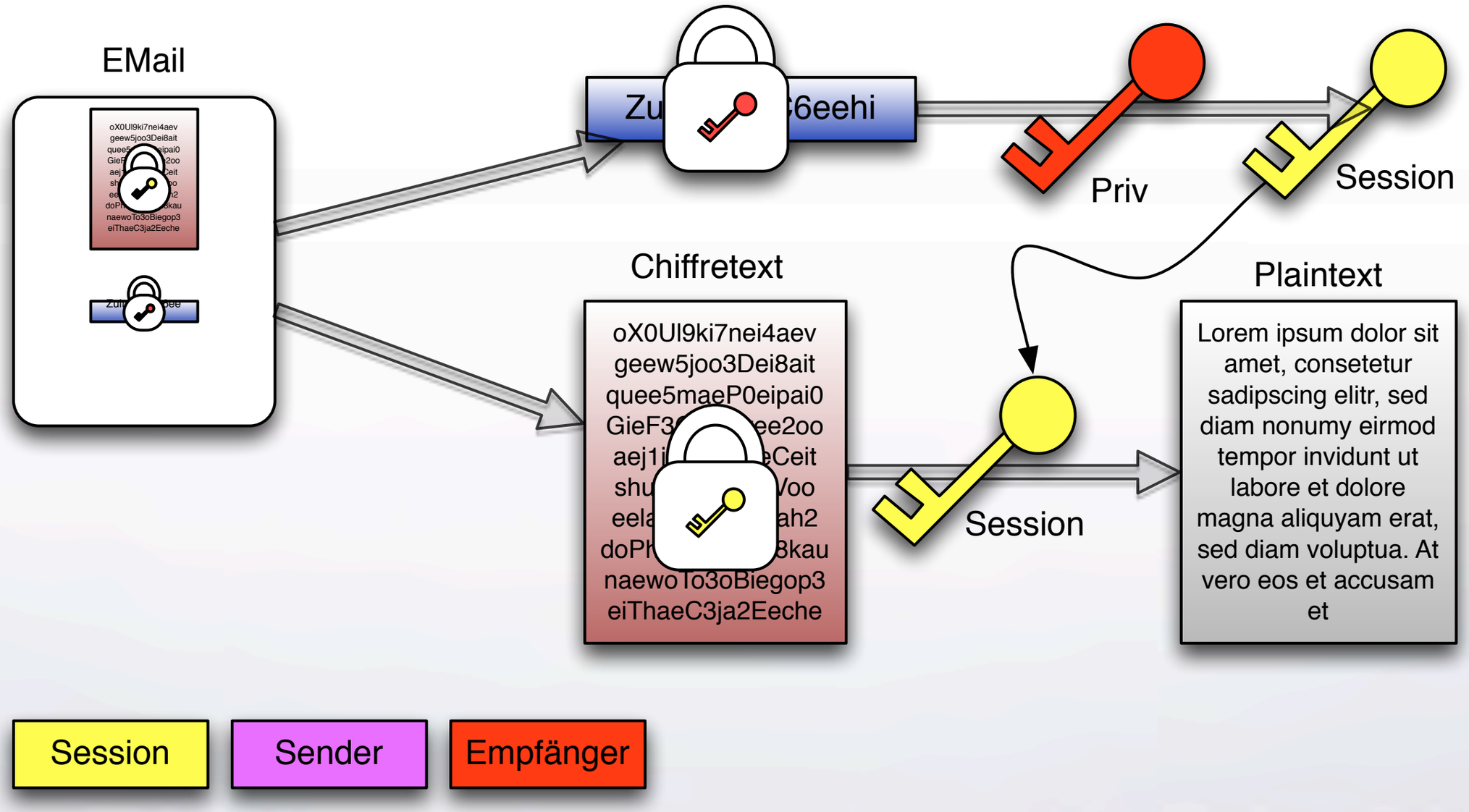
Hybridverfahren

- Es wird ein schnelles, symmetrisches Verfahren benutzt, um die Massendaten zu chiffrieren, z.B. AES oder Blowfisch
- Der symmetrische Schlüssel heisst Sessionkey (Erinnern: Enigma)
- Es wird ein asymmetrisches Verfahren benutzt, um den Sessionkey zu chiffrieren

