Einführung in die Kryptologie

Dr. Patryk Brzezinski   Dr. Alexander Ullmann

Diskrete Mathematik 2 (I168)

3. Quartal 2024

NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

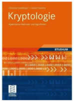
Überblick

Überblick

Quellen/Literaturempfehlungen

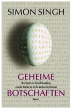
- Folien basieren auf dem Skript von Uwe Neuhaus
- Beutelspacher/Neumann/Schwarzpaul: „Kryptografie in Theorie und Praxis“, 2. Aufl., Vieweg + Teubner, 2010.

- Karpfinger/Kiechle: „Kryptologie“, Vieweg + Teubner, 2010.



- Beutelspacher: „Kryptologie: Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen“, 9. Aufl., Vieweg + Teubner, 2009.

- Singh: „Geheime Botschaften“, dtv, 2001.



Überblick

Begriff: Kryptografie

- $\kappa\rho\upsilon\pi\tau\acute{o}\varsigma$ (kryptós): verborgen, geheim
- $\gamma\rho\acute{\alpha}\phi\varepsilon\iota\nu$ (gráphein): schreiben
- Wörtlich: Die Lehre vom geheimen Schreiben
- Traditionelles Verständnis:
  Die Lehre vom Verschlüsseln von Informationen zu deren Geheimhaltung
- Modernes Verständnis:
  Einbeziehung weiterer Aspekte (siehe „Schutzziele“)

Begriff: Kryptoanalyse

- $\alpha\nu\alpha\lambda\acute{\upsilon}\sigma\varepsilon\iota\nu$ (analysein): Auflösen (in Bestandteile)
- Bedeutung:
  Methoden und Techniken, um Informationen aus verschlüsselten Texten zu gewinnen
  (verschlüsselte Texte, verwendete Schlüssel, eingesetzte Verfahren)
- Ziele:
  - $\rightarrow$ Brechen der verwendeten kryptografischen Verfahren
  - $\rightarrow$ Nachweis/Quantifizierung der Sicherheit von Verfahren

Begriff: Kryptologie

- $\lambda\acute{o}\gamma o\varsigma$ (lógos): Wort, Lehre, Sinn
- Bedeutung:
  Wissenschaft von der Entwicklung und Analyse von Kryptosystemen

Kryptologie = Kryptografie + Kryptoanalyse

- Kryptologie selbst ist ein Teilgebiet der Informationssicherheit

Begriff: Schutzziele

	Brief	Einschreiben mit Rückschein	Flaggen-alphabet
(a) Vertraulichkeit: Nachricht für Unbefugte nicht lesbar	nein	nein	nein
(b) Integrität: Nachricht nicht unbemerkt veränderbar	nein	nein	ja
(c) Authentizität: Nachricht stammt tatsächlich von der angegebenen Quelle	nein	ja	ja
(d) Nichtabstreitbarkeit/Verbindlichkeit: Versand/Empfang der Nachricht unleugbar	nein	ja (?)	ja/