Geschichten der Kryptographie

- WebMontag Erfurt -

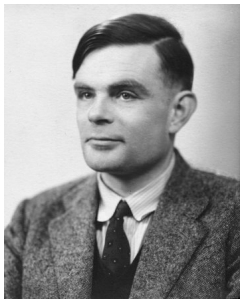
Dr. Sascha Grau

Technische Universität Ilmenau

23. September 2013

So viel zu sagen, so wenig Zeit...

Heute: Entwicklung, Konzepte, *ein paar Geschichten*



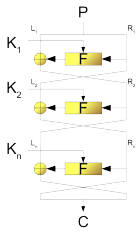
Kryp·to·gra·phie

(altgr. kryptós ‚verborgen', ‚geheim' und gráphein ‚schreiben')

Wissenschaft der Verschlüsselung von Informationen

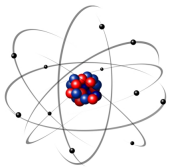
Überblick

1. Klassische symmetrische Kryptographie



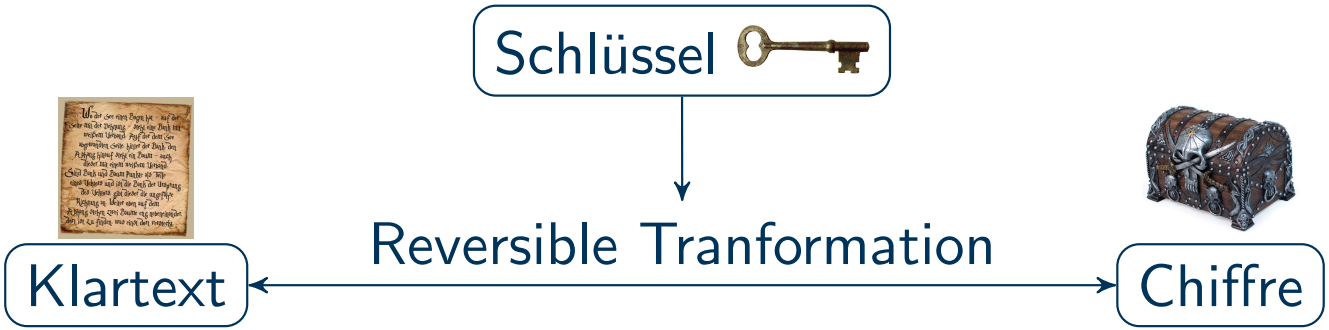
2. Blockchiffren, S-Boxen & die NSA

3. Clifford Cocks & die Public-Key Kryptographie



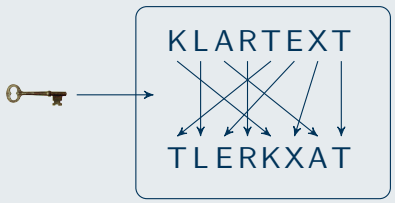
4. Peter Shor & Kryptographie mit Quantencomputern

Grundlagen Symmetrischer Kryptographie

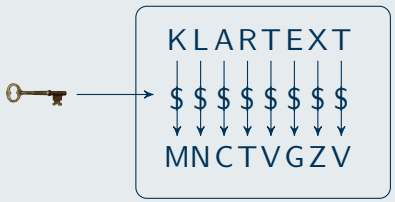


Typische Transformationen

- Permutation (Umordnung)
  - Schlüssel definiert Umordnung der Klartext-Symbole
  - Klassisches Beispiele: Skytale  404 v. Chr.



- Substitution (Ersetzung)
  - Schlüssel definiert Ersetzung der Klartext-Symbole

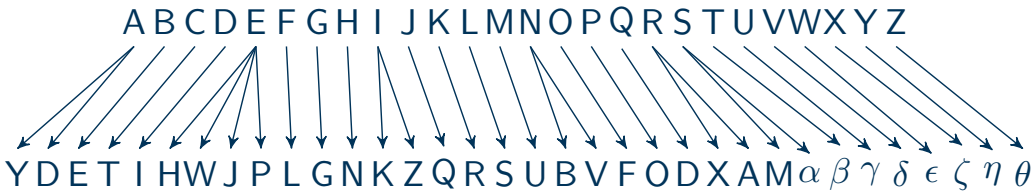


Substitutionschiffren und Alphabete

- Substitution entspricht *Umstellung des Alphabets*
- Beispiel: Caesar-Chiffre
  - Schlüssel ist 'Verschiebung' des Alphabets

