

SCHÜLERKRYPTO 2017

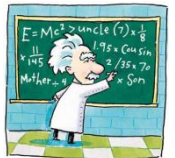
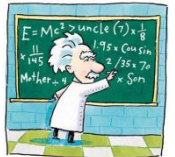
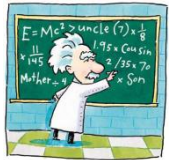
Kryptographie – Die Sprache der Geheimagenten

Prof. Arno Wacker
Angewandte Informationssicherheit
Universität Kassel

Programm



- **Einführung in die Kryptologie – Teil 1 (8:30-10:15)**
 - Einleitung
 - Einfache und verbesserte Verschlüsselung
- **Agenteneinsatz – Teil 1 (10:30-11:30)**
 - Grundausbildung zum Geheimagent
 - Aufträge 1-2
- **Einführung in die Kryptologie – Teil 2 (11:45-13:00)**
 - Maschinelle Verschlüsselung
 - Moderne Verschlüsselung (Teil 1/2)
- **Mittagessen**
- **Einführung in die Kryptologie – Teil 3 (13:30-14:30)**
 - Moderne Verschlüsselung (Teil 2/2)
- **Agenteneinsatz – Teil 2 (14:45-16:15)**
 - Agentenfortbildung
 - Aufträge 3-4





- **Codeknacker gegen Codemacher**
Klaus Schmeh
- **Geheime Botschaften**
Simon Singh
- **Entzifferte Geheimnisse**
Friedrich L. Bauer
- **Versteckte Botschaften**
Klaus Schmeh





- **Kryptografie**
Klaus Schmeh
- **Angewandte Kryptographie**
Bruce Schneier
- **Applied Cryptanalysis**
M. Stamp, R. Low
- **Modern Cryptanalysis**
Christopher Swenson



- **CrypTool 1.4.40** (Release)
<http://www.cryptool.org/cryptool1>
- **CrypTool 2.1** (Stable / Nightly builds)
<http://www.cryptool.org/cryptool2>
- **Schülerkrypto** Webseite
<http://www.cryptool.org/schuelerkrypto/>
- **MysteryTwister C3**
<http://www.mysterytwisterc3.org>



Übersicht

- Einleitung
- **Einfache** Verschlüsselung
- **Verbesserte** Verschlüsselung
- **Maschinelle** Verschlüsselung
- **Moderne** Verschlüsselung
- Zusammenfassung

Einleitung

Schülerkrypto 2017

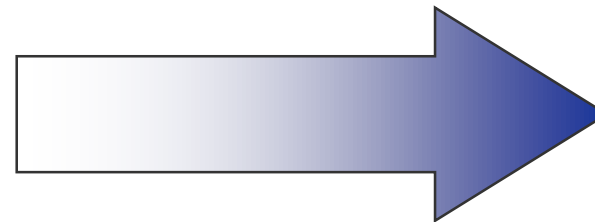
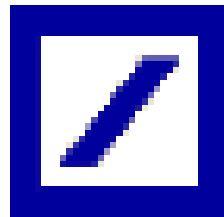
Kryptographie im Alltag

- Eurocheckkarten, Geldautomaten



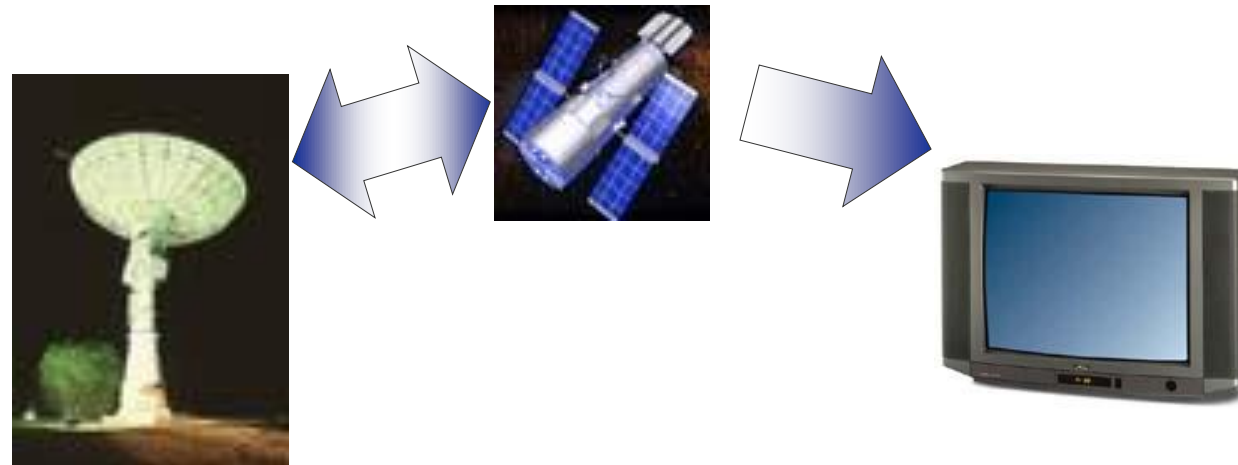
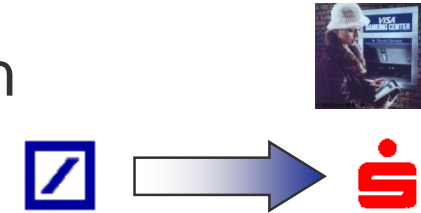
Kryptographie im Einsatz

- Eurocheckkarten, Geldautomaten
- Geldverkehr zwischen Banken



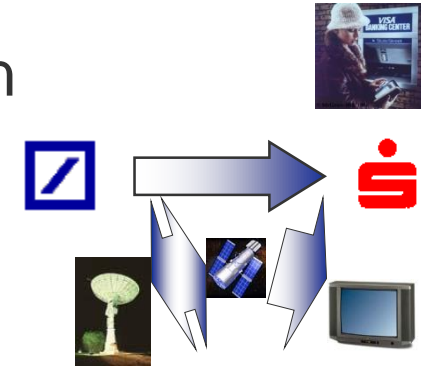
Kryptographie im Einsatz

- Eurocheckkarten, Geldautomaten
- Geldverkehr zwischen Banken
- Satellitenfernsehen, PayTV



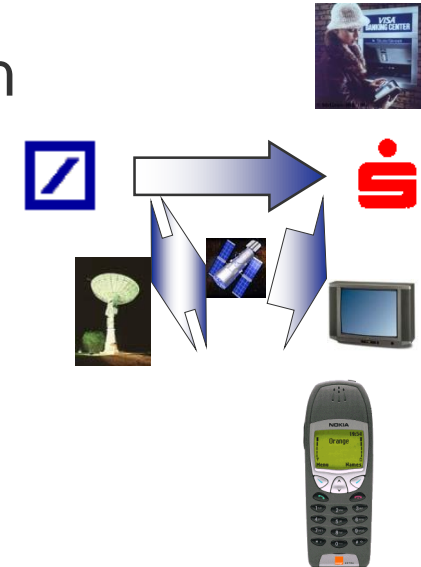
Kryptographie im Einsatz

- Euroscheckkarten, Geldautomaten
- Geldverkehr zwischen Banken
- Satellitenfernsehen, PayTV
- Telefon, Handy



Kryptographie im Einsatz

- Euroscheckkarten, Geldautomaten
- Geldverkehr zwischen Banken
- Satellitenfernsehen, PayTV
- Telefon, Handy
- Einkauf im Internet



Kryptographie im Einsatz

- Euroscheckkarten, Geldautomaten
- Geldverkehr zwischen Banken
- Satellitenfernsehen, PayTV
- Telefon, Handy
- Einkauf im Internet
- und vieles mehr ...



Emailsicherheit



Geheime Botschaften

Steganologie

- Steganographie
- Steganalyse

Kryptologie

- Kryptographie
- Kryptoanalyse

Steganologie

- **Steganographie**

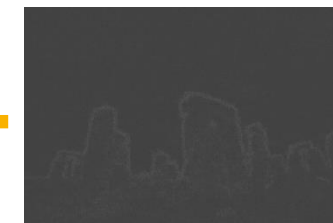
- Wissenschaft der verborgenen Übermittlung
- Altgriechisch
 - **Στεγανός, steganos** = *bedeckt, versteckt*
 - **γράφειν, gráphein** = *schreiben*
- Daten werden in unverdächtigem „Träger“ versteckt
- Träger: Bilder, Texte, Ton- und Videodokumente

- **Steganalyse**

- Analyse von steganographischen Verfahren
- Ziel
 - Extraktion der Daten
 - Existenznachweis der Daten

Steganographie – Beispiele

- **Antikes Griechenland: Tätowierung**
 - Nachrichten auf den Kopf tätowieren
 - Haare wachsen lassen/Helm aufsetzen
- **Unsichtbare Tinte**
 - Z. B. Essig oder Zitronensaft
 - Sichtbar durch Erwärmen
- **Steganos Security Suite**
 - Bettet Daten in Bilder- und Audio-Dateien ein
- **Unsichtbare Digitale Wasserzeichen**
 - Urheberschutz
 - Integritätsprüfung



- **Kryptographie**

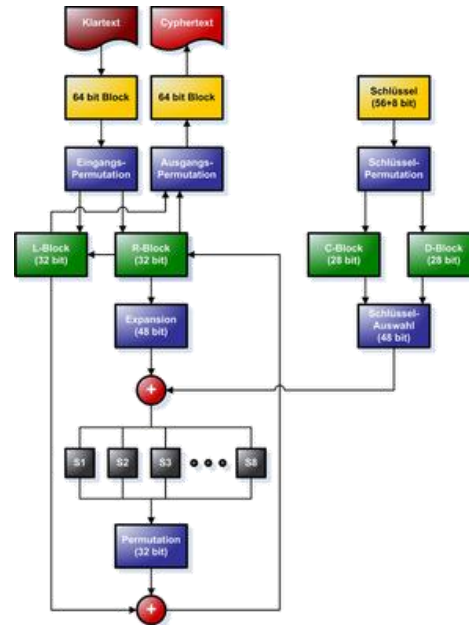
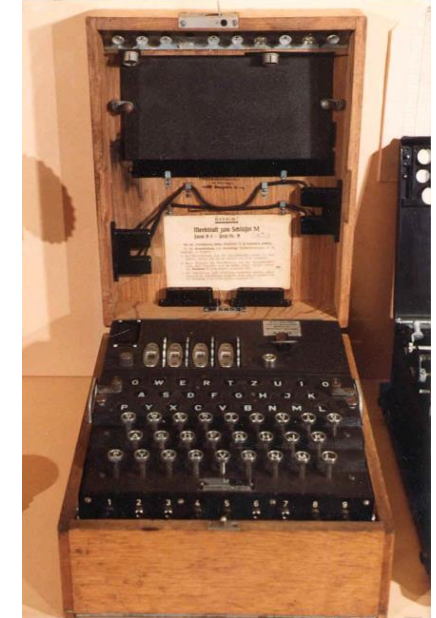
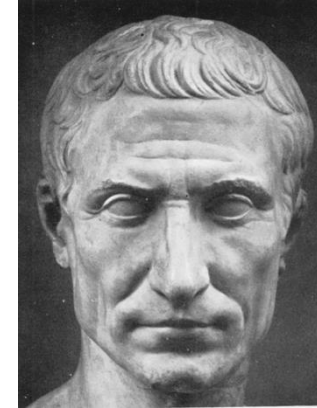
- Wissenschaft der Verschlüsselung von Informationen
- Griechisch
 - **κρυπτός, kryptós** = *verborgen, geheim*
 - **γράφειν, gráphein** = *schreiben*
- Nur autorisierte Person kann Nachricht lesen
- Ziele
 - Vertraulichkeit, Integrität, Authentizität, Verbindlichkeit

- **Kryptoanalyse**

- Gewinnung von Informationen aus verschlüsselten Texten
- Analyse von kryptographischen Verfahren
 - Brechen
 - Sicherheit nachweisen, quantifizieren

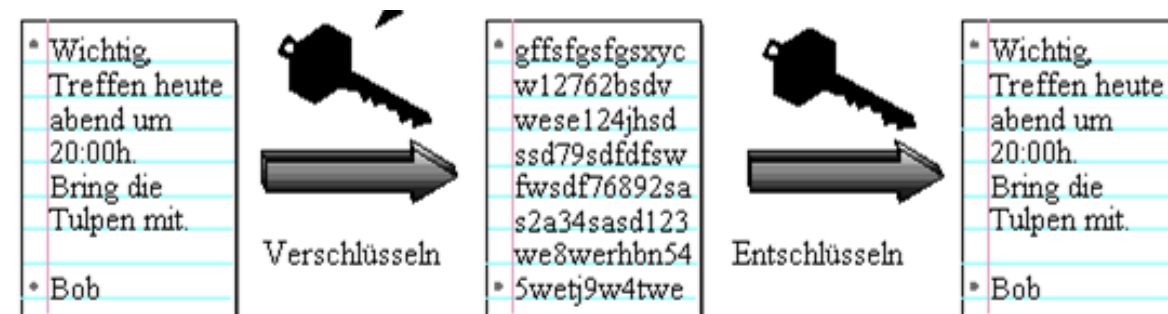
Kryptographie – Beispiele

- **Einfach**
 - Skytale
 - Caesar
- **Mechanisch**
 - Enigma
 - SIGABA
- **Modern**
 - DES, AES
 - RSA

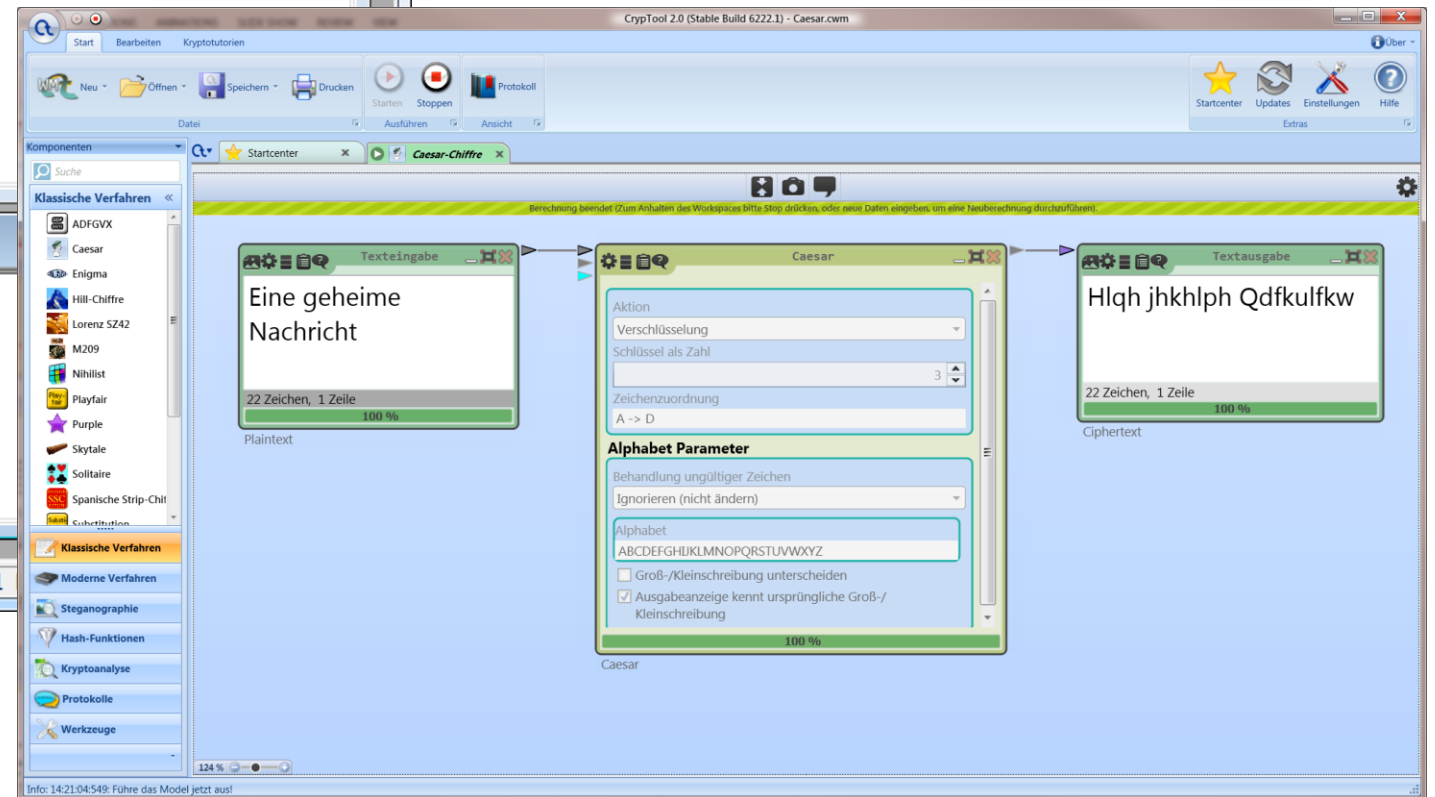
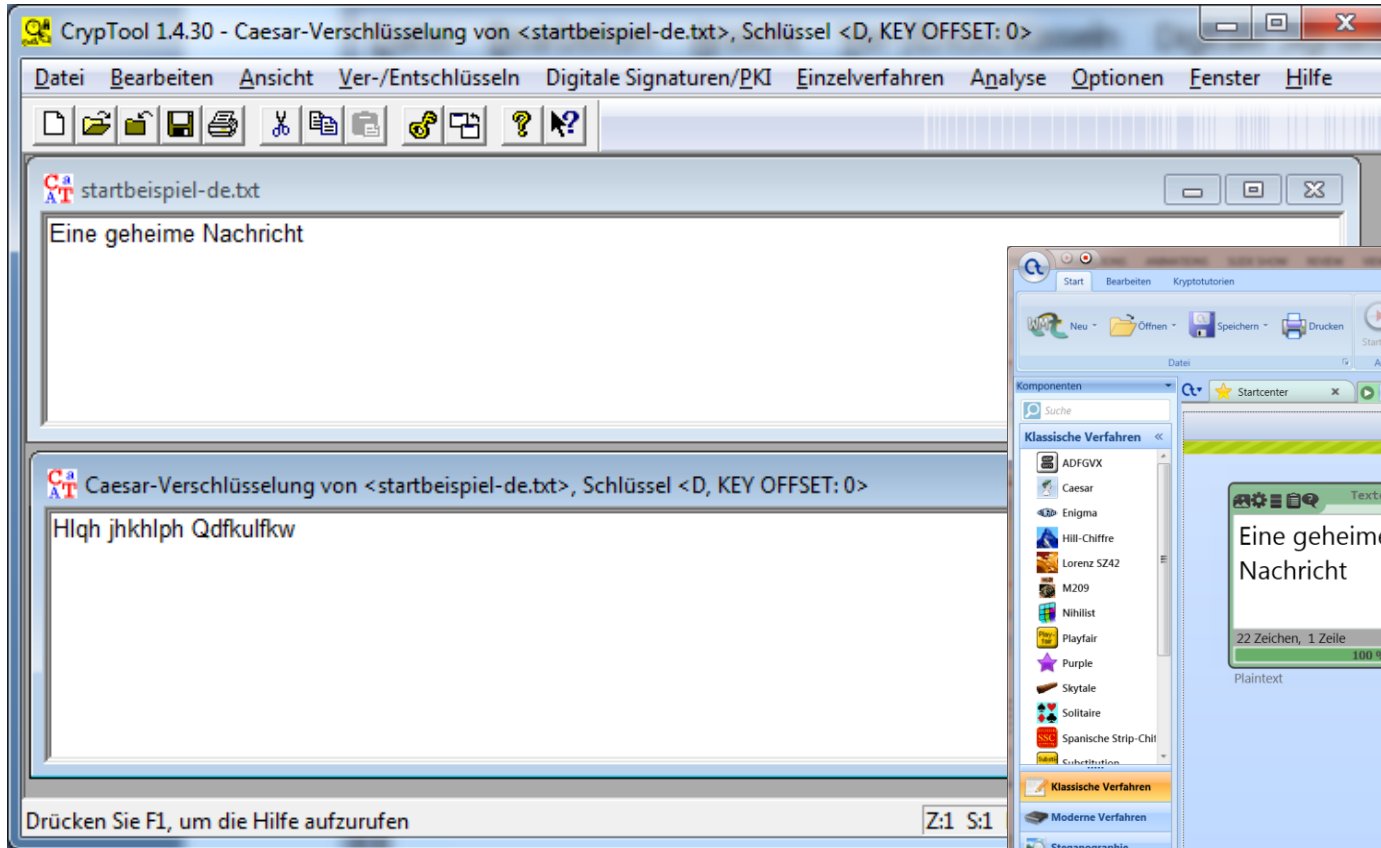