Hybrid chiffrieren



Hybrid dechiffrieren



Ergebnis

- Das Verfahren schützt die Daten
- Es ist schnell
- Es kann spontan kommuniziert werden
- Jeder braucht nur einen öffentlichen Schlüssel zur Verfügung stellen um von beliebig vielen Personen Nachrichten empfangen zu können

Signatur

- Erinnerung:
  - Chiffrieren = Public Key
  - Dechiffrieren = Private Key
- bei RSA und ElGamal kann man auch mit dem Private Key verschlüsseln und mit dem Public Key entschlüsseln
- Die Nachricht kann **jeder dechiffrieren**

Signatur

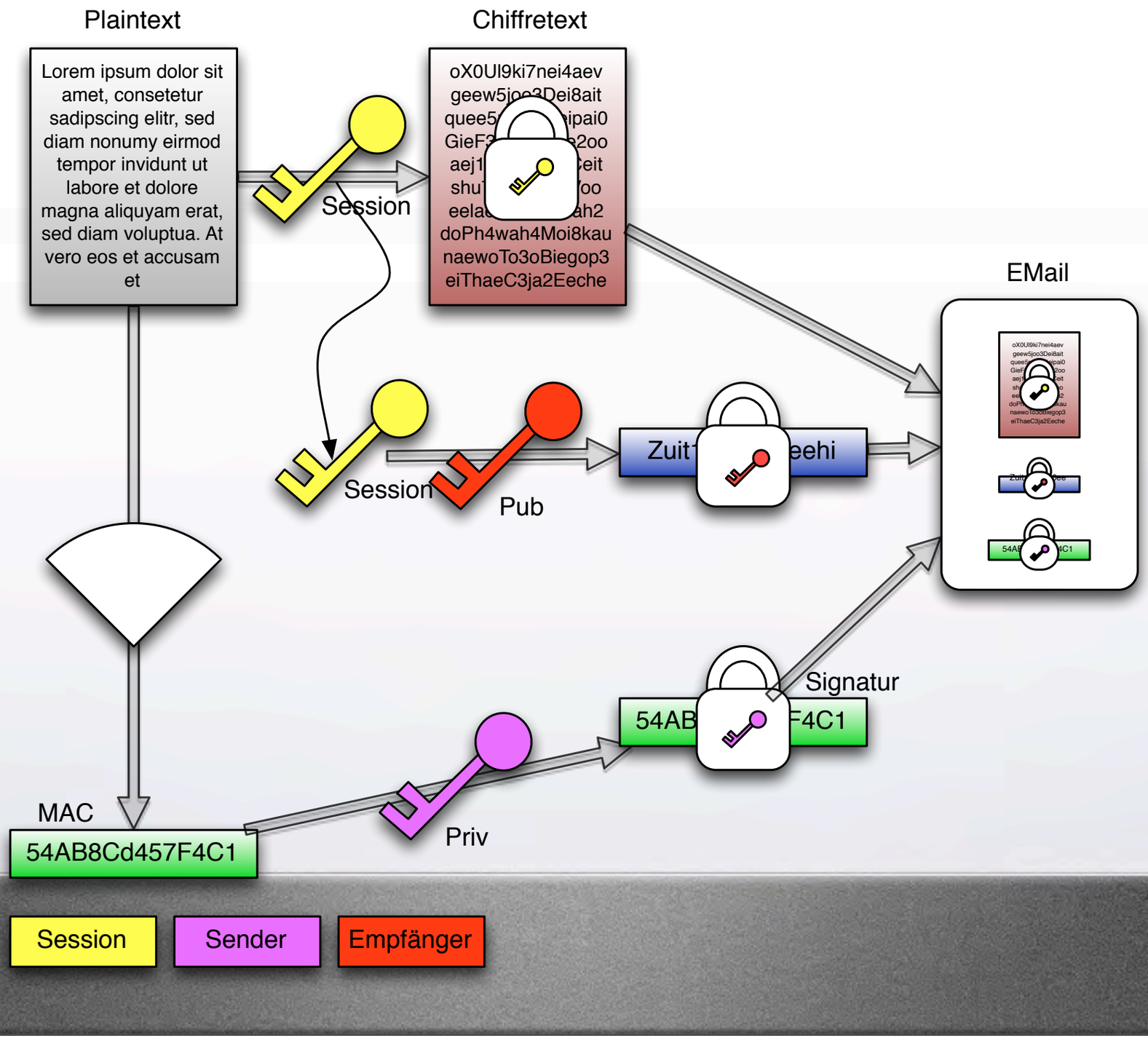
- Chiffretexte, die sich mit dem Public Key in eine sinnvolle Nachricht dechiffrieren lassen, können nur mit dem Private Key chiffriert worden sein
- Damit belegt eine sinnvolle, verschlüsselte Nachricht den Besitz des Private Key
- Der Besitz des Private Key ist die Identifikation des Absenders