  - *mono-alphabetischer* Ansatz: Problem Buchstabenhäufigkeiten

- *homophone* Substitution
  - Häufigkeitausgleich durch *Alphabeterweiterung*



  - Problem: Häufigkeit von Buchstabentupeln, bekannte Textpassagen

Polyalphabetische Substitution

Prinzip

Verwendung mehrerer Zielalphabete, abhängig z.B. von Schlüssel, Klartextposition und Klartext

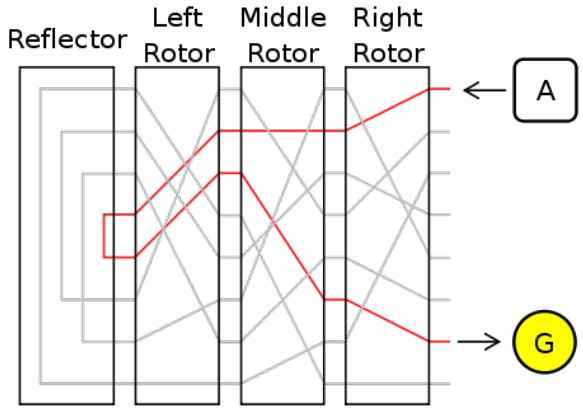
Beispiel: Vigenère-Verschlüsselung

- Schlüsselbuchstaben definieren Alphabetverschiebung (Caesar)
- Probleme:
  - zyklische Wiederholungen des Schlüssel (Abhilfe *Autokey*)
  - Buchstabenhäufigkeiten zu gleichem Schlüsselsymbol
  - wiederholt zusammenfallende Buchstabentupel

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

Polyalphabetische Substitution - Enigma

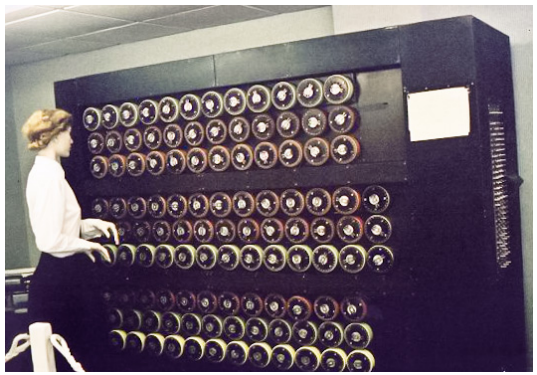
- Eingabe: Schreibmaschinentastatur
- Schlüssel: Steckbrett + Walzentyp, -position & -drehung
- Ausgabe: Lampenfeld

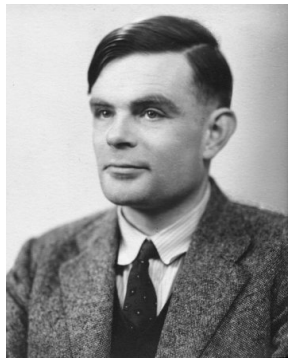


- jede Eingabe dreht Walzen weiter $\rightarrow$ neues Alphabet

Enigma - Kryptoanalyse

1932	Marian Rejewski (Polen) kauft & analysiert Enigma-Vorgänger
1939	Übergabe aller Informationen an britische Kryptographen
1940	Automatisierte Angriffe mittels *Turing-Bomben*





Angriff zweistufig

1. Analyse: Steckbretteinfluss entfernbar
   $\rightarrow$ mgl. Schlüsselanzahl schrumpft *drastisch*
2. Durchprobieren aller Walzenstellungen

Folgen: Tagesschlüssel noch vormittags bekannt

Perfekte Sicherheit

Idee

Für *jeden* Geheimtext ist unter Annahme zufällig erzeugter Schlüssel *jeder* Klartext gleich wahrscheinlich.