Verschlüsselung allgemein

- **Klartext** mit einem **Schlüssel** verschlüsseln
 - Wie = **Verschlüsselungsverfahren**
 - Schlüssel = Geheimnis
- **Verschlüsselte Nachricht (Geheimtext, Chiffre) übermitteln**
 - Funk
 - Internet
 - Zettel ...
- **Chiffre** mit **Schlüssel entschlüsseln**
 - Gleiches Verschlüsselungsverfahren wie beim Verschlüsseln
 - Meist auch gleicher Schlüssel



Demo: CrypTool 1.4.40 & CrypTool 2.1



Einfache Verschlüsselung

Schülerkrypto 2017

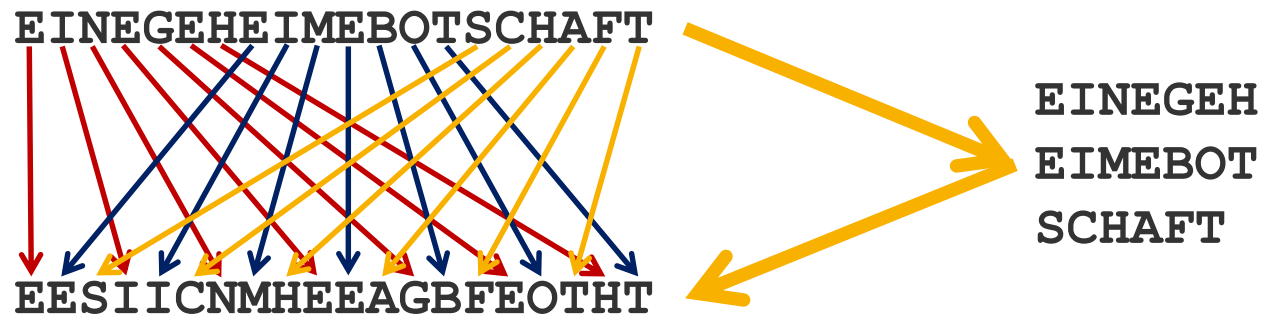
Die spartanische Skytale

- Ca. 400 v. Chr.
- Einsatz für militärische Zwecke
- Funktionsweise
 - Stab aus Holz mit bestimmtem Radius
 - Pergament (Papier, Leder) wird auf den Stab gewickelt
 - Nachricht wird nun auf das Pergament auf dem Stab geschrieben
 - Pergament wird abgewickelt und dem Kurier gegeben
 - Ohne den Stab machen die Buchstaben auf dem Streifen keinen Sinn
 - Der Empfänger wickelt den überbrachten Pergamentstreifen auf seinen Stab aus Holz



Die Skytale – Kryptographischer Hintergrund

- Klasse der **Transpositionsverfahren**
- Alle Buchstaben bleiben gleich
- Die Position der Buchstaben ändert sich

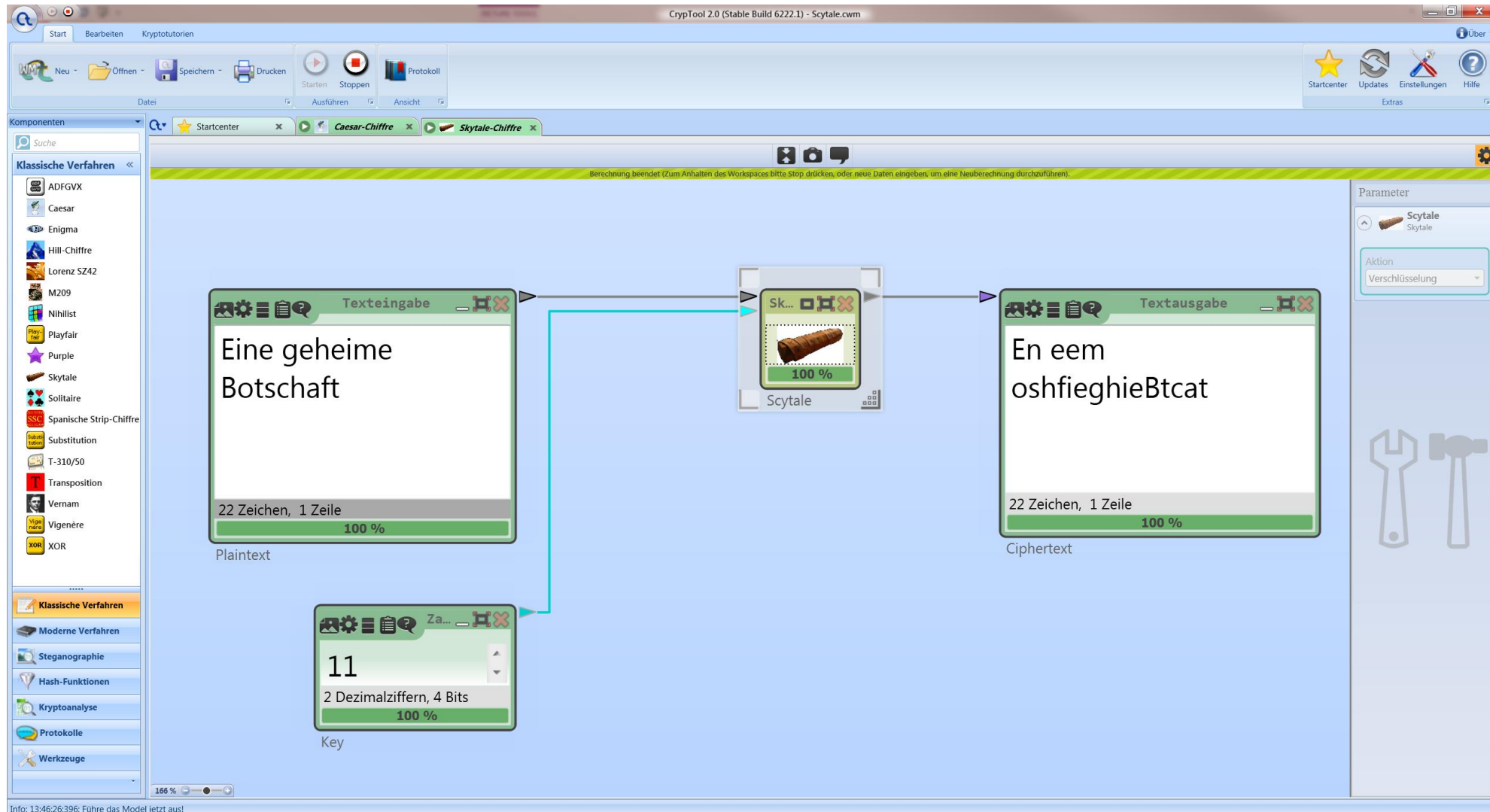


- Mathematiker sprechen von einer „**Permutation des Textes**“

Die Skytale – Kryptoanalyse

- **Länge des Textes: n Zeichen**
- **Maximaler Radius**
 - n Zeichen passen auf Umfang ($=2\pi R$)
 - $R_{\max} = n/2\pi$
- **Angriffsansatz:**
 - Alle Radien $x/2\pi$ mit $x=1\dots n$ probieren (also linear)
 - Textuell: Umbrüche nach $1\dots n$ Zeichen
- **Kann von Hand in Minuten gebrochen werden**
- **Unbrauchbar, sobald Verfahren bekannt ist**

Demo: Skytale mit CrypTool 2.1



Hands-On: Skytale – Rätsel



Empfangene Nachricht:

U R | H H | Z Z H > O | A O R | H Z D |

2 Spalten:

GR
_E
E_
NN
FV
O_
AO
R_
HN
D_
_

3 Spalten:

GR_
EE_
NNF
VO_
AOR
_HN
D_
_

4 Spalten:

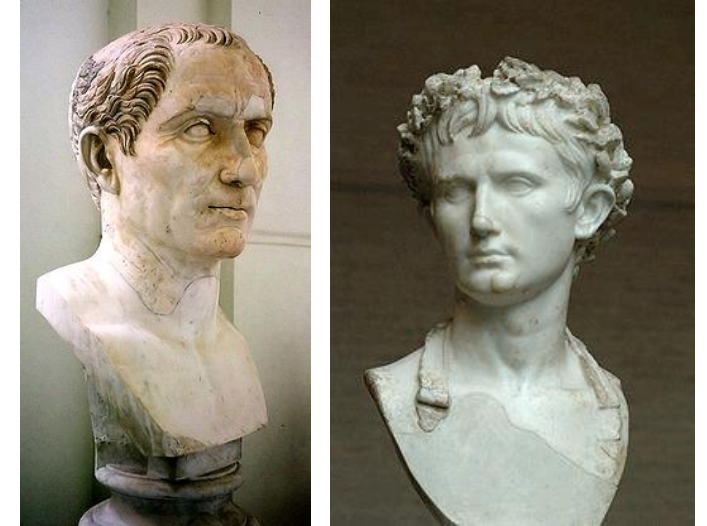
GR_E
E_NN
FVO_
AOR_
HND_
_

Entschlüsselte Nachricht:

GEFAHR_VON_NORDEN

Die Caesar-Verschlüsselung

- **Eingesetzt von**
 - Julius Caesar (100 – 44 v.Chr.)
 - Augustus Caesar (63 v.Chr. – 14 n.Chr.)
- **Militärische Zwecke**
- **Funktionsweise**
 - Ersetzung der Buchstaben im Alphabet
 - z.B. anstelle von A schreibt man D, anstelle von B wird E geschrieben usw.
 - Empfänger kennt diese Ersetzung und kann sie rückgängig machen
 - Abgefangene Nachrichten machen keinen Sinn



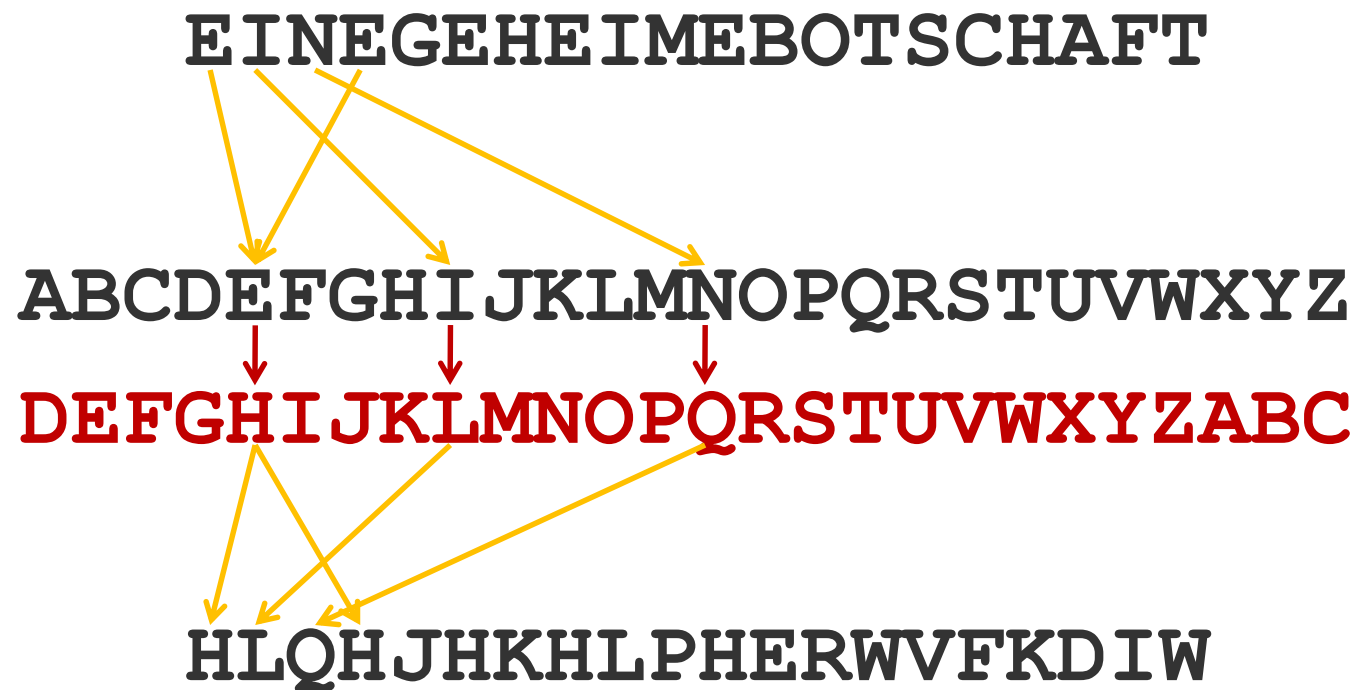
Caesar – Kryptographischer Hintergrund

- Klasse der **Substitutionsverfahren**
- Ein Zeichen wird durch ein anderes ausgetauscht
- Position der Zeichen ändert sich nicht
- Mathematiker sprechen von einer „**Permutation des Alphabets**“
 - Bei Caesar: Verschiebung (um 3)

ABCDEFGHIJKLMNOPQRSTUVWXYZ
...
DEFGHIJKLMNOPQRSTUVWXYZABC

Caesar – Beispiel

- Nachricht: „EINEGEHEIMEBOTSCHAFT“
- Schlüssel: 3 (Verschiebung im Alphabet)



Caesar – Chiffrierscheiben

ABCDEFGHIJKLMNOPQRSTUVWXYZ

DEFGHIJKLMNOPQRSTUVWXYZABC

Verschlüsseln
↓
Entschlüsseln
↑



Giovanni Batista Porta (1563)
De Furtivis Literarum Notis

Caesar – Kryptoanalyse (1/3)

- **Einfachster Angriff**

- Nur 26 verschiedene Schlüssel (Verschiebungen)
- Alle probieren (brute force)

- **Statistischer Angriff**

- Buchstaben einer Sprache haben bestimmte Häufigkeiten
- Häufigkeiten der Buchstaben im Chiffretext berechnen
- Buchstabenhäufigkeiten der Sprache mit der der Chiffre vergleichen

Caesar – Kryptoanalyse (2/3)

Statistische Häufigkeitsanalyse

- **Häufigkeitsverteilung von Buchstaben einer Sprache**
- **Kann für jede Sprache berechnet werden**
 - Z. B. ein Buch
 - Allg. „Textkorpus“

Häufigkeitsverteilung
der deutschen Sprache

Letter	%	Letter	%
A	6.51	N	9.78
B	1.89	O	2.51
C	3.06	P	0.79
D	5.08	Q	0.02
E	17.40	R	7.00
F	1.66	S	7.27
G	3.01	T	6.15
H	4.76	U	4.35
I	7.55	V	0.67
J	0.27	W	1.89
K	1.21	X	0.03
L	3.44	Y	0.04
M	2.53	Z	1.13

Caesar – Kryptoanalyse (3/3)