Signature

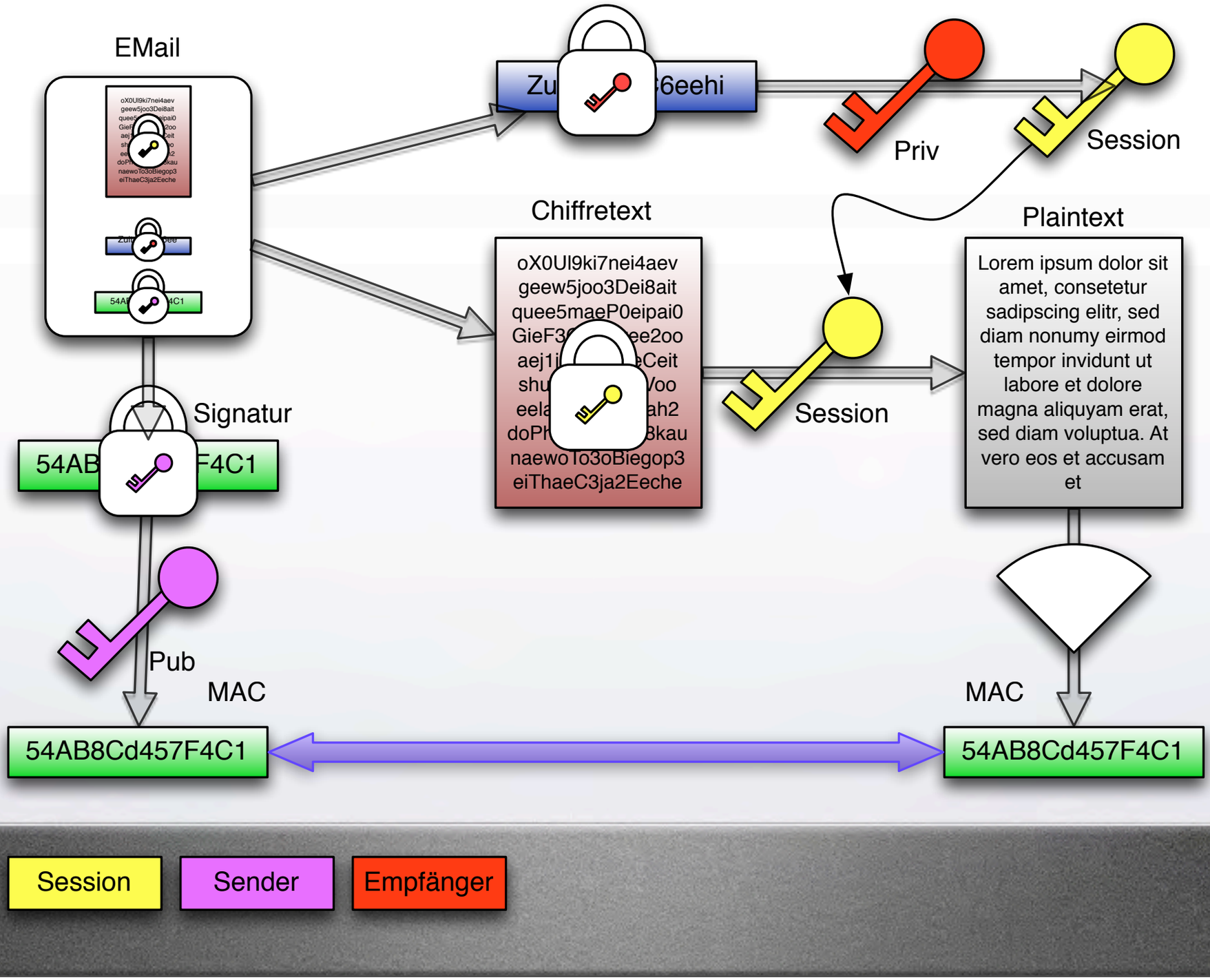
- Es wird eine Checksumme des Plaintext oder des Chiffretextes gebildet*)
- Die Checksumme wird mit dem Private Key verschlüsselt und mitgesendet
- Beim Empfänger muss die dort gebildete Checksumme mit der entschlüsselten Checksume übereinstimmen