- Schlüssel 'durchprobieren' wird sinnlos
- Beispiel: One-Time-Pad mit XOR

Chiffre	Schlüssel	Klartext
010101	010101	000000
	010100	000001
	010111	000010
	010110	000011
	⋮	⋮

XOR

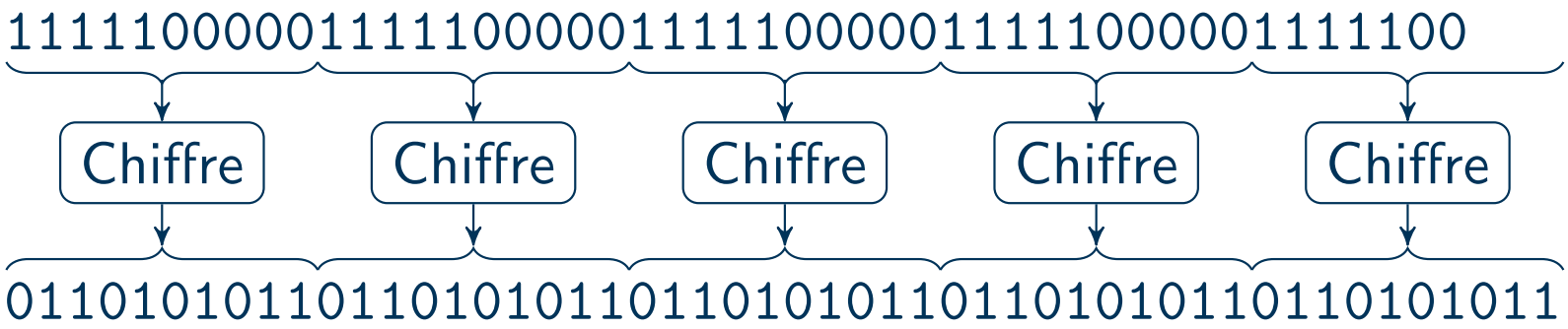
$\oplus$	0	1
0	0	1
1	1	0

- Voraussetzung: Schlüssellänge $\geq$ Textlänge
- Prob.: Wiederverwendung offenbart Klartextkomb.

$$(p_1 \oplus k) \oplus (p_2 \oplus k) = p_1 \oplus p_2$$

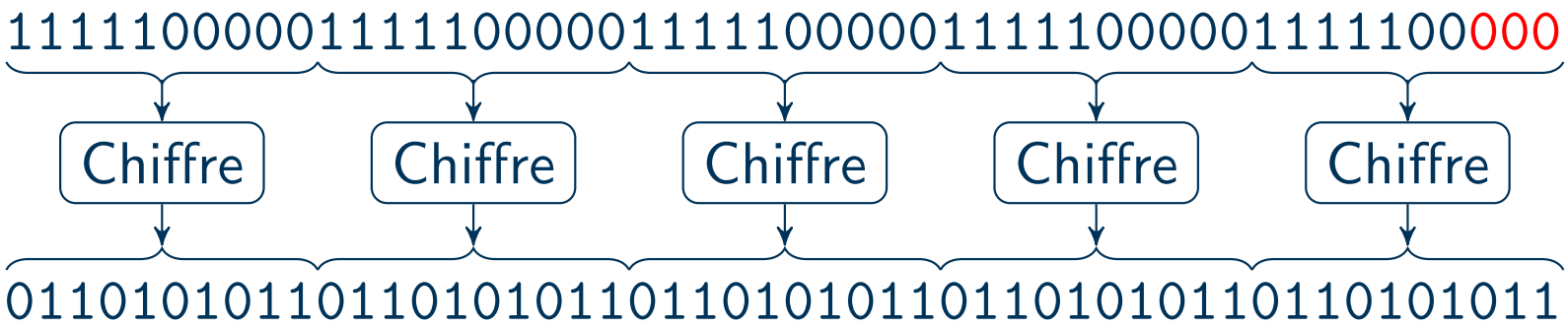
Blockchiffren

- Datenrepräsentation im Rechner: Binärstrings variabler Länge
- Idee: Aufteilung in Blöcke fester Länge



Blockchiffren

- Datenrepräsentation im Rechner: Binärstrings variabler Länge
- Idee: Aufteilung in Blöcke fester Länge