Statistische Häufigkeitsanalyse - Beispiel

- **Empfangene Nachricht:**

„MQOOGP UKG BWT UEJWGNGTMTARVQ!“

- **Bestimmung der Buchstabenhäufigkeiten**

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
1	1	0	0	1	0	4	0	0	1	1	0	2	1	2	1	2	1	0	3	2	1	2	0	0	0

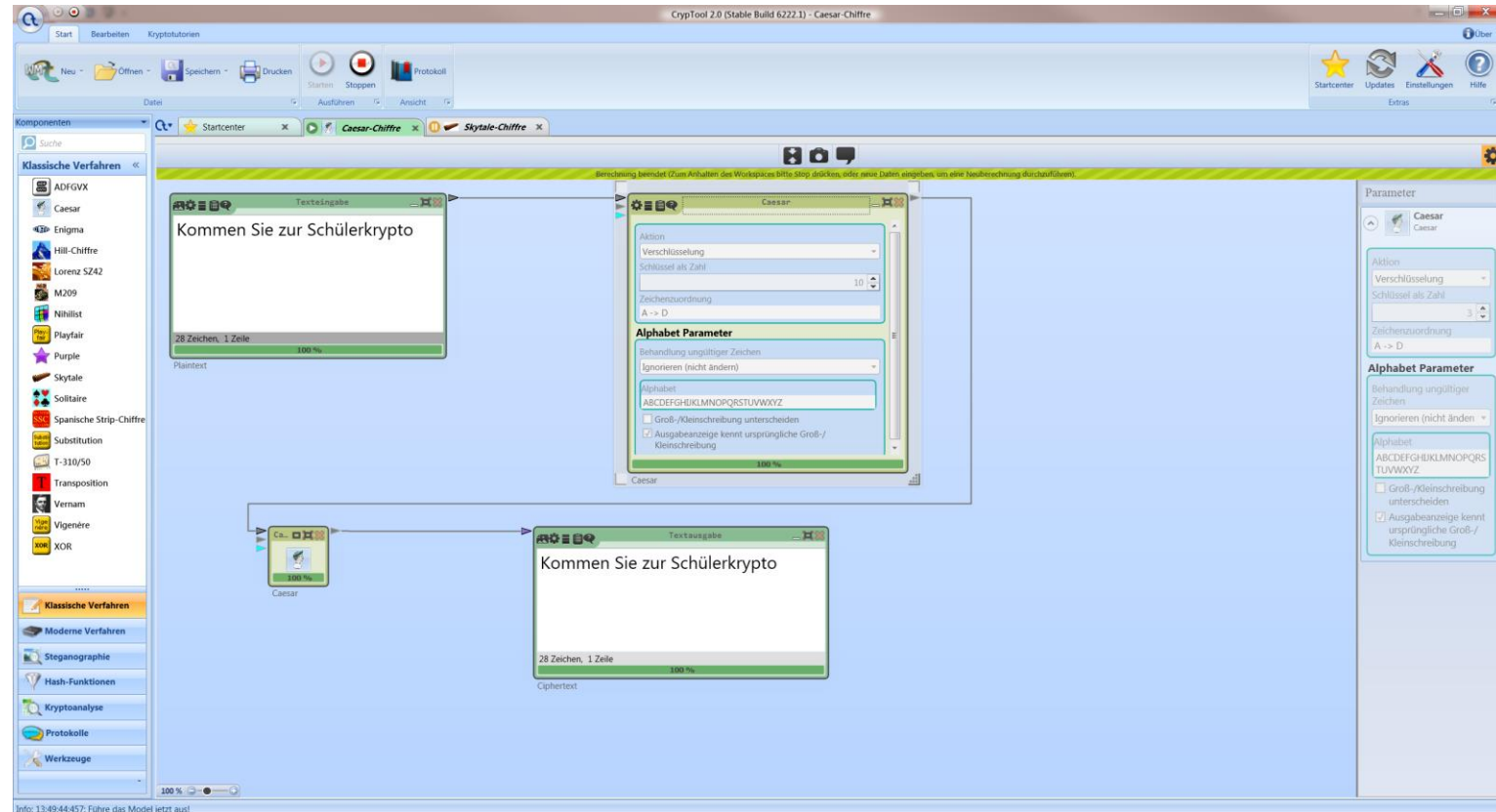
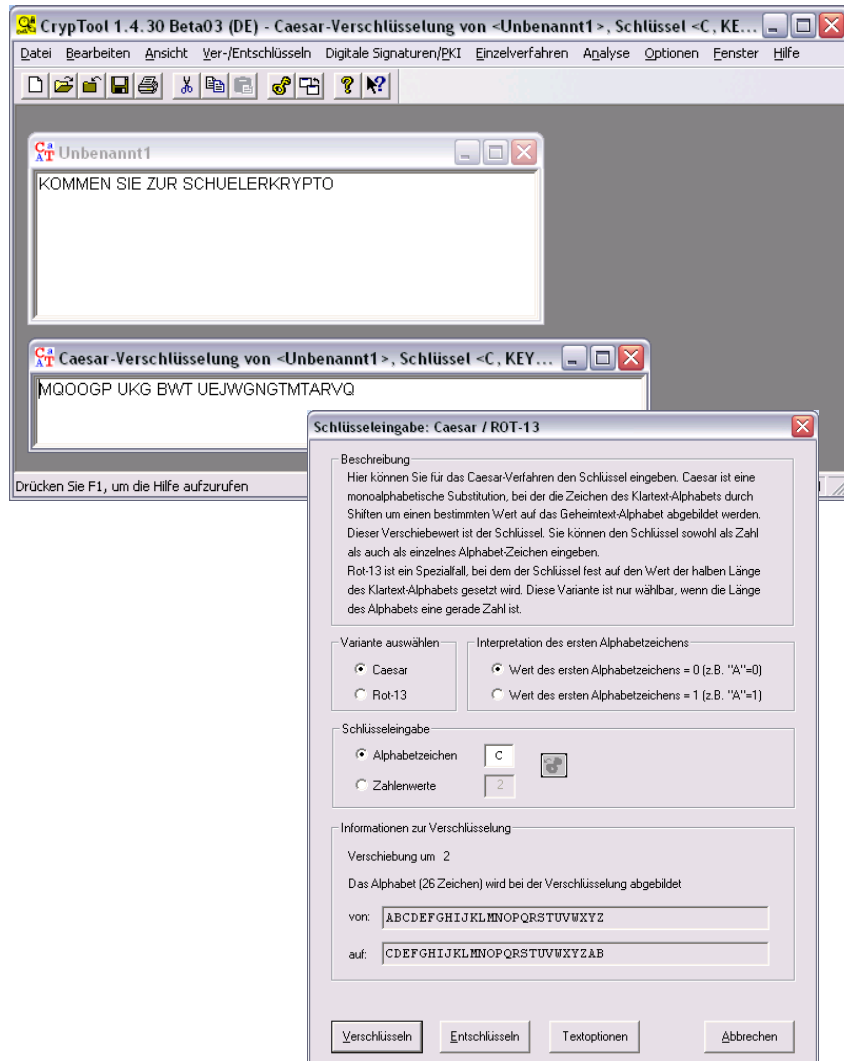
- **Kryptoanalyse**

- Annahme: $G \leftrightarrow E$
- Damit handelt es sich um eine Verschiebung von 2

- **Ergebnis:**

„KOMMEN SIE ZUR SCHUELERKRYPTO!“

Demo: Caesar mit CrypTool 1.4.40 und 2.1





GLH JDOOLHU NRPPHQ!

Caesar – Rätsel: Kryptoanalyse (Variante 1)

- Bestimmung der Buchstabenhäufigkeiten

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
0	0	0	1	0	0	1	3	0	1	0	2	0	1	2	2	1	1	0	0	1	0	0	0	0	0

- Kryptoanalyse von „GLH JDOOLHU NRPPHQ“

- Annahme: H = E
- Also Verschiebung um 3:

DEFGHIJKLMNOPQRSTUVWXYZABC
ABCDEFGHIJKLMNOPQRSTUVWXYZ

- Ergebnis:
„DIE GALLIER KOMMEN!“

Caesar – Rätsel: Kryptoanalyse (Variante 2)

C H D	F Z K K H D Q	J N L L D M
D I E	G A L L I E R	K O M M E N !
E J F	H B M M J F S	L P N N F O
F K G	I C N N K G T	M Q O O G P
GLH	JDOOLHU	NRPPHQ!
H M I	K E P P M I V	O S Q Q I R
I N J	L F Q Q N J W	P T R R J S
J O K	M G R R O K X	Q U S S K T
K P L	N H S S P L Y	R V T T L u

„Running down the alphabet...”

Verbesserte Verschlüsselung

Schülerkrypto 2017

Einfache Spaltentransposition

- **Klartext wird in Zeilen fixer Länge geschrieben**
 - Alle Zeilen genau gefüllt → regelmäßige Spaltentransposition
 - Letzte Zeile unvollständig → unregelmäßige Spaltentransposition
- **Spaltenreihenfolge wird verändert (permutiert)**
 - Verwendung eines Schlüsselwortes
 - Lange Schlüsselwörter erhöhen die Sicherheit
- **Spaltenweise auslesen**

Einfache Spaltentransposition – Beispiel (1)

- Klartext: „WE ARE DISCOVERED. FLEE AT ONCE.“
- Schlüssel: ZEBRAS (→ 6 3 2 4 1 5)

ZEBRAS	EVLN		
632415	ACDT		
WEARED	ESEA	EVLNA	CDTES
ISCOVE	ROFO	EAROF	ODEEC
REDFLE	DEEC	WIREE	
EATONC	WIRE		
E	E		

Einfache Spaltentransposition – Beispiel (2)

- Chiffretext: EVLNA CDTES EAROF ODEEC WIREE
- Schlüssel: ZEBRAS, also 6 3 2 4 1 5
- Vorbereitung (unregelmäßig)
 - Insgesamt 25 Zeichen in 6 Spalten
 - 5 Spalten mit 4 Zeichen
 - 1 Spalte mit 5 Zeichen

Z	E	B	R	A	S
6	3	2	4	1	5
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	F	L	E
E	A	T	O	N	C
E					

Spaltentransposition – Kryptoanalyse (1/3)

- **Länge des Schlüssels**
 - Extremfall: Länge des Schlüssel = Länge des Textes
 - In diesem Extremfall $n(n-1)(n-2) \dots 2 = n!$ Möglichkeiten (Permutationen)
 - Längen > 20 sind schwierig (ohne Klartextkenntnis)
- **Kryptoanalyse besteht aus zwei Schritten**
 - 1. Schlüssellänge bestimmen
 - 2. Spaltenpermutation rückgängig machen

Spaltentransposition – Kryptoanalyse (2/3)

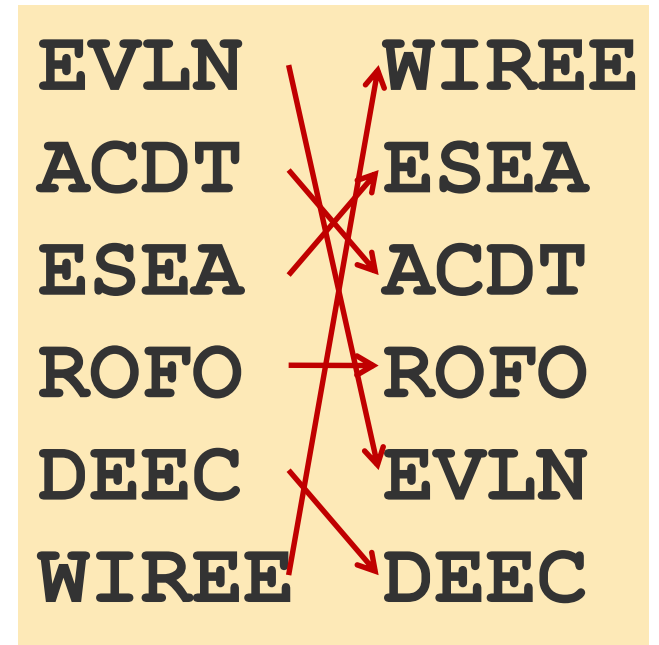
Bestimmung der Schlüssellänge

- **Schlüssellänge bestimmen**
 - Ohne Klartextkenntnis
 - Alle probieren von 1 ... n (n= Länge des Textes)
 - Vokale zählen und mit Statistik der Sprache vergleichen
 - Für jede Länge Anagramme suchen
 - (Teil-)Kenntnis des Klartextes
 - Suchen nach Textmuster für alle Längen
 - Wenn Muster gefunden, dann Schlüssellänge möglich
 - Wenn Muster nicht vorhanden, dann Schlüssellänge falsch
- **Beispiel**
 - Annahme wir erwarten **DISCOVERED** im verschlüsselten Text.
 - Chiffretext: EVLNA CDTES EAROF ODEEC WIREE
 - Suche nach
 - Schlüssellänge 2: **DSOEE** und **ICVRD** → nicht vorhanden
 - Schlüssellänge 3: **DCED** , **IOR** und **SVE** → nicht vorhanden
 - Schlüssellänge 4: **DOE**, **IVD**, **SED**, **CR** → nicht vorhanden
 - Schlüssellänge 5: **DV**, **IE**, **SR**, **CE**, **OD** → nicht vorhanden
 - Schlüssellänge 6: **DE**, **IR**, **SE**, **CD** → wahrscheinliche Länge!

Spaltentransposition – Kryptoanalyse (3/3)

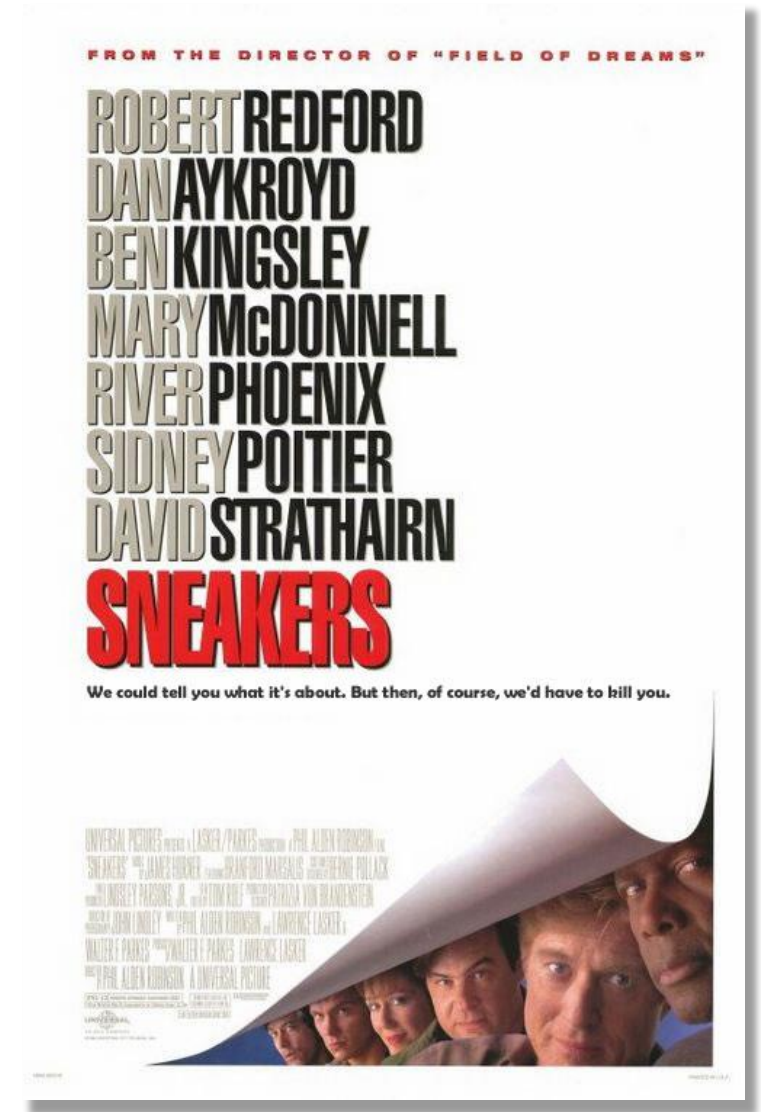
Bestimmung der Spaltenpermutation

- **Kenntnis der Schlüssellänge führt zu**
 - 5 Spalten mit 4 Zeichen
 - 1 Spalte mit 5 Zeichen
- **Text nach jeweils 4 Zeichen trennen (letzte 5)**
- **Umordnen, um DISCOVERED zu bilden (Anagramm suchen)**



Anagramme

- A TURNIP CURES ELVIS
- A FEW ASTRAL CLERKS
- REPEL NEWARK
- BLOND RHINO SPANIEL
- FORT RED BORDER
- SETEC ASTRONOMY



Demo: Spaltentransp. mit CrypTool 1.4.40 / 2.1

The screenshot displays the CrypTool 1.4.30 interface with the title "Permutations-/Transpositions-Verschlüsselung von <Unbenannt1>, S...". The main window shows a workflow for transposition encryption. The input text is "WEAREDISCOVEREDFLEEATONCE". The intermediate text is "EVLNACDTESEAROFODEECWIREE". The final output is "w cle tred dnidt,h n nefwusuekinkWrntena".