*) MAC-then-Encrypt / Encrypt-then-MAC, MAC: Message Authentication Code

Hybrid signieren



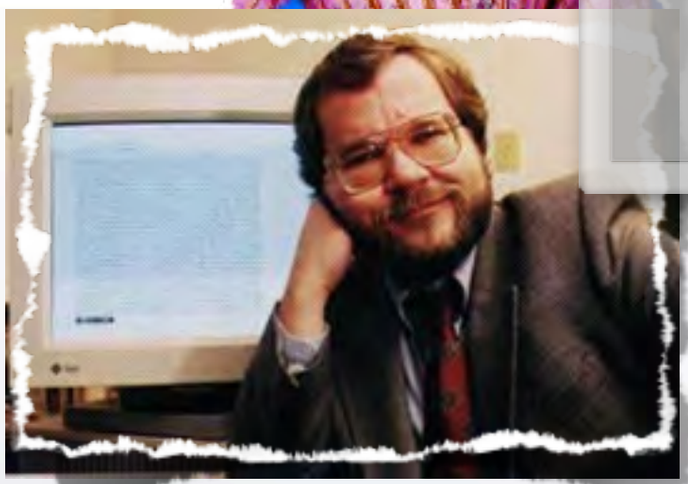
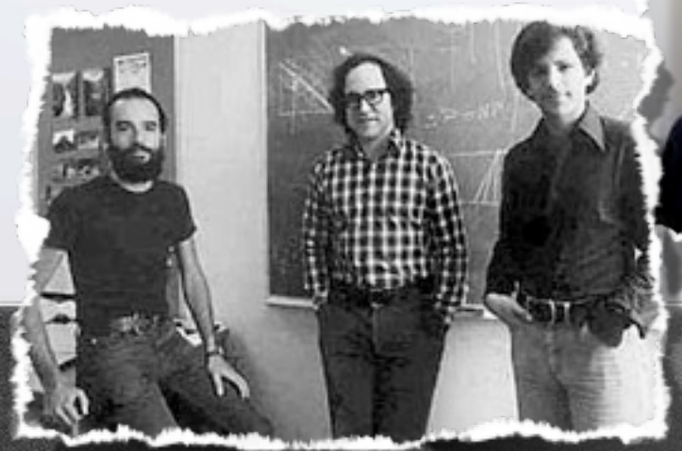
Hybrid prüfen






DE VIGENERE BOVRBONOIS. EN LAGE DE LXXIII ANS 1588. DEVS NOBIS HÆC OTIA FECIT





Vielen Dank

Literatur

- Bruce Schneier: Angewandte Kryptogaphie
- Klaus Schmeh: Kryptographie
- Rudolf Krippenhahn: Verschlüsselte Botschaften
- Simon Singh: Geheime Botschaften
- Russel et.al.: Die Hacker-Bibel
- Peter Higgins: Das kleine Buch der Zahlen
- www.wolframalpha.com
- de.wikipedia.org
- google.de