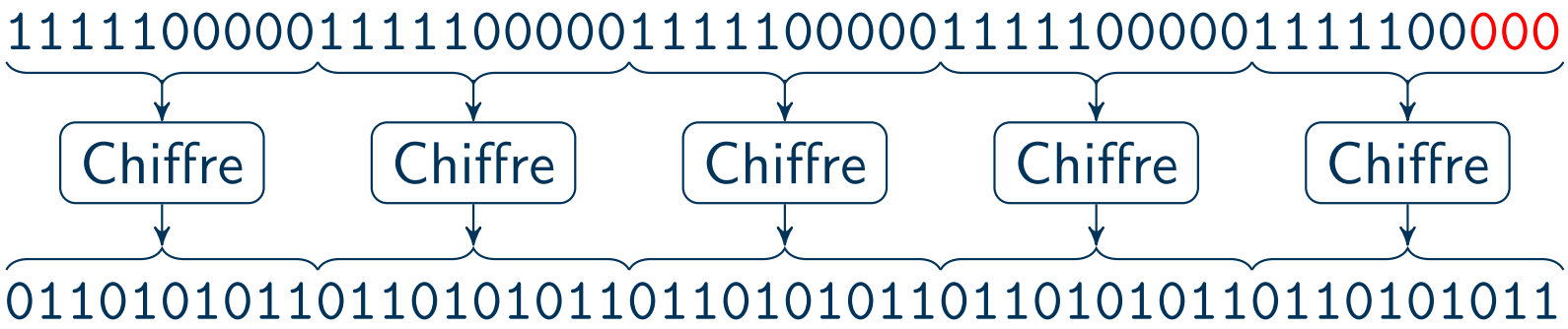


Verarbeitung von Bitstrings bedingt zus. Maßnahmen:

- "Auffüllen" (Padding) nötig

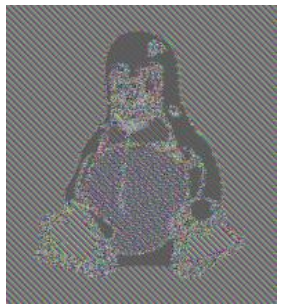
Blockchiffren

- Datenrepräsentation im Rechner: Binärstrings variabler Länge
- Idee: Aufteilung in Blöcke fester Länge



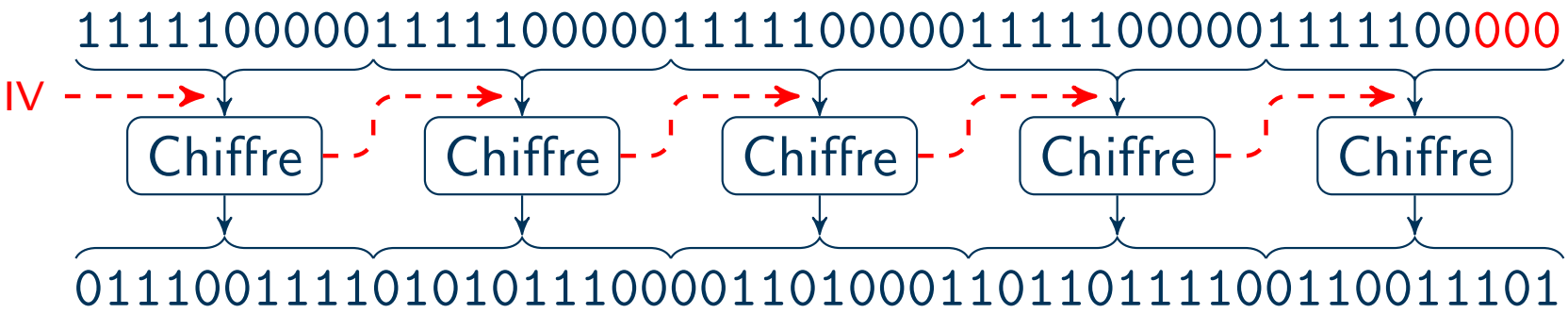
Verarbeitung von Bitstrings bedingt zus. Maßnahmen:

- "Auffüllen" (Padding) nötig
- Festlegung eines Blockmodus (ECB,CBC,CTR,...)