The workflow diagram shows the following steps:

- Input: WEAREDISCOVEREDFLEEATONCE
- Transposition (Spaltenweise): EVLNACDTESEAROFODEECWIREE
- Transposition (Zeilenweise): w cle tred dnidt,h n nefwusuekinkWrntena
- Output: w cle tred dnidt,h n nefwusuekinkWrntena

The settings dialog "Schlüssel eingabe: Permutation / Transposition" is open, showing the following options:

- 1. Permutation (einfache Spaltentransposition)
 - Schlüssel (Buchstaben oder durch Kommas separierte Zahlen): 6,3,2,4,1,5
 - Permutation, wenn der Schlüssel als Buchstabenfolge eingegeben wurde: [Empty]
 - Zeilenweise: ☒ einlesen, ☐ permutieren, ☐ auslesen
 - Spaltenweise: ☐ einlesen, ☒ permutieren, ☒ auslesen
- 2. Permutation (doppelte Spaltentransposition)
 - Schlüssel (Buchstaben oder durch Kommas separierte Zahlen): [Empty]
 - Permutation, wenn der Schlüssel als Buchstabenfolge eingegeben wurde: (1)
 - Zeilenweise: ☒ einlesen, ☐ permutieren, ☐ auslesen
 - Spaltenweise: ☐ einlesen, ☒ permutieren, ☒ auslesen
- Optionen
 - ☐ Jeweils die inverse Permutation anwenden
 - ☐ Zwischendialog mit der inversen Permutation anzeigen
 - Betrachte Eingabedokument als: ☐ Binäre Daten, ☒ Text

Buttons: Verschlüsseln, Entschlüsseln, Abbrechen

Monoalphabetische Substitution

- **Definition:**
Ein Klartextbuchstabe wird **immer gleich ersetzt**
- **Beispiele:**
 - Caesar (Verschiebung)
 - Allgemeine Permutation des Alphabets, z. B.
 - Schlüssel = {Wort: GEHEIMSCHRIFT; Startposition: E}
 - Doppelte Buchstaben entfernen: GEHIMSCRFT
 - Permutation des Alphabets wie folgt erstellen:

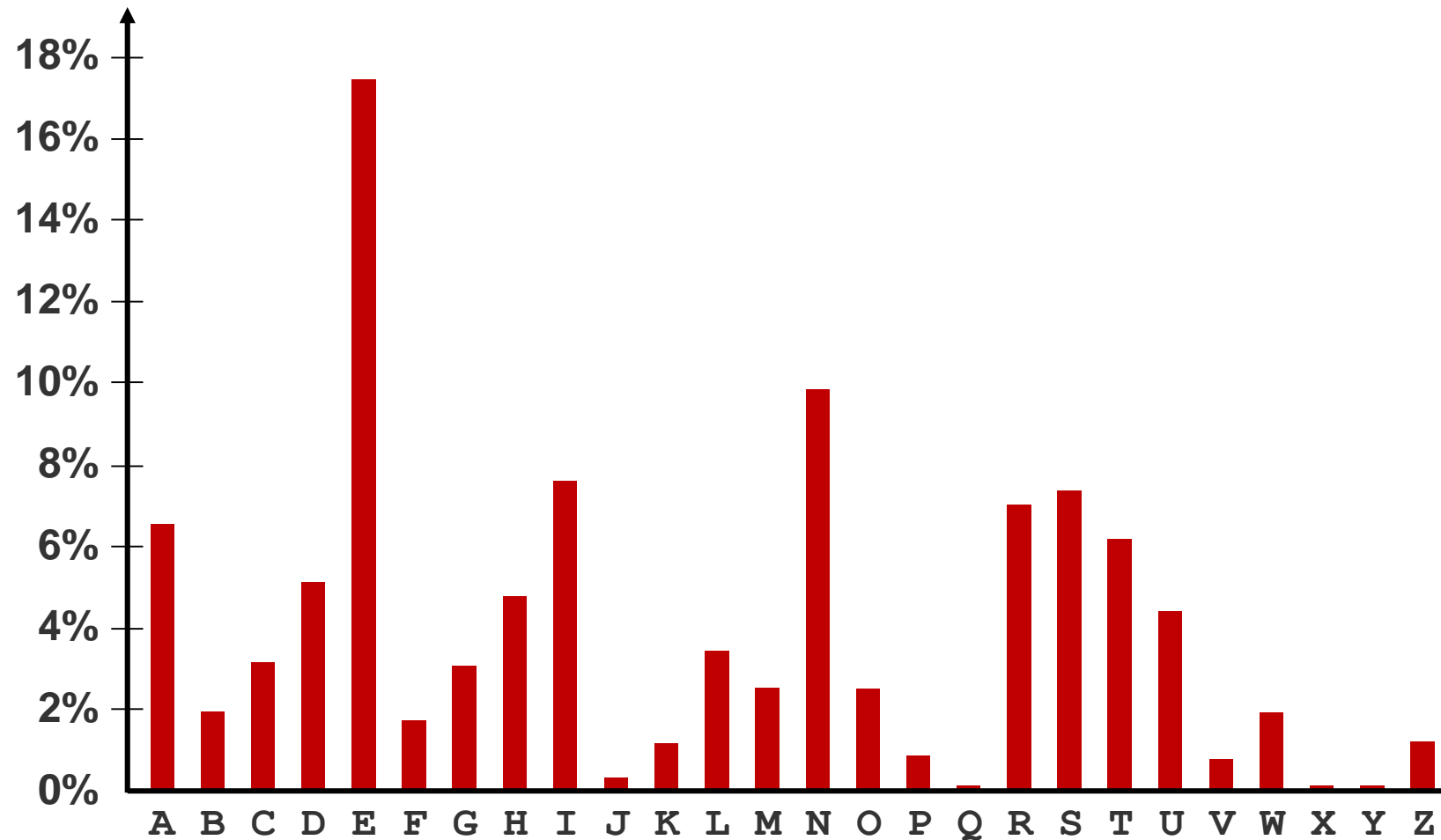
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
W	X	Y	Z	G	E	H	I	M	S	C	R	F	T	A	B	D	J	K	L	N	O	P	Q	U	V

Substitution – Kryptoanalyse (1/2)

- **Es gibt $26 \cdot 25 \cdot 24 \cdot \dots \cdot 2 \cdot 1 = 26! \sim 2^{88}$ Permutationen**
 - Zu viele, um alle zu probieren (selbst heute!)
- **Annahme: Wir kennen *nur* den verschlüsselten Text (schwierigster Fall für den Kryptoanalytiker)**
- **Idee:**
 - Die 10 häufigsten Buchstaben im Deutschen bilden ca. 75% des gesamten Textes
 - Berechne/analysiere die 10 häufigsten Buchstaben
 - Rate den Rest
- **Beispiel**
 - Die –ehn haeu-i-sten –u-hsta-en i- deuthen –i-den drei –ierte- des –esa-tte-tes

Substitution – Kryptoanalyse (2/2)

Häufigkeitsverteilung der Buchstaben



Substitution – Beispiel

$\nabla \setminus \emptyset \Delta \swarrow \quad \nabla :: \Delta$

$\vdash : \square \setminus \square \quad \Delta \square \setminus$

$\pm : \square \square : \setminus \times$

$\ominus \setminus + \times \square \odot \quad \cdot \emptyset :$

$\vdash + \emptyset : : \square$

Mit freundlicher Genehmigung
von Klaus Schmeh

Substitution – Beispiel

▽.\:∅Δ↙ ▽ ■ ■ Δ

† ■ □.\:□ Δ□.\:

±.\:□□.\:.\:×

⊖.\:†×□⊙ · ∅ ■

†+∅.\: ■ □

Mit freundlicher Genehmigung
von Klaus Schme

Substitution – Beispiel

▽\∅Δ↙ ▽ ■ ■ Δ

E E

† ■ □\□ Δ□\°

E

±\□□\..X

Θ\+X□⊙ ·∅ ■

E

†+∅\ ■ □

E

Mit freundlicher Genehmigung
von Klaus Schme

Substitution – Beispiel

▽\∅Δ↙ ▽: :Δ
F O R T Y F E E T

†:□\□ Δ□\∅
B E L O W T W O

±:□□:..\X
M I L L I O N

⊖\+X□⊙ ∅:
P O U N D S A R E

†+∅: :□
B U R I E D

Mit freundlicher Genehmigung
von Klaus Schme

Demo: Substitution mit CrypTool 1.4

The screenshot displays the CrypTool 1.4.30 interface with several windows open:

- Main Window:** Shows the file "Substitution-Klartext.txt" with a text area containing German text about substitution encryption. The text is partially obscured by other windows.
- Methodenauswahl zur automatische Substitutionsanalyse:** A dialog box asking to choose between two algorithms:
 - ☒ Verfahren 1 basierend auf der Häufigkeitsanalyse der Digramme im Text. (Selected)
 - ☐ Verfahren 2 basierend auf der Erkennung der Sprache.
- Substitutionsanalyse: Manuelle Nachbearbeitung:** A dialog box for manual analysis, showing a grid of letters and their corresponding ciphertext letters. The grid is as follows:

a: K	b: G	c: H	d: I	e: B	f: X	g: A
h: C	i: D	j: V	k: P	l: L	m: E	n: N
o: O	p: M	q: Q	r: R	s: S	t: T	u: U
v: J	w: W	x: F	y: Y	z: Z		
- Schlüsseleingabe: Monoalphabetische Substitution / Atbash:** A dialog box for key input. It has three options for key generation, with "Schlüsseleingabe: Restliche Zeichen aufsteigend füllen" selected. The key input field contains "GEHEIM". Below it, there are fields for "Das Alphabet (26 Zeichen) wird abgebildet" with "von: ABCDEFGHIJKLMNOPQRSTUVWXYZ" and "auf: GEHEIMABCDEFGHIJKLNOPQRSTUVWXYZ".

Maschinelle Verschlüsselung

Schülerkrypto 2017

Die Enigma



- Rotor-Schlüsselmaschine aus dem zweiten Weltkrieg
- Verwendet von:
 - deutschem Militär
 - Polizei
 - Geheimdiensten
 - ...
- „Enigma“ (αίνιγμα) [Griech.] = Rätsel



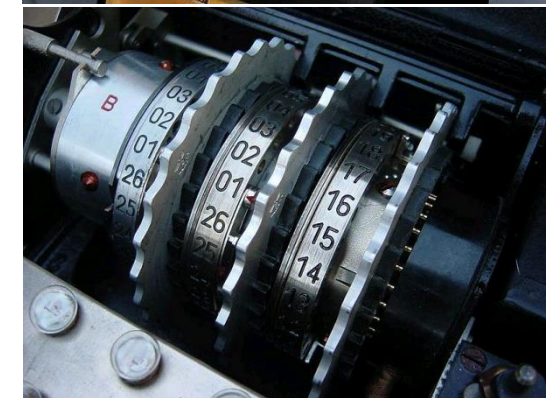
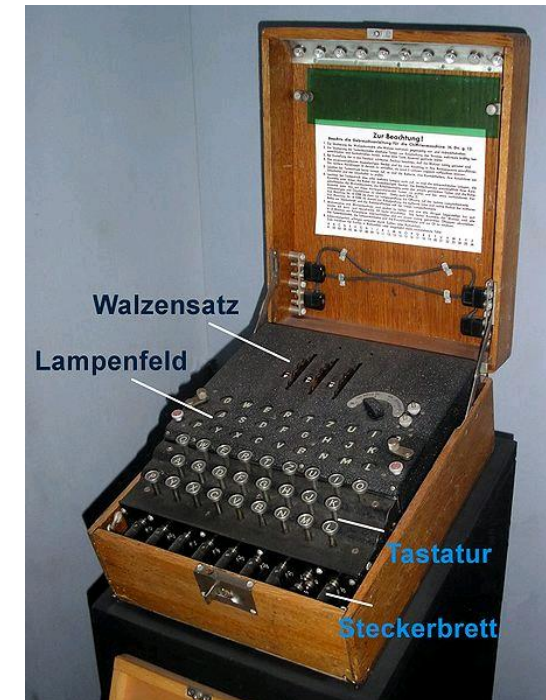
Enigma – Die Entstehungsgeschichte

- **Erfunden von Arthur Scherbius (1878-1929)**
 - Erstes Patent vom 23. Feb. 1918
 - Gründung der Firma Chiffriermaschinen AG (1923)
- **Konzipiert als ziviles Chiffriersystem**
 - Vorgestellt (und zum Verkauf angeboten) auf Messen
 - Postkongress in Bern (1923)
- **Verschwand Ende der 1920er vom zivilen Markt**
 - Verstärktes militärisches Interesse
- **Es wurden über 30.000 Maschinen produziert**
 - Sehr viele unterschiedliche Modelle
 - Meistgebrauchte: ENIGMA I (Reichswehr, Wehrmacht)

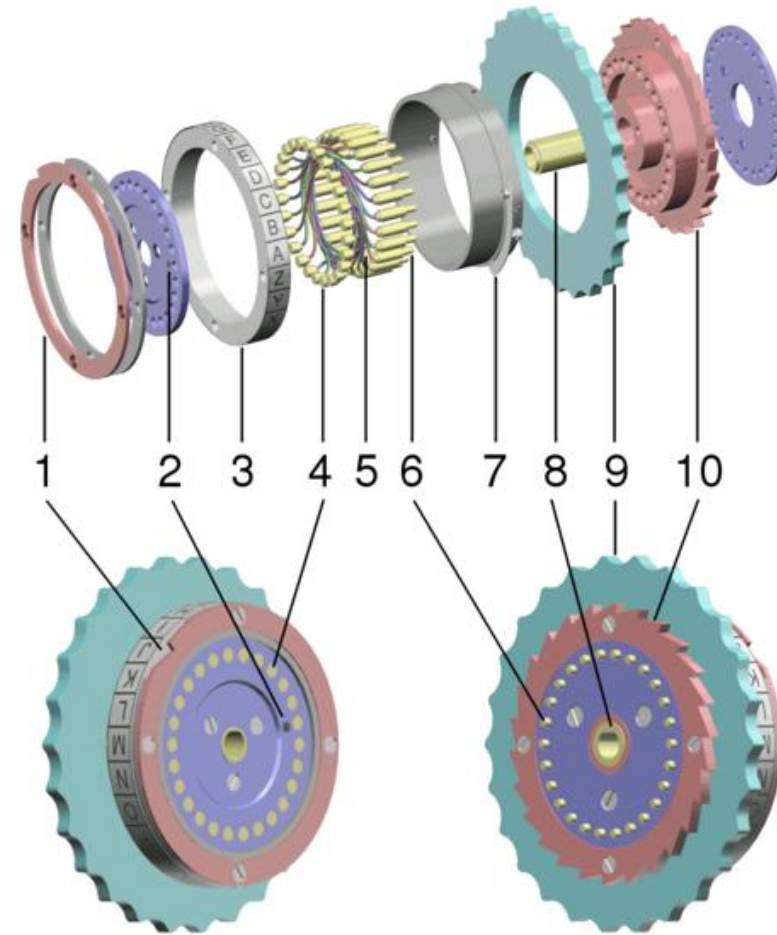
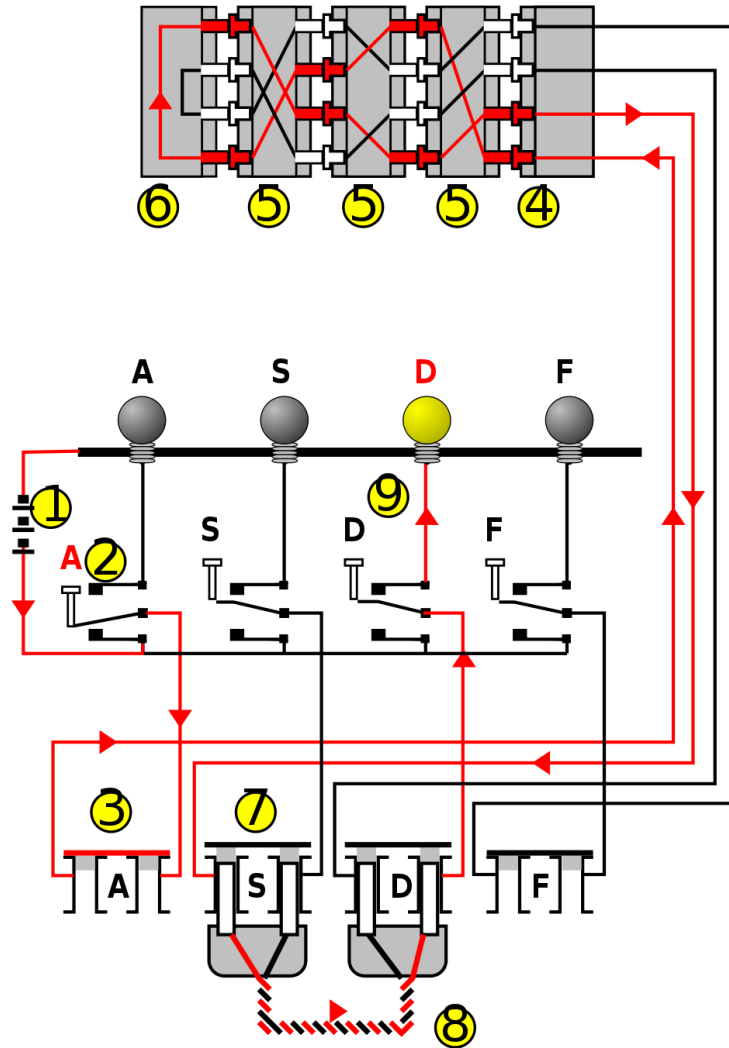
Enigma – Bestandteile



- **Tastatur (Eingabe)**
- **Lampenfeld (Ausgabe)**
- **Walzensatz (polyalphabetische Verschlüsselung)**
 - Wichtigstes Element
 - 26 Kontakte auf beiden Seiten
 - Unregelmäßig verbunden
 - Dreht mit jedem Buchstaben weiter
 - Austauschbar
- **Steckerbrett (monoalphabetische Verschlüsselung)**
 - Vertauscht einzelne Buchstaben

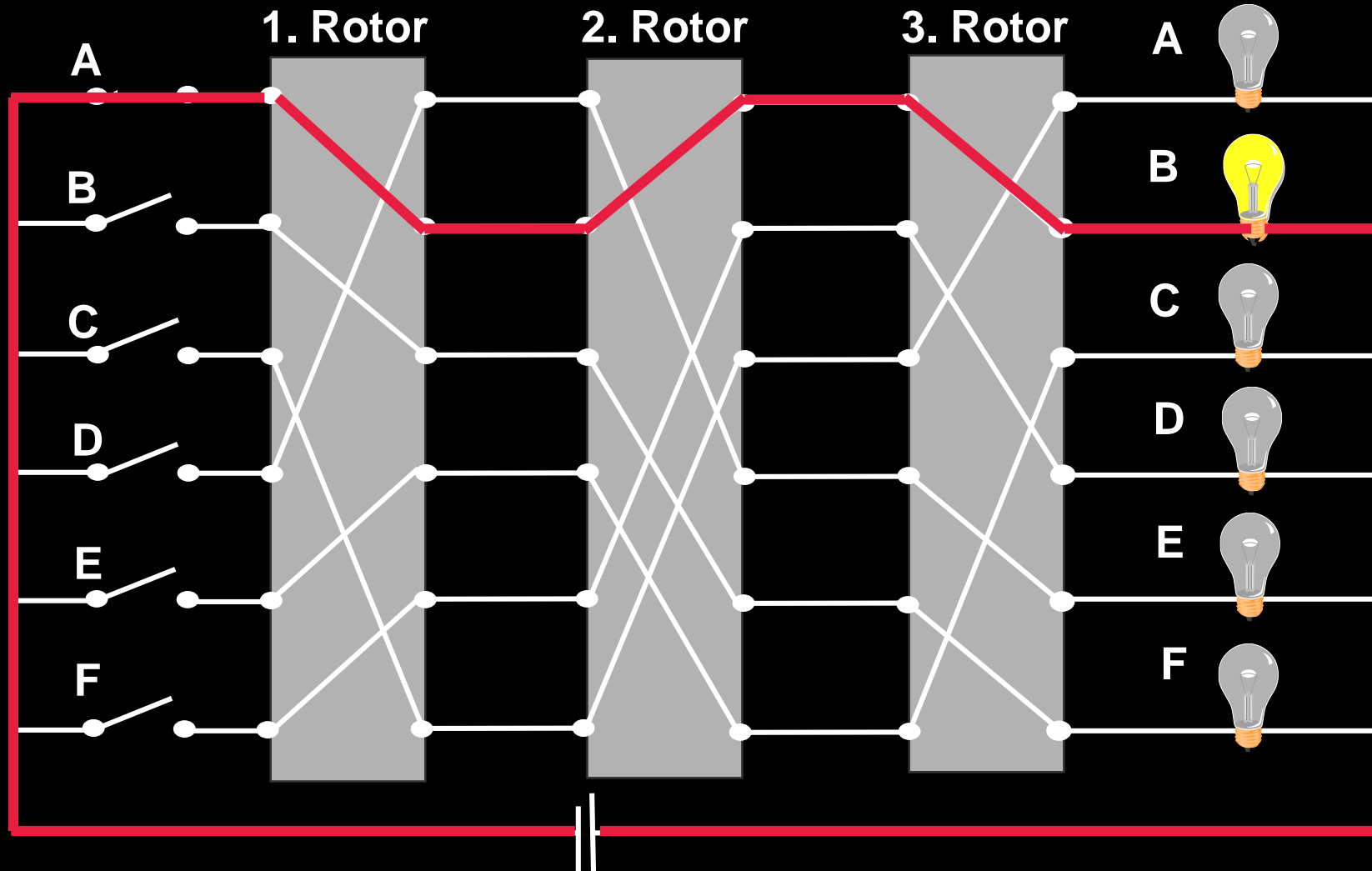


Enigma – Aufbau und Funktionsweise



Quelle: Wikipedia

Enigma – Stromlauf



Mit freundlicher Genehmigung
von Klaus Schmeh

Enigma – Vorbereitung

- **Tagesschlüssel besteht aus**
 - Auswahl von 3 Walzen (insgesamt gab es 5, bei M4 8)
 - Steckerverbindungen
 - Ringstellung
- **Beispiel einer Schlüsseltafel:**

Tag	UKW	Walzenlage	Ringstellung	-----	Steckerverbindungen	-----
31	B	I	IV III	16 26 08	AD CN ET FL GI JV KZ PU QY WX	
30	B	II	V I	18 24 11	BN DZ EP FX GT HW IY OU QV RS	
29	B	III	I IV	01 17 22	AH BL CX DI ER FK GU NP OQ TY	

Enigma – Funkspruch (1/3)

- **Verwendung eines Spruchschlüssels**
 - Ändert sich mit jeder Nachricht
 - Prozedur zur Erstellung wurde ein paarmal geändert
- **Prozedur für Spruchschlüssel**
 - Wahl einer zufälligen Grundstellung, z. B. „QWE“
 - Wahl eines zufälligen Schlüssels, z. B. „RTZ“
 - „RTZ“ verschlüsseln, man erhält „EWG“
 - Grundstellung und *verschlüsselter* Spruchschlüssel werden im Kopf der Nachricht versendet

22 : 20 – 204 – QWE EWG

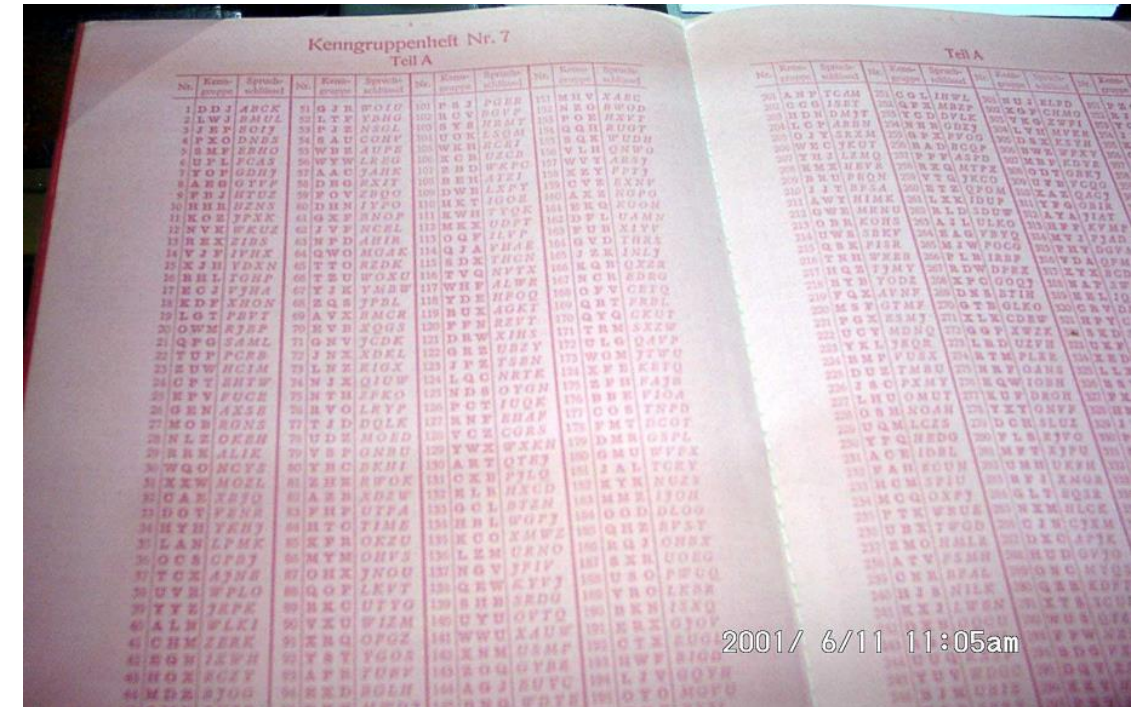
Enigma – Funkspruch (2/3)

• Kenngruppe

- Aus Kenngruppenbücher
- Dient als Empfänger-Identifikation, z. B. „NOW“
- Kenngruppe wurde permutiert, z. B. „OWN“
- Aufgefüllt mit beliebigen Buchstaben zu 5 Zeichen, z.B. XYOWN
- Diese 5 Zeichen wurden unverschlüsselt an den Anfang der Nachricht gestellt

• Vorbereiten der Nachricht

- Enigma kann nur 26 Buchstaben verschlüsseln
- Satzzeichen werden zu X
- Eigennamen verdoppelt und mit X eingeschlossen
- Zahlen ausgeschrieben
- „ch“ als Q geschrieben



Enigma – Funkspruch (3/3)

Das Oberkommando der Wehrmacht gibt bekannt: Aachen ist gerettet. Durch gebündelten Einsatz der Hilfskräfte konnte die Bedrohung abgewendet und die Rettung der Stadt gegen 18:00 Uhr sichergestellt werden.

DASOB	ERKOM	MANDO	DERWE	HRMAQ	TGIBT	BEKAN	NTXAA	CHENX	AACHE
NXIST	GERET	TETXD	URQGE	BUEND	ELTEN	EINSA	TZDER	HILFS	KRAEF
TEKON	NTEDI	EBEDR	OHUNG	ABGEW	ENDET	UNDDI	ERETT	UNGDE	RSTAD
TGEGE	NXEIN	SXAQT	XNULL	XNULL	XUHRS	IQERG	ESTEL	LTWER	DENX

22:20	- 204 -							QWE EWG	
XYOWN	LJPQH	SVDWC	LYXZQ	FXHIU	VWDJO	BJNZX	RCWEO	TVNJC	IONTF
QNSXW	ISXKH	JDAGD	JVAKU	KVMJA	JHSZQ	QJHZO	IAVZO	WMSCK	ASRDN
KKKSR	FHCXC	MPJGX	YIJCC	KISYY	SHETX	VVOVD	QLZYT	NJXNU	WKZRX
UJFXM	BDIBR	VMJKR	HTCUJ	QPTEE	IYNYN	JBEAQ	JCLMU	ODFWM	ARQCF
OBWN									

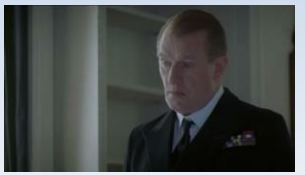
Enigma – Filmausschnitt



Enigma – Kryptoanalyse

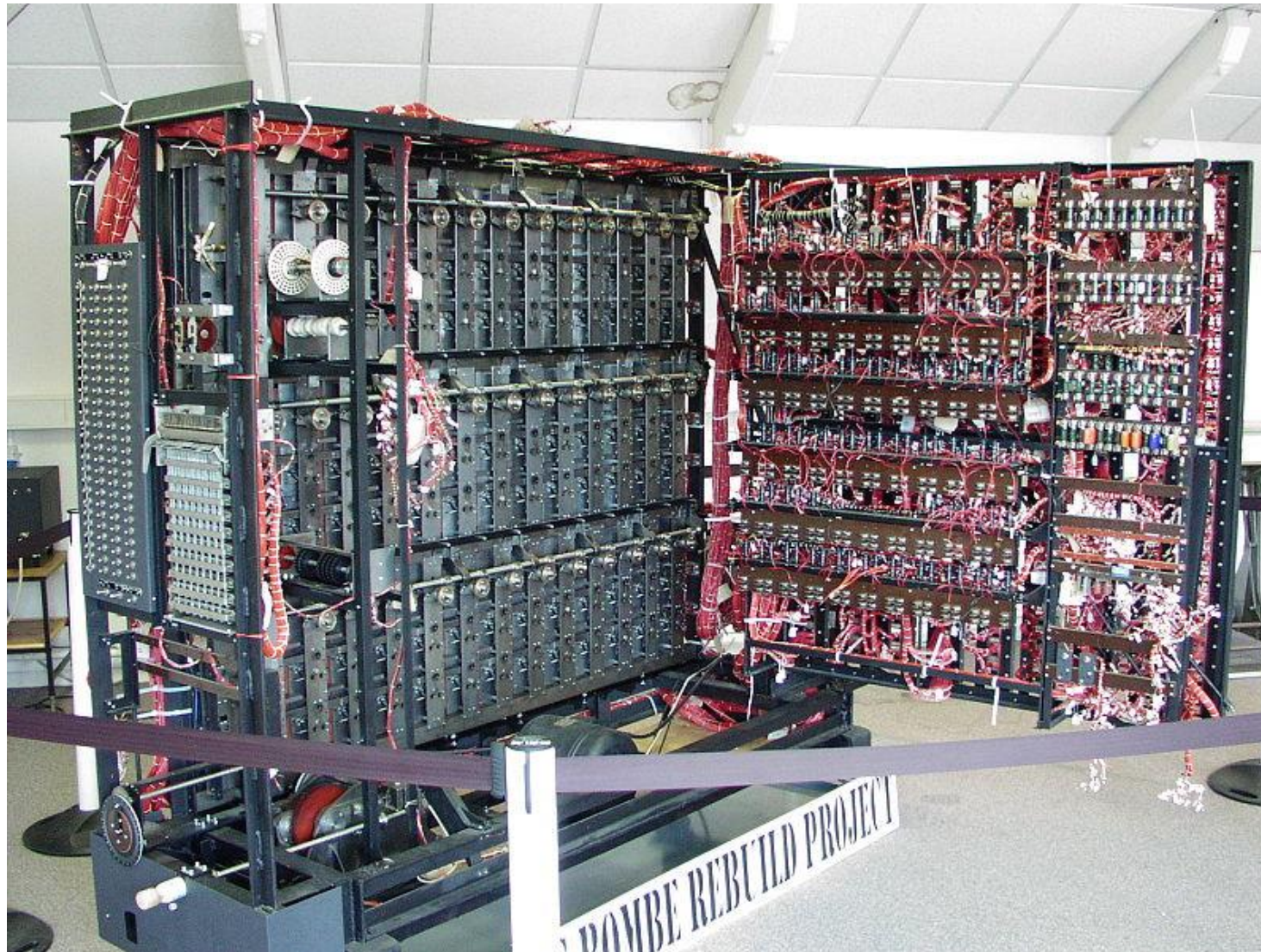
- **Erfunden in den 1920er – sicher bis ca. 1930**
- **Maschinelle Verschlüsselung**
- **Unangreifbar durch damalige Verfahren**
 - Periodenbestimmung unsinnig, da Periodenlänge 19.900 (bei 3 Walzen)
 - Häufigkeitsanalyse unbrauchbar
- **Entscheidend für die Sicherheit**
 - Geheimhaltung der Walzenverdrahtung
 - Anzahl der Walzen
 - Anzahl der Steckverbindungen

Enigma – Kryptoanalyse

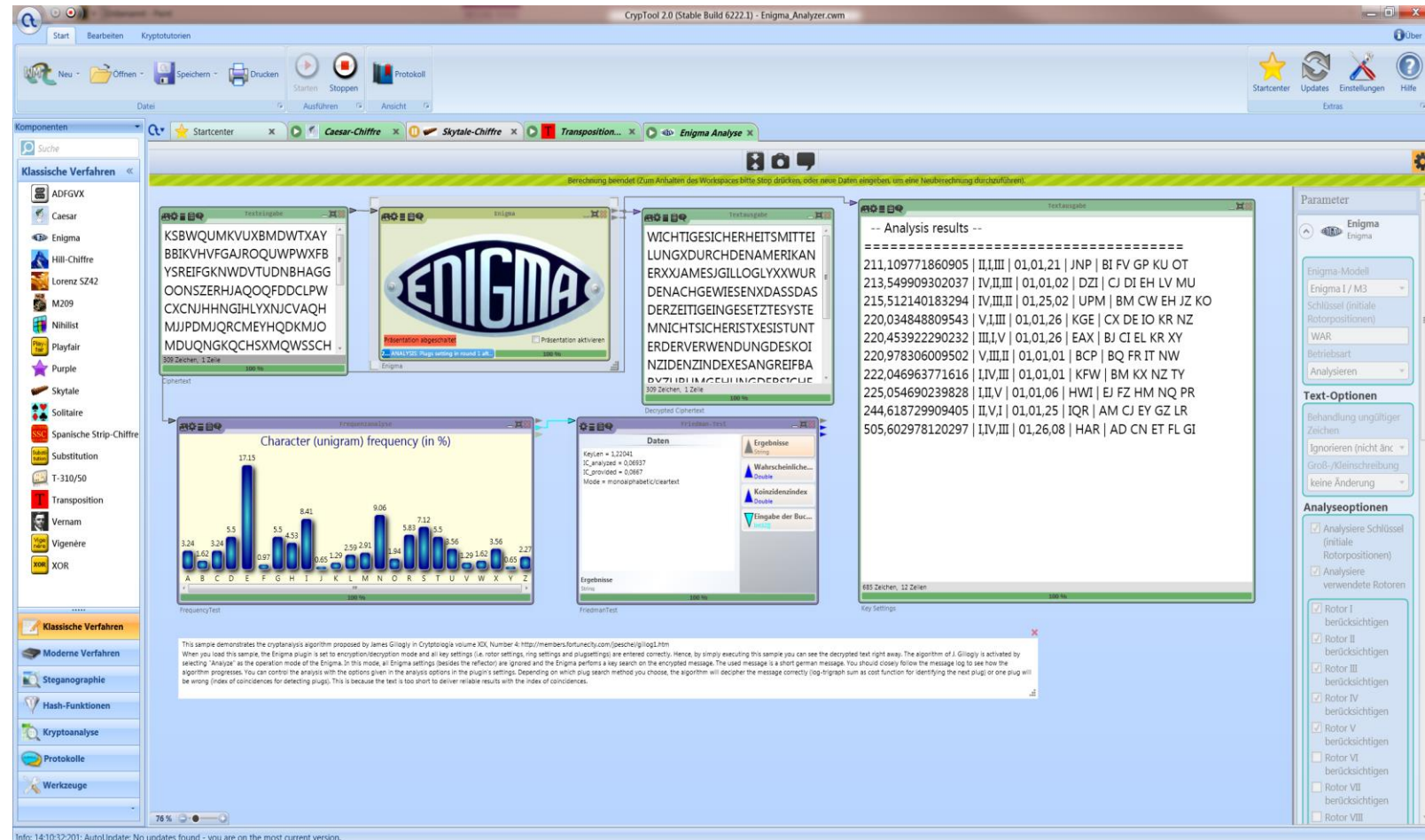


Enigma – Kryptoanalyse

Die Turing-Bombe



UNIKASSEL
VERSITÄT



Moderne Verschlüsselung

Schülerkrypto 2017

Beginn der Computerverschlüsselung

- **Erfindung des Computers**
 - Die ersten Computer waren mechanisch (Relais)
 - 1938 – 1948: Z1, Z3, Atanasoff-Berry, Colossus, ENIAC
 - Lösten erst in den 70er Jahren die mechanischen Verschlüsselungsgeräte ab
- **Was ist nun anders?**
 - Aus Buchstaben werden Bits
 - Text wird in ein binäres Format gewandelt (z.B. ASCII)
 - Zeichenvorrat/Alphabet steigt von 26 auf 256 (1 Byte)
 - Bits werden substituiert/transponiert
 - Geschwindigkeit nimmt drastisch zu → komplexere Algorithmen
- **Was bleibt gleich?**
 - Substitution und Transposition (Permutation)
 - Gleicher Schlüssel zum Ver- und Entschlüsseln (zunächst):
Symmetrische Kryptographie