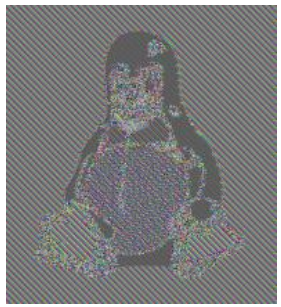
Blockchiffren

- Datenrepräsentation im Rechner: Binärstrings variabler Länge
- Idee: Aufteilung in Blöcke fester Länge



Verarbeitung von Bitstrings bedingt zus. Maßnahmen:

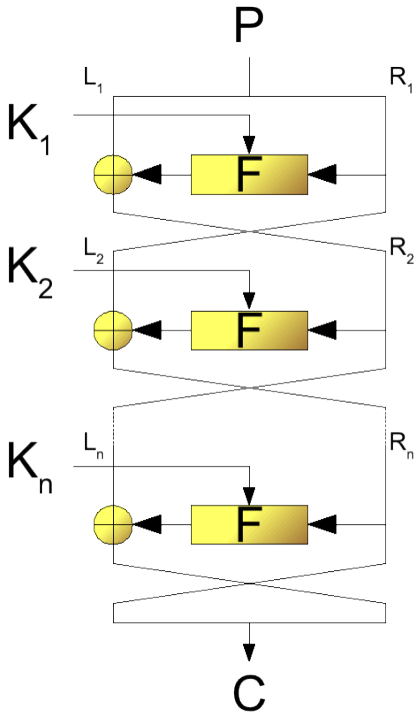
- "Auffüllen" (Padding) nötig
- Festlegung eines Blockmodus (ECB,CBC,CTR,...)



Blockchiffren

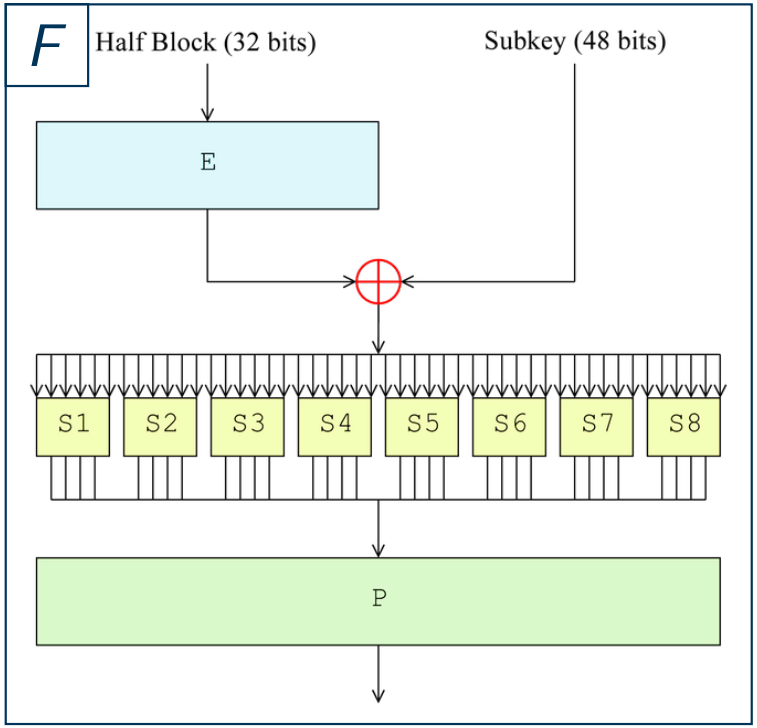
Feistel-Chiffre (1971, IBM-Projekt "Lucifer")

- generischer Blockchiffre
- rundenweiser Ablauf
- je Runde Permutations-, Substitutions- und Kombinationsschritt
- Eigenschaften abh. von Rundenfunktion F unter Einfluss von Rundenschlüssel
- Entschl. ist Verschl. mit vert. Rundenschlüsseln
- Prototyp vieler moderner Blockchiffren: DES, Tiple-DES, Blowfish, Twofish



Beispiel: Data Encryption Standard (DES) (1975)

- 16 Runden Feistel-Chiffre mit Permutation zu Beginn und Ende
- F enthält fixe Expansion, Substitution (S-Boxen) und Permutation
- NSA-Einfluss auf Standardisierung:
  - 56-Bit Schlüssel
  - S-Boxen stabil gegen differentielle Kryptoanalyse (1994)
- abgelöst durch: Triple-DES, AES

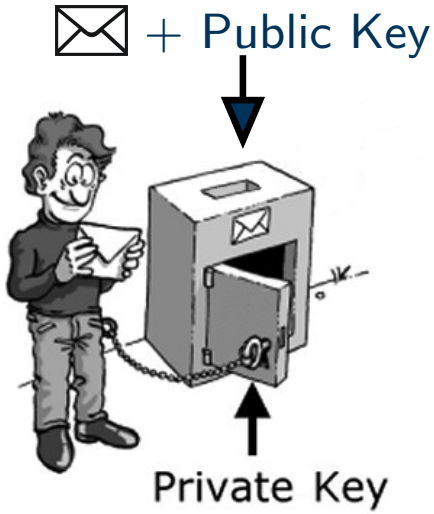


111111

S1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
111111 3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

1101

Public-Key Kryptographie / Asymmetrische Kryptographie



- kein paarweiser Austausch *geheimer* Schlüssel
- Schlüssel *identitätsgebunden*
- öffentlicher Schlüssel zum Verschlüsseln,
  privater Schlüssel zum Entschlüsseln

Verwendung scheinbarer Einwegfunktion

- Verschlüsselung ist einfache Tranformation
- ohne privaten Schlüssel entspricht Entschlüsselung
  notorisch schwerem mathematischem Problem
- privater Schlüssel zeigt Hintertür

'Schwere' Probleme

Informatiker teilen Probleme in Klassen ein.

Geeignet sind *NP-schwere* Probleme, aber auch:

Faktorisierung

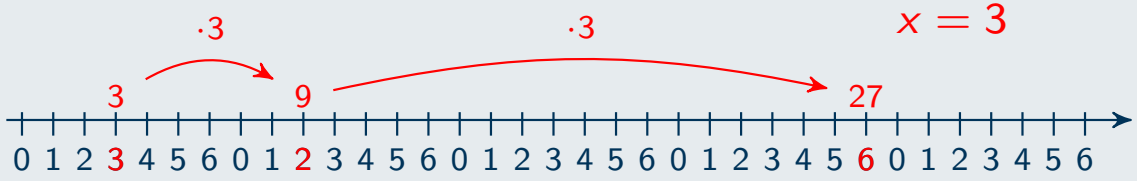
Für $n \in \mathbb{N}$ finde Primzahlen $p_1, \ldots, p_z$ mit $n = p_1 \cdot \ldots \cdot p_z$.

Bsp.: $140 = 7 \cdot 5 \cdot 2 \cdot 2$

Diskreter Logarithmus

$3^x \equiv 6 \mod 7$

$x = 3$



Für $g, n, t \in \mathbb{N}$ finde x mit $g^x \equiv t \mod n$.

RSA

Das RSA-Verfahren

- Wähle $p, q \in \mathbb{N}$ prim, groß, mit Abstand
- Ergebnis $n = p \cdot q$ und $\phi(n) = (p-1) \cdot (q-1)$
- Wähle e mit $\mathrm{ggT}(e, \phi(n)) = 1$, berechne d mit $e \cdot d \equiv 1 \mod \phi(n)$.

- Public Key (e, n), Private Key (d, n)
- Verschl. $c \equiv m^e \mod n$, Entschl. $m \equiv c^d \mod n$

- Clifford Cocks (GCHQ) 1973
  (1997 deklassifiziert)
- Rivest, Shamir, Adleman (MIT) 1977
  später *RSA Security Inc.*, Wert 2.1 Mrd. $

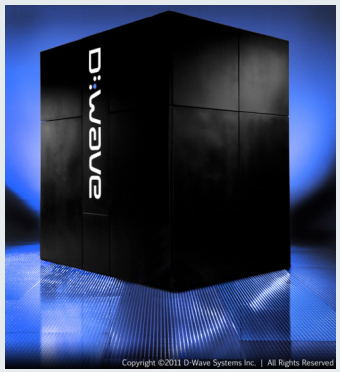
Sicherheit basiert auf Schwierigkeit von *Faktorisierung* und *diskretem Logarithmus*.

Neue Möglichkeiten durch Quantencomputer (1)

Quantencomputer

- Q-Bits speichern 0 und 1 je mit bestimmter Wahrscheinlichkeit
- Traum: Verarbeitung aller $2^{\text{Anzahl Q-Bits}}$ Registerzustände gleichzeitig (allerdings nicht erreichbar)
- verschiedene physische Realisierung

D-Wave



- 512 Q-Bits (*nicht universell einsetzbar*)
- ab 10.000.000$
- Stickstoff-Kühlung, 10m $\times$ 10m-Grundfläche
- Kunden: Google, NASA, Lockheed Martin

"Existenz funkt. Quantencomputers kann vorausgesetzt werden"

Neue Möglichkeiten durch Quantencomputer (2)

BigBangTheory S01E13

`Gablehouser:` How does a quantum computer factor large numbers. *(Buzz)* PMS?

`Leslie Winkle:` Shor's Algorithm.

Shors Algorithmus

- *effizienter* Faktorisierungsalgorithmus
- Entwurf 1995 für (damals hypothetische) Quantencomputer
- 2012: erfolgreiche Faktorisierung von $21 = 3 \cdot 7$

Plötzlich werden Faktorisierung und verwandte Probleme *effizient* lösbar.

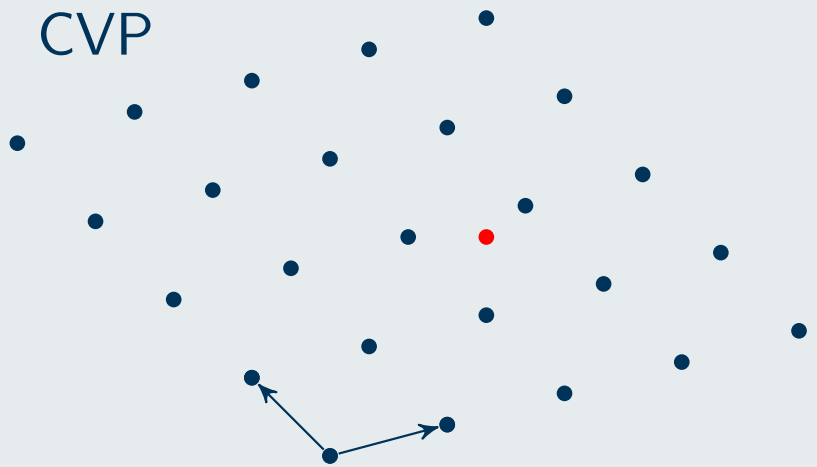
Aber wir haben RSA und viele andere Schemata darauf aufgebaut?!

Was nun?

Kryptographie im Zeitalter von Quantencomputern

- NP-schwere Probleme bleiben schwer (falls P≠NP)
- Umstellung auf Konstruktionen aufbauend auf NP-schweren Problemen (erstaunlich schwierig)
- heiße Kandidaten:

Verbandsbasierte (lattice-based) Krypto

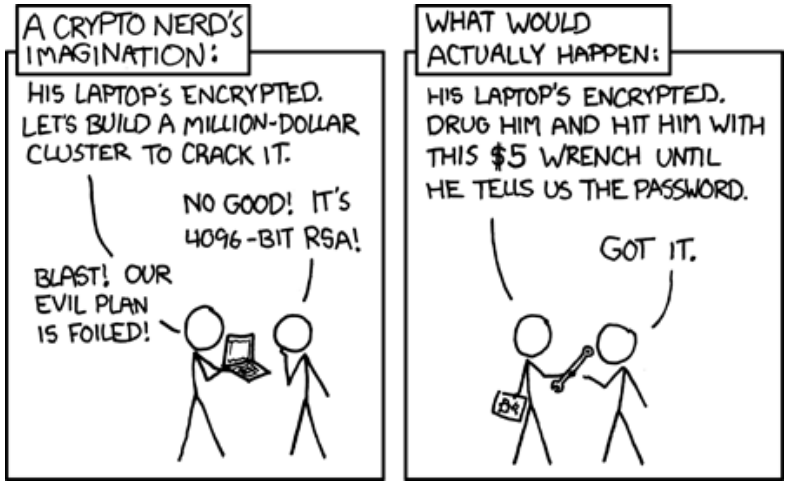


Ein paar Worte zum Schluss

- Zugangsmgl. der NSA liegen nicht an krypt. Verfahren
- Schwächung der Verschlüsselung durch Fortschritte bei Rechentechnik moderat
- Eigentliche Probleme liegen in der Infrastruktur
  - Hintertüren und Programmfehler in Software-Implementierungen
  - Weiternutzung bekannt unsicherer Verfahren (häufig kein DH im SSL)
  - Schlechte Zufallszahlen (Android, Debians OpenSSL)
  - Zu viele, zu unsichere und zu marktorientierte CAs

$(g^a)^b \mod p$

DigiNotar
Internet Trust Services



Es bleibt spannend!