Der Data Encryption Standard (DES)

Geschichtlicher Hintergrund (1/4)

- **Beteiligte US Regierungsstellen**
 - NBS: National Bureau of Standards
 - NIST: National Institute of Standards and Technology (ersetzte das NBS)
 - NSA: National Security Agency
 - Manchmal auch als “No Such Agency” bezeichnet
 - Weltgrößter Arbeitgeber für Mathematiker und Kryptologen
- **Industrie**
 - IBM

Der Data Encryption Standard (DES)

Geschichtlicher Hintergrund (2/4)

- **1973: Ausschreibung des NBS für einen nationalen Chiffrierstandard:**
 - Hohe Sicherheit
 - Vollständig spezifiziert und einfach zu verstehen
 - Sicherheit liegt nur im Schlüssel (Kerkhoffs' Prinzip)
 - Für alle verfügbar
 - Anpassbar für unterschiedliche Anforderungen
 - Ökonomisch/günstig integrierbar in elektronische Geräte
 - Einfach zu benutzen
 - Überprüfbar
 - Exportierbar

Der Data Encryption Standard (DES)

Geschichtlicher Hintergrund (3/4)

- **Erste Einreichungen entsprachen alle nicht den geforderten Kriterien**
- **IBM trat mit LUCIFER an**
 - Symmetrische Blockchiffre
 - Blockgröße: 128 Bit
 - Schlüssellänge: 128 Bit
- **NSA untersucht LUCIFER ... und verändert ihn**
 - DES verhält sich genau wie LUCIFER, aber
 - Blockgröße: 64 Bit
 - Schlüssellänge: 56 Bit (Blockgröße != Schlüssellänge)
 - Warum? Hat die NSA eine Hintertür eingebaut?

Der Data Encryption Standard (DES)

Geschichtlicher Hintergrund (4/4)

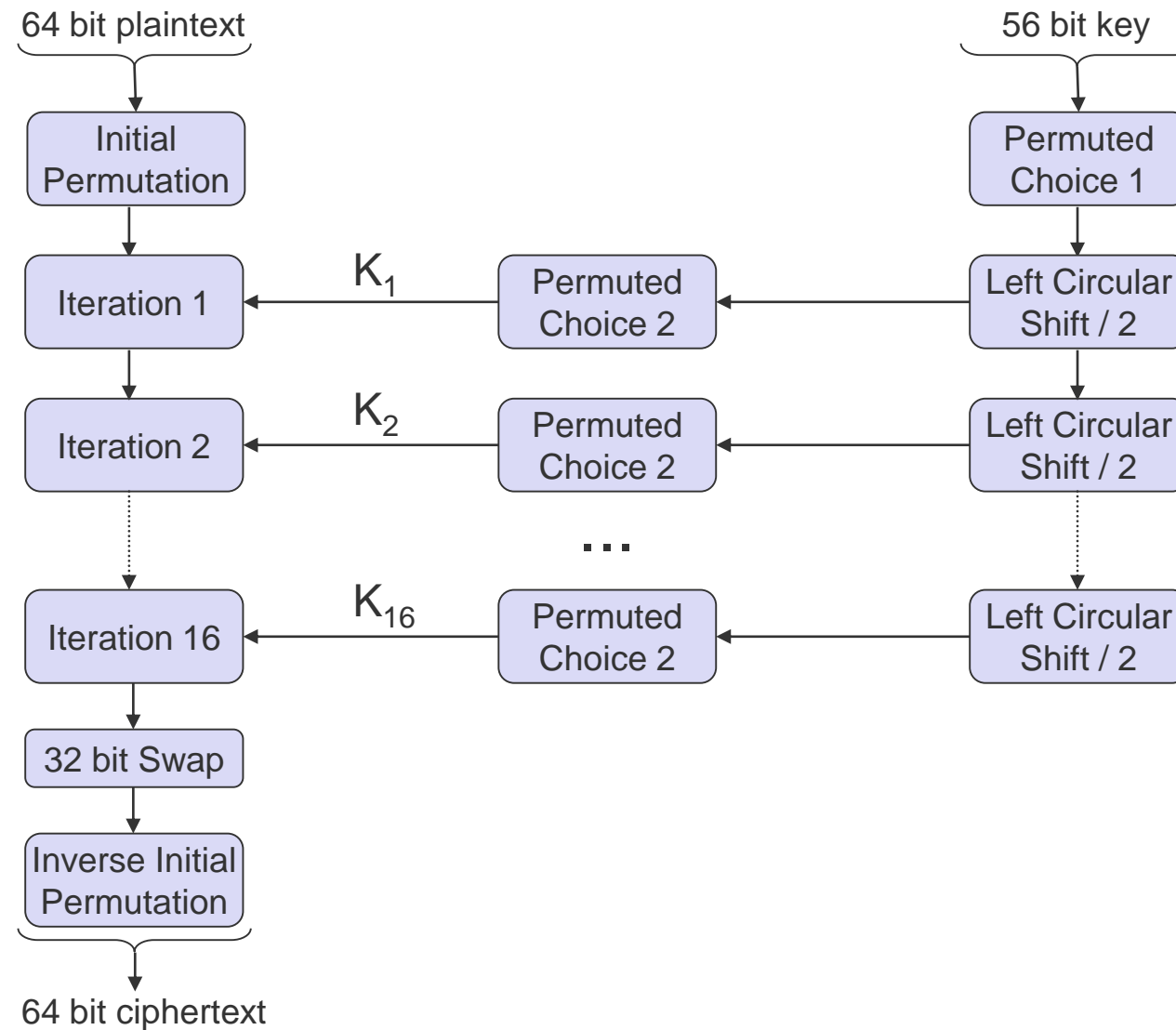
- **Verschwörungstheorie**

- Die NSA veränderte die sogenannten S-Boxen (da kommen wir gleich dazu)
- Kein Kryptologe konnte damals sagen, warum (die, die bei der NSA angestellt waren, durften nicht)
- Grundlage für viele Verschwörungstheorien

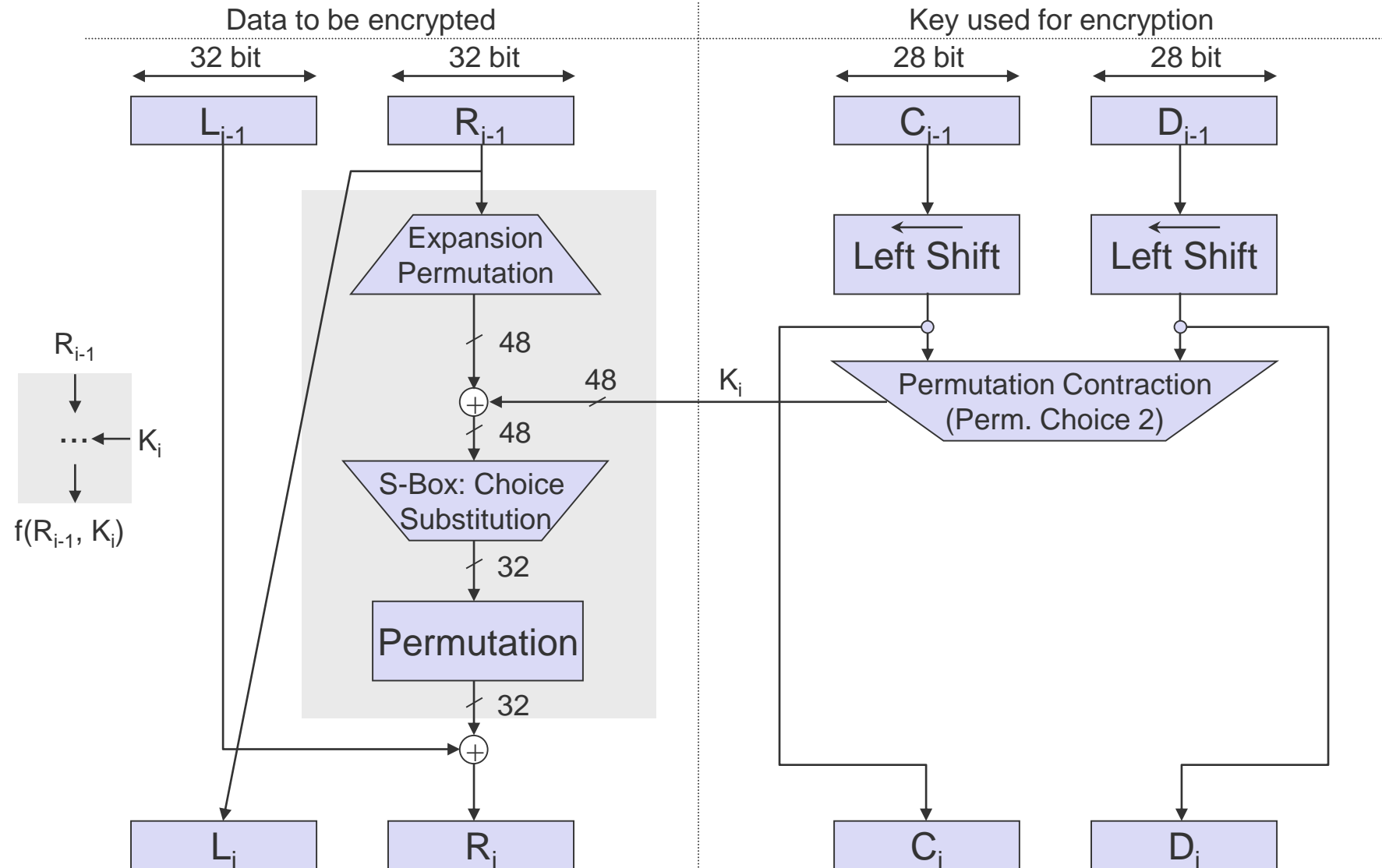
- **Was wirklich geschah**

- NSA kennt die „differentielle Kryptoanalyse“ seit den 70ern
- Durch die Änderungen wurde DES davor geschützt
- In den 90ern wurde die differentielle Kryptoanalyse (nochmal) entdeckt – durch öffentliche Einrichtungen
- Jetzt verstand die Welt, was die wirklichen Gründe waren
 - Die neue S-Box verstärkte den DES
 - Die gekürzte Schlüssellänge schwächte ihn

DES-Algorithmus Übersicht



Eine DES-Iteration (Runde)



Demo: DES mit CrypTool 1.4

The screenshot displays the CrypTool 1.4.30 interface with several windows open:

- Unbenannt1**: A text editor window showing the encrypted message: `00000000 45 69 6E 65 20 67 65 68 65 69 6D Eine geheim
0000000B 65 20 42 6F 74 73 63 68 61 66 74 e Botschaft
00000016 21`
- Schlüsseleingabe: DES (ECB)**: A dialog box for entering the key. The key length is set to 64 Bit (56 Bit effektiv). The key value is `01 23 45 67 89 10 11 12`. Buttons: **Verschlüsseln**, **Entschlüsseln**, **Abbrechen**.
- Brute-Force-Analyse von DES (ECB)**: A dialog box showing the progress of a brute-force search. It indicates: **Vollständige 21 Bit-Suche zu 7% erledigt. Restzeit: 00:00:32**. Button: **Abbrechen**.
- Brute-Force-Analyse**: A window showing the results of the brute-force analysis. It contains a table with two columns: **Entschlüsselung** and **Entschlüsselung: Hex-Dump**. The first row shows the correct decryption: `Eine geheime Botschaft!`. Buttons: **Auswahl übernehmen**, **Abbrechen**.

Der Advanced Encryption Standard (AES)

- **Jan 1997: Das NIST veröffentlicht die Absicht, einen DES-Nachfolger einzuführen**
 - Verwendung bei Regierung
 - Verwendung in der Industrie
- **Sep 1997: Öffentliche Ausschreibung**
 - Offizieller DES-Nachfolger, genannt AES
 - Jeder auf der Welt kann einen Algorithmus einreichen
 - Algorithmus muss offen und ohne Gebühren weltweit verwendbar sein

Der Advanced Encryption Standard (AES)

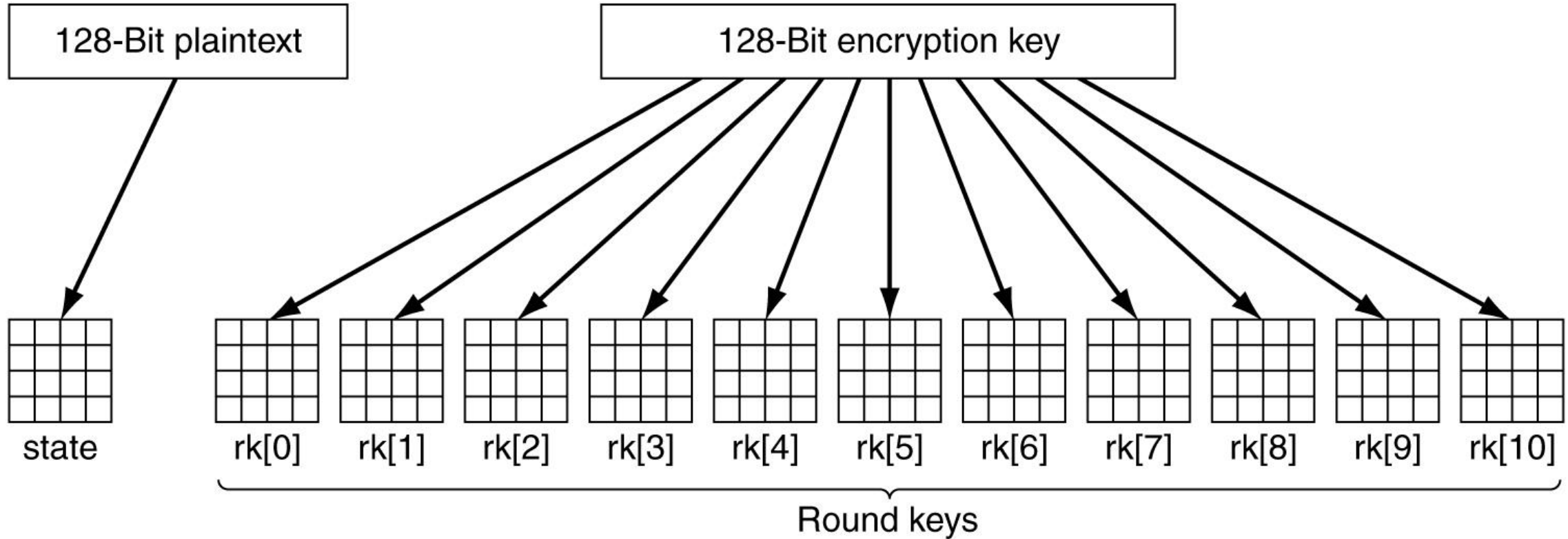
- **Aug 1998: Erste AES Konferenz**
 - Die besten 15 Kandidaten wurden gewählt
- **Mär 1999: Zweite AES Konferenz**
 - Analyse der Kandidaten durch Kryptologen
- **Apr 1999: Fünf „Finalisten“ werden gewählt**
 - MARS, RC6, Rijndael, Serpent, und Twofish
- **Okt 2000: Rijndael (=AES) wird offizieller DES-Nachfolger**

Der Advanced Encryption Standard (AES)

Details

- **Schlüssel- und Blocklängen**
 - Schlüssellänge: 128, 192 oder 256 Bit (Rijndael zusätzlich: 160, 224)
 - Blocklänge: 128 Bit (Rijndael: 160,192,224,256)
- **AES ist somit ein 128 Bit Blockchiffre**
 - Genau wie DES arbeitet auch AES in Runden (10)
- **Der Algorithmus arbeitet auf (128 Bit Schlüssel):**
 - state [4,4]: Byte-Tabelle mit 4 Zeilen und 4 Spalten
 - key[4,4]: Byte-Tabelle mit 4 Zeilen und 4 Spalten
 - Ein Schlüssel für jede Runde

Der Advanced Encryption Standard (AES)



Erstellen des *state* und des *rk* arrays

Der Advanced Encryption Standard (AES)

- **Runde 1 – 9**
 - ByteSub: eine nicht-lineare Substitution (S-Box)
 - ShiftRow: Die Zeile der state-Tabelle werden zyklisch geschoben (rotiert) – mit verschiedenen offsets
 - MixColumn: Man betrachtet die Spalten der state-Tabelle als Polynome im $GF(2^8)$ und multipliziert diese
 - RoundKey: Der Rundenschlüssel wird mit dem state XOR-verknüpft
- **Runde 10 ist ohne MixColumn**

Demo: AES mit CrypTool 1.4.40 & CrypTool 2.1

The image displays two versions of the CrypTool software interface. The top-left window is CrypTool 1.4.30, showing the 'Rijndael (AES)-Verschlüsselung' process. It includes a menu bar with options like 'Datei', 'Bearbeiten', and 'Ansicht'. A text area shows the plaintext 'Eine geheime Botschaft!' being encrypted into hexadecimal. A separate dialog box titled 'Schlüssel eingabe: Rijndael (AES)' prompts for a key, showing a 128-bit key: '01 23 45 67 89 10 11 12 13 14 15 16 17 18 19 20'. The bottom-left window shows the 'Einzelverfahren' menu with 'AES' selected, leading to 'Rijndael-Animation...'. The right window is CrypTool 2.0 (Stable Build 6222.1), showing a more complex workflow with multiple components like 'AES', 'PKCS5', and 'StreamCipher' connected in a sequence. It includes a 'Parameter' panel on the right for configuring the AES algorithm.

CrypTool 1.4.30 - Rijndael (AES)-Verschlüsselung von <Unbenannt1>, Schlüssel <01 23 45 67 89 1...

Datei Bearbeiten Ansicht Ver-/Entschlüsseln Digitale Signaturen/PKI Einzelverfahren Analyse Optionen Fenster Hilfe

Unbenannt1

00000000	45 69 6E 65 20 67 65 68 65 69 6D	Eine geheim
0000000B	65 20 42 6F 74 73 63 68 61 66 74	e Botschaft
00000016	21	!

Rijndael (AES)-Verschlüsselung von <Unbenannt1>

00000000	27 B0 12 AD 15 E8 91 99 A1
0000000A	BD 08 22 31 B5 81 29 49 98
00000014	CF C0 66 D3 84 EB AE 51 20
0000001E	6B 0A

Schlüssel eingabe: Rijndael (AES)

Geben Sie den Schlüssel mit Hexadezimalzeichen (0..9, A..F) ein.

Schlüssellänge: 128 Bit

01 23 45 67 89 10 11 12 13 14 15 16 17 18 19 20

Einzelverfahren Analyse Optionen Fenster Hilfe

- Hashverfahren
- RSA-Kryptosystem
- Protokolle
- Anwendungen des Chinesischen Restsatzes
- Visualisierung von Algorithmen
 - Caesar...
 - Vigenère...
 - Nihilist...
 - DES...
 - AES**
 - Rijndael-Animation...
 - Rijndael-Inspektor...
 - Rijndael-Flussvisualisierung...
 - Enigma...
- Lernspiele
- Zahlentheorie interaktiv

CrypTool 2.0 (Stable Build 6222.1) - AES PKCS5.com

Start Bearbeiten Kryptotutorials

Neu Öffnen Speichern Drucken Starten Stoppen Protokoll Ausführen Ansicht

Komponenten

- Klassische Verfahren
 - ADFGVX
 - Caesar
 - Enigma
 - Hill-Chiffre
 - Lorenz SZ42
 - M209
 - Nihilist
 - Playfair
 - Skytale
 - Solitaire
 - Spanische Strip-Chiffre
 - Substitution
- Moderne Verfahren
 - Steganographie
 - Hash-Funktionen
 - Kryptoanalyse
 - Protokolle
 - Werkzeuge

Parameter

AES

Kryptographischer Algorithmus: Advanced Encryption Standard (AES)

Aktion: Verschlüsselung

Schlüssellänge: 256 Bits

Blockverkettingsmodus: Cipher Feedback (CFB)

Auflös-Methode (Padding): PKCS7

Workflow components: Texteingabe, AES, PKCS5, StreamCipher, BooleanOutput, StringDecoder, Textausgabe.

Textausgabe 1: In cryptography, the Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. 500 Zeichen, 1 Zeile

Textausgabe 2: 30 C6 39 B9 7B 95 17 B2 B6 35 93 E1 6E 6B 02 1F 79 5C BA 61 2C 5A 62 D9 34 63 40 51 89 82 D3 6B 82 43 87 1D 89 42 BE 26 53 A9 61 17 65 AF F5 97 8F 2A 1E 30 0C 54 76 28 0C 1F 55 14 BD B5 A7 F0 06 DA 37 CB 7D 96 D7 D6 E6 87 1535 Zeichen, 1 Zeile

Textausgabe 3: In cryptography, the Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger 500 Zeichen, 1 Zeile

Info: 14:17:51:476: NOTE: No IV provided. Using 0x0000.000

Das Schlüsselaustauschparadoxon

- „Austausch von geheimen Botschaften ohne vorherigen Austausch eines Geheimnisses“
- Die meisten Kryptologen glaubten nicht an eine Lösung des Schlüsselaustauschparadoxons
- Das folgende Beispiel gab Hoffnung:

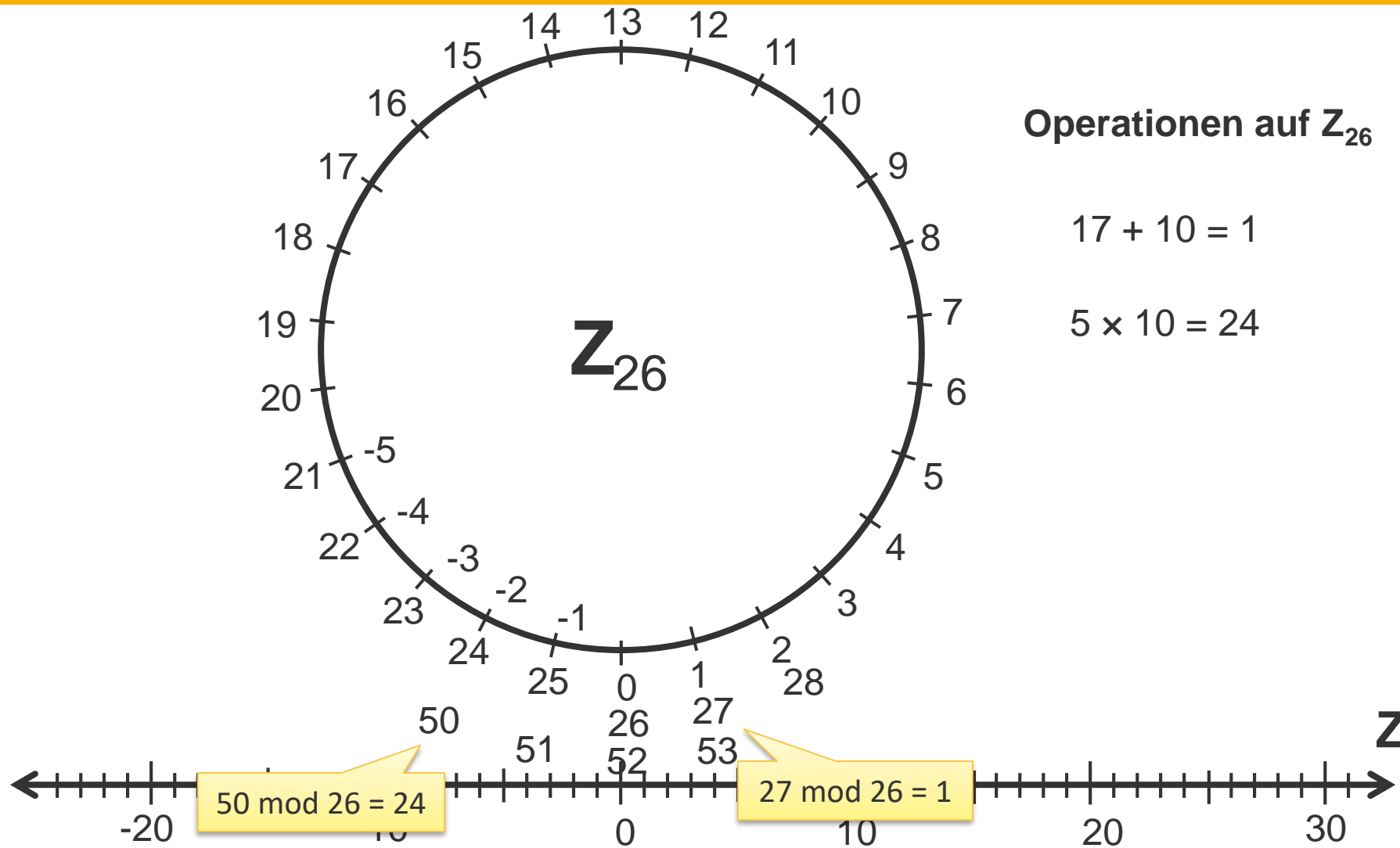
Demo: Schlüsselaustausch



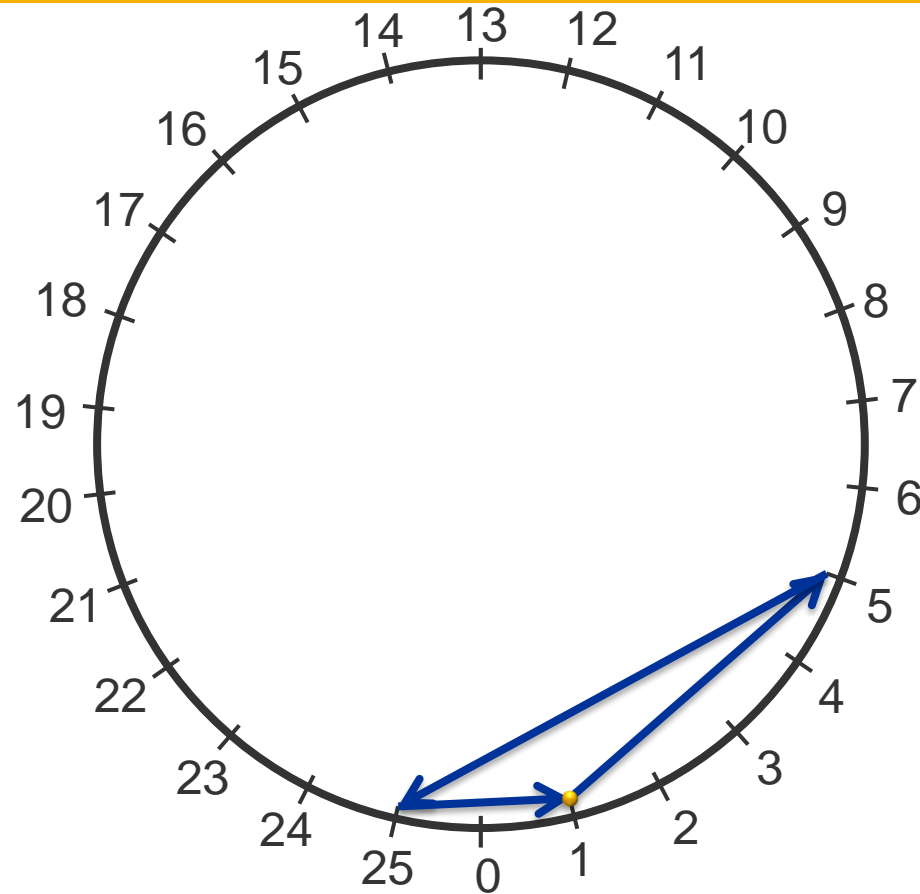
Schlüsselaustausch

- **Problem: Reihenfolge ist wichtig bei Verschlüsselung**
- **Zweifache Verschlüsselung einer Nachricht (2 Schlösser)**
 - $c = E_{c_2}(E_{c_1}(m))$
- **Entschlüsselung muss in der richtigen Reihenfolge sein!**
 - $m = D_{c_1}(D_{c_2}(c))$
 - Allgemein gilt $E_{c_2}(E_{c_1}(m)) \neq D_{c_2}(D_{c_1}(c))!!$
 - Nur wenige Ausnahmen, z.B. Caesar
- **Suche nach speziellen mathematischen Funktionen**
 - ➔ Falltürfunktionen (einfach in eine Richtung, schwer in die andere)
 - ➔ Rechnen in Ringen

Rechnen in Ringen (1)



Rechnen in Ringen (2)



5^n in \mathbb{Z}_{26}

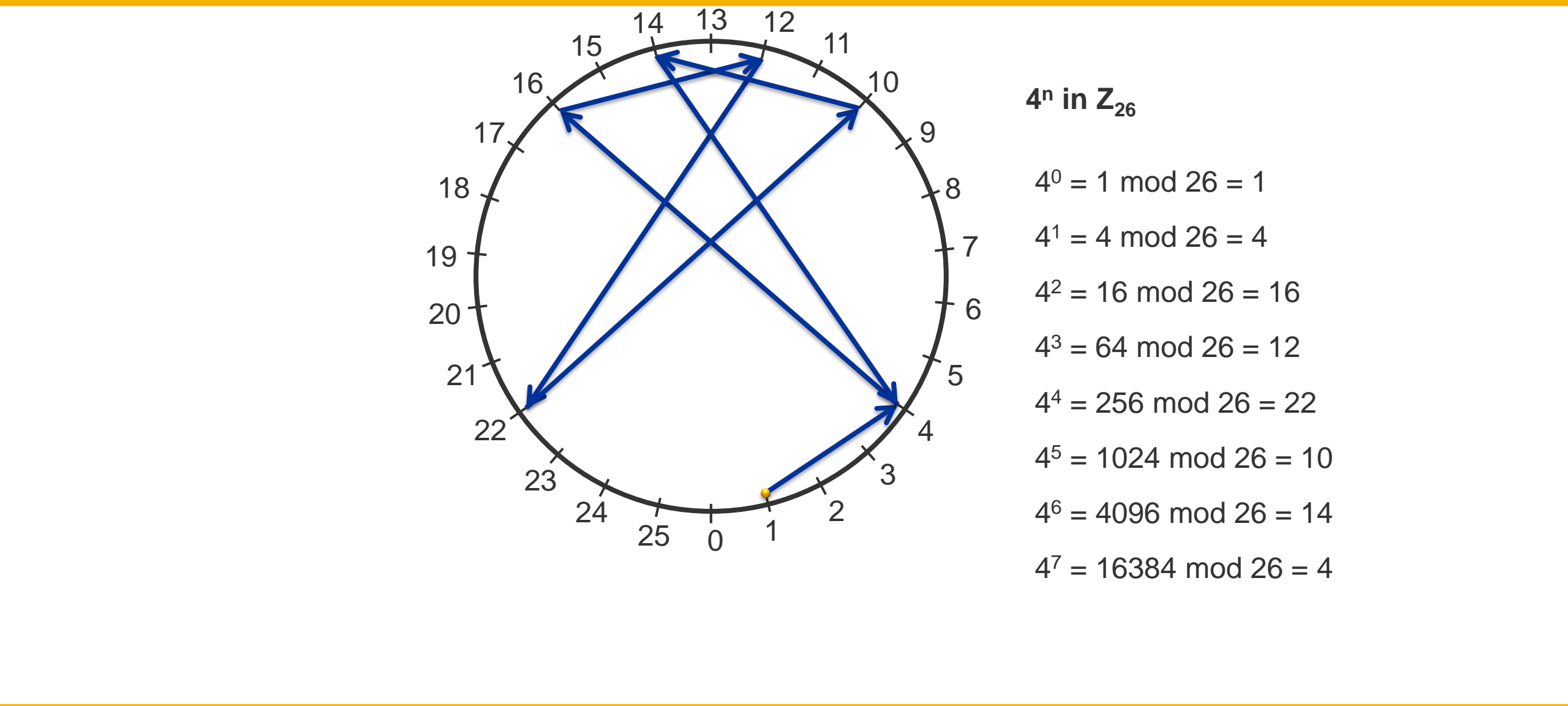
$$5^0 = 1 \bmod 26 = 1$$

$$5^1 = 5 \bmod 26 = 5$$

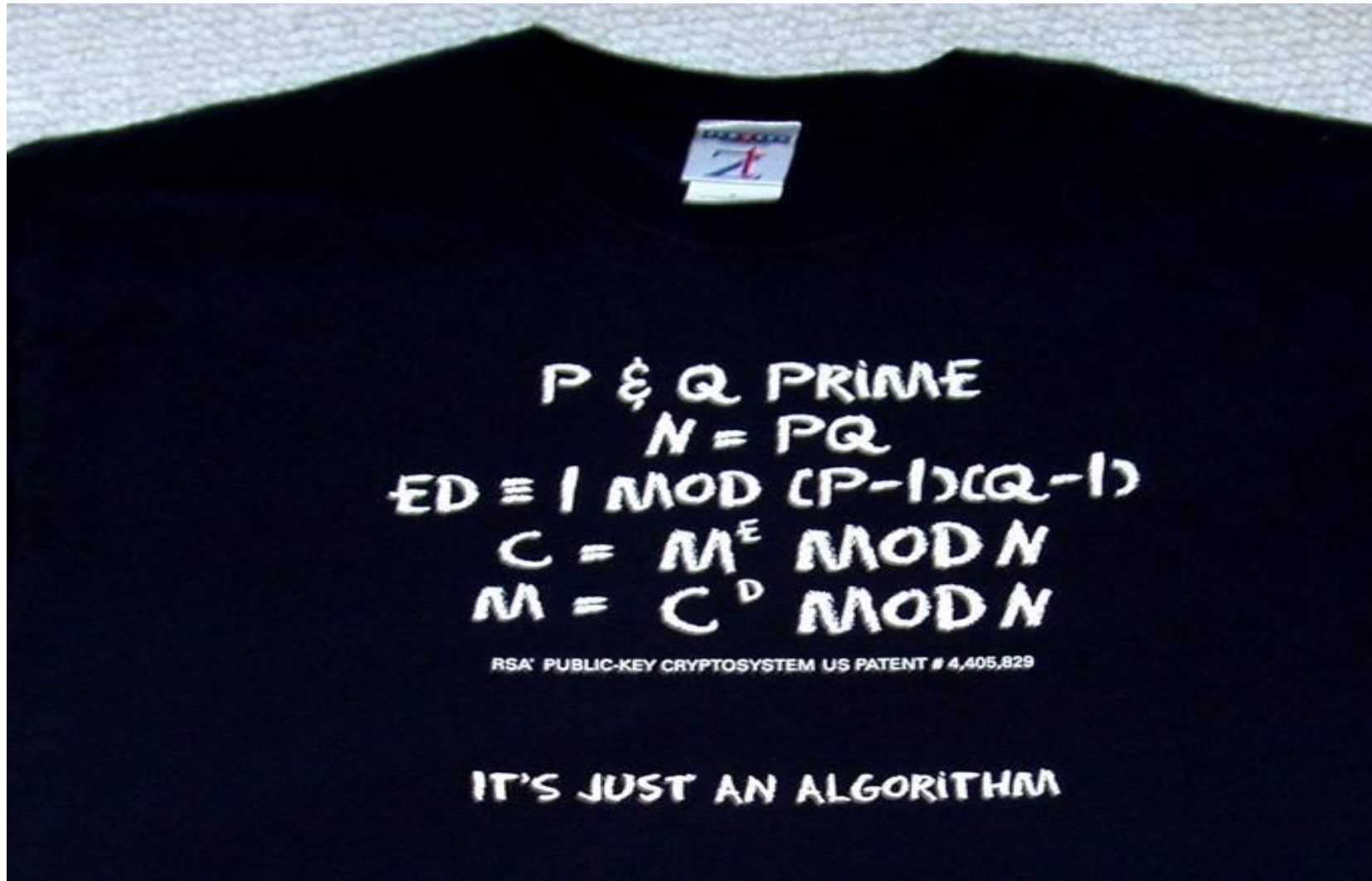
$$5^2 = 25 \bmod 26 = 25$$

$$5^3 = 125 \bmod 26 = 1$$

Rechnen in Ringen (3)


$$\begin{aligned}4^0 &= 1 \bmod 26 = 1 \\4^1 &= 4 \bmod 26 = 4 \\4^2 &= 16 \bmod 26 = 16 \\4^3 &= 64 \bmod 26 = 12 \\4^4 &= 256 \bmod 26 = 22 \\4^5 &= 1024 \bmod 26 = 10 \\4^6 &= 4096 \bmod 26 = 14 \\4^7 &= 16384 \bmod 26 = 4\end{aligned}$$

RSA



RSA Schlüsselerzeugung (1)

1. Wähle zwei große Primzahlen p, q
 - Jeweils mindestens 200 Dezimalstellen
2. Berechne $n = p \cdot q$
3. Berechne $\Phi(n) = \Phi(p, q) = (p-1)(q-1)$
4. Wähle eine Zahl e teilerfremd zu $\Phi(n)$
 - Mathematisch: $\text{ggT}(e, \Phi(n)) = 1$
 - Jede Primzahl $< \Phi(n)$ kann verwendet werden
5. Berechne c, d mit dem erweiterten euklidischen Algorithmus:
$$e \cdot d + c \cdot \Phi(n) = 1 \quad \text{oder} \quad e \cdot d = k \cdot \Phi(n) + 1$$
 - Somit gilt: $e \cdot d \bmod \Phi(n) = 1$

RSA Schlüsselerzeugung (2)

- Öffentlicher Schlüssel: (e,n)
 - Privater Schlüssel: (d,n)
 - Lösche die Primzahlen p,q
 - Warum?
 - Angenommen ein Angreifer kennt p,q
 - Somit kennt er sofort $\Phi(n) = (p-1)(q-1)$
 - Jetzt kann er d berechnen
 - Der private Schlüssel ist nicht mehr geheim ☹️
- **RSA ist sicher,
solange es in der Praxis zu aufwändig ist, p,q zu berechnen**

RSA Ver- und Entschlüsselung

- Sei $m < n$ eine zu verschlüsselnde Nachricht
- Verschlüsselung
$$c = m^e \bmod n$$
- Entschlüsselung
$$m' = c^d \bmod n$$
- Derselbe Algorithmus für Ver-/Entschlüsselung
- $m = m'$ ist mathematisch beweisbar
→ RSA ist formal korrekt 😊

RSA Beispiel – Überblick

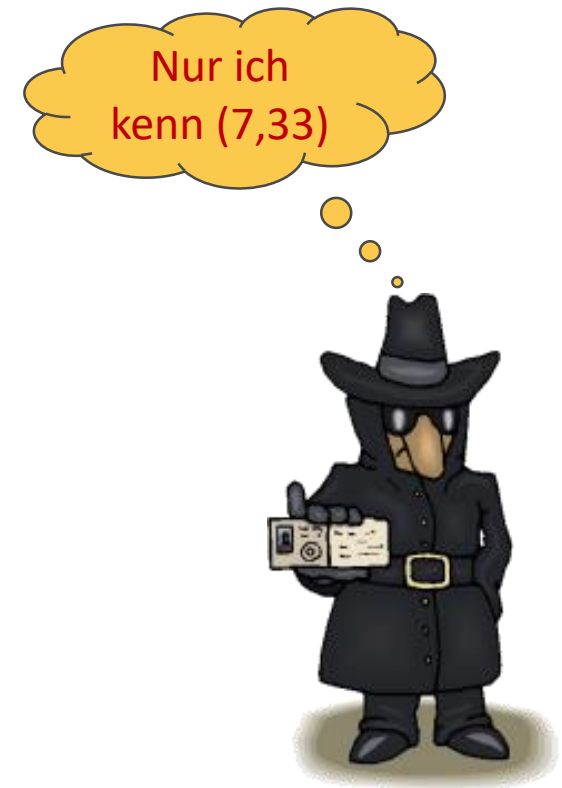
Alice schickt eine geheime Nachricht an Bob



1. Bob muss ein Schlüsselpaar generieren
2. Alice muss mit Bobs öffentlichem Schlüssel die Nachricht verschlüsseln

RSA Beispiel – Schlüsselerzeugung

- Wir erzeugen ein Schlüsselpaar für Bob
- Wähle zwei Primzahlen, z. B. $p=3$ und $q=11$
 - Sehr kleine Primzahlen im Beispiel
- Berechne $n = 3 \cdot 11 = 33$
- Berechne $\Phi(33) = (3-1)(11-1) = 20$
- Wähle e teilerfremd zu $\Phi(33) = 20$, z. B. $e = 3$
- Berechne ein $d < \Phi(33)$, sodass $3 \cdot d = k \cdot 20 + 1$
 - Bei $k=1$ ist $d=7$, denn $3 \cdot 7 = 1 \cdot 20 + 1$
- Bobs öffentlicher Schlüssel ist $(e, n) = (3, 33)$
- Bobs privater Schlüssel ist $(d, n) = (7, 33)$



RSA Beispiel – Verschlüsselung

- Alice möchte die Zahl 4 verschlüsseln
 - Somit ist die Nachricht $m=4$
- Alice kennt Bobs öffentlichen Schlüssel
 - $(e,n) = (3,33)$
- Alice berechnet Chiffretext $c = m^e \bmod n$
 - $c = 4^3 \bmod 33 = 64 \bmod 33 = 31$
- Alice sendet 31 an Bob



RSA Beispiel – Verschlüsselung

- Bob empfängt $c = 31$
- Nur Bob kennt seinen privaten Schlüssel
 - $(d, n) = (7, 33)$
- Bob berechnet Klartext $m = c^d \bmod n$
 - $m = 31^7 \bmod 33 = 27512614111 \bmod 33 = 4$



RSA Beispiel – Kryptoanalyse

- **Und Dave? Er kennt**
 - Bobs öffentlichen Schlüssel $(e,n) = (3,33)$
 - Die verschlüsselte Nachricht $c = 31$
- **Um c zu entschlüsseln braucht er aber (d,n)**
- **Dave ist nicht dumm:**
 - Er faktorisiert $n = 33 = 3 \cdot 11$
 - Nun berechnet er $\Phi(33) = (3-1)(11-1) = 20$
 - .. und d so dass $3 \cdot d = k \cdot 20 + 1$, und findet $d=7$!
 - Mit $d=7$ kann er die Nachricht auch entschlüsseln

Ich muss nur 33
faktorisieren...



RSA – Kryptoanalyse

- **RSA ist so schwierig wie Faktorisieren**
 - Wenn man Faktorisieren kann, kann man RSA brechen
- **Algorithmen zur Faktorisierung**
 - Probedivision (langsam für wenig Primfaktoren)
 - Kettenbruchmethode, CFRAC (besser)
 - Quadratisches Sieb, QS (ziemlich gut)
 - Allgemeines Zahlenfeldsieb, GNFS (bester bekannter Algorithmus)
- **Programme zur Faktorisierung**
 - CrypTool 1.4.40 (QS, Probedivision) – langsam
 - CrypTool 2.1 (QS, Probedivision) – schnell (msieve) & verteilt
 - ggnfs (GNFS) – Projekt wird nicht mehr gepflegt
 - msieve (QS, GNFS) – QS schnell (nicht verteilt), GNFS noch beta

Demo: RSA mit CrypTool 1.4.40

Primzahlen generieren

Primzahlen spielen in der modernen Kryptographie eine wichtige Rolle. Hier können Sie sich Primzahlen aus einem vorgegebenen Wertebereich [Untergrenze, Obergrenze] erzeugen.

Anzahl der zu generierenden Primzahlen:

- ☒ Zwei Primzahlen zufällig aus dem Wertebereich (den Wertebereichen) generieren
- ☐ Alle Primzahlen in dem (für p vorgegebenen) Wertebereich generieren

Trennzeichen für die Ausgabe der Primzahlen:

Algorithmen zur Generierung:

- ☒ Miller-Rabin-Test
- ☐ Solovay-Strassen-Test
- ☐ Fermat-Test

Wertebereich der Primzahlen:

- ☒ Unabhängig eingeben
- ☐ Beide gleich

Primzahl p

Untergrenze:

Obergrenze:

Ergebnis:

Primzahl q

Untergrenze:

Obergrenze:

Ergebnis:

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

- ☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.
- ☐ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e.

Primzahleingabe

Primzahl p:

Primzahl q:

RSA-Parameter

RSA-Modul N: (öffentlich)

$\phi(N) = (p-1)(q-1)$: (geheim)

Öffentlicher Schlüssel e:

Geheimer Schlüssel d:

RSA-Verschlüsselung mit e / Entschlüsselung mit d

Eingabe als: ☒ Text ☐ Zahlen

Eingabetext:

Der Eingabetext wird in Blöcke der Länge 1 aufgeteilt (das Symbol '#' dient als Trennzeichen).

Zahlendarstellung der Eingabe zur Basis 10.

Verschlüsselung in den Chiffretext $c[i] = m[i]^e \pmod{N}$.

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

- ☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.
- ☐ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e.

Primzahleingabe

Primzahl p:

Primzahl q:

RSA-Parameter

RSA-Modul N: (öffentlich)

$\phi(N) = (p-1)(q-1)$: (geheim)

Öffentlicher Schlüssel e:

Geheimer Schlüssel d:

RSA-Verschlüsselung mit e / Entschlüsselung mit d

Eingabe als: ☐ Text ☒ Zahlen

Chiffretext in Zahlendarstellung zur Basis 10.

Entschlüsselung in den Klartext $m[i] = c[i]^d \pmod{N}$.

Ausgabebetext aus der Entschlüsselung (in Blöcken der Länge 1; das Symbol '#' die Trennzeichen).

Klartext

Faktorisieren einer Zahl

Algorithmen zur Faktorisierung:

- ☒ Brute-Force
- ☒ Brent
- ☒ Pollard
- ☒ Williams
- ☒ Lenstra
- ☒ Quadratisches Sieb

Eingabe

Geben Sie die zu faktorisierende Zahl ein:

Faktorisierung (schrittweise)

Durch das Anklicken des Buttons "Weiter" wird initial die Zahl im Eingabefeld und dann jeweils die nächste zusammengesetzte Zahl im Feld "Produktdarstellung" in zwei Faktoren zerlegt.

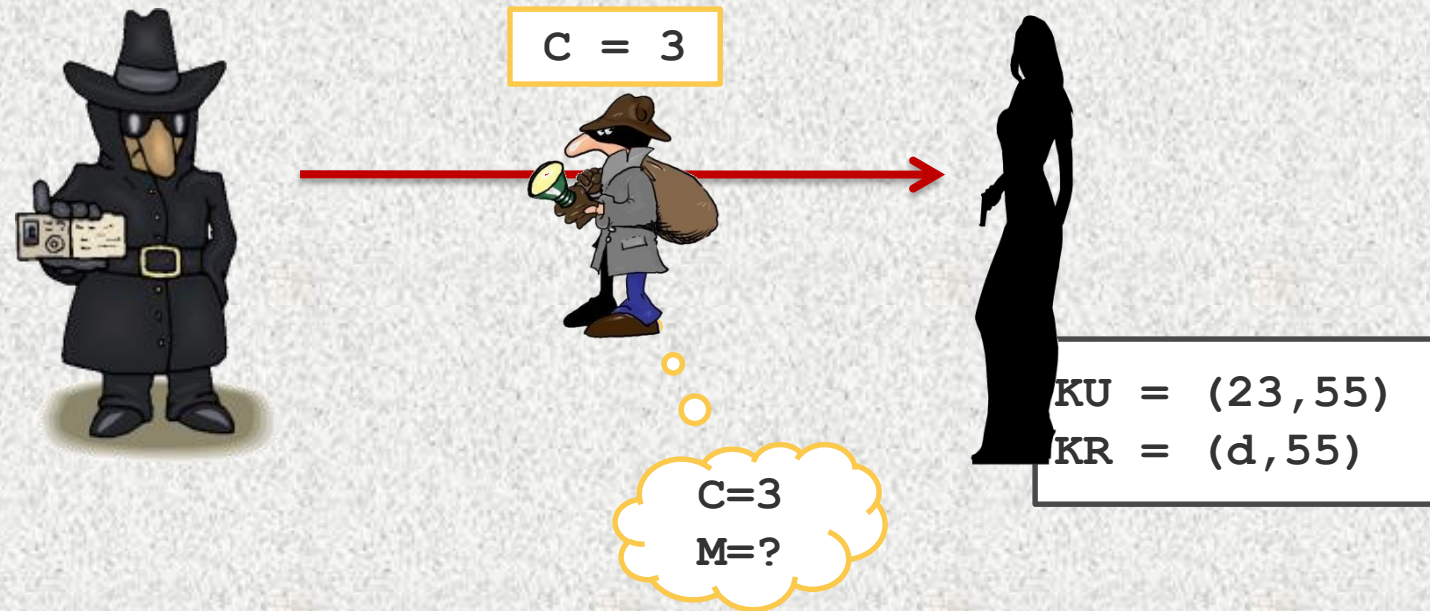
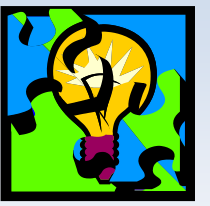
Faktorisierungsergebnis

Die Faktorisierung wird in dem Format $\langle z_1^{a_1} \cdot z_2^{a_2} \cdot \dots \cdot z_n^{a_n} \rangle$ dargestellt. Zusammengesetzte Zahlen sind rot markiert.

Letzte Faktorisierung durch: 2 Faktoren gefunden in 0,015 Sekunden.

Produktdarstellung der Faktorisierung:

Das RSA – Rätsel



Welche Nachricht hat Bob an Alice gesendet?

Kryptoanalyse – RSA-Rätsel

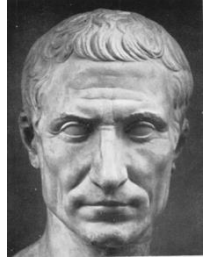
- **Faktorisierung** von $n = p \cdot q$
 - $55 = 5 \cdot 11$
- Aus p und q berechnet man $\Phi(n) = (p-1)(q-1)$
 - $\Phi(55) = (5-1)(11-1) = 4 \cdot 10 = 40$
- Finde d so dass gilt $e \cdot d = k \cdot \Phi(n) + 1$
 - $23 \cdot d = k \cdot 40 + 1$
 - Durch probieren, mit $k = 1, 2, 3..$ findet man $k=4$
 - $23 \cdot d = 4 \cdot 40 + 1 = 161 \rightarrow d=161/23 = 7$
- **Entschlüsseln:** $M = C^d \bmod n$
 - $M = 3^7 \bmod 55$
 - $M = 2187 \bmod 55 = 42$

Zusammenfassung

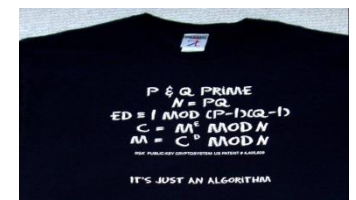
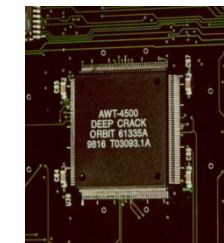
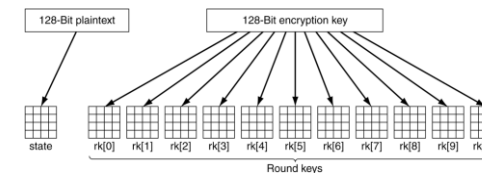
Schülerkrypto 2017

Zusammenfassung

- **Einfache Verschlüsselung**
 - Skytale
 - Caesar
- **Verbesserte Verschlüsselung**
 - Spaltentransposition
 - Substitution
- **Maschinelle Verschlüsselung**
 - Enigma
- **Moderne Verschlüsselung**
 - Symmetrisch: DES, AES
 - Asymmetrisch: RSA



UJC C00R>JN000 UF
J0000000000 J00000
J <J>>J J000 C00R
>JN000 UFJ000000000
J J0000000000 VJ
J<0000000000 J0000
J <J>>J J000 0J0
>J0000000000000000
0 0000000000000000
00000 J00 0J000
00000 <0J00 00000
00000 00000000000
>000 000000



**Vielen Dank
für Ihre Aufmerksamkeit!**

Prof. Arno Wacker
arno.wacker@uni-kassel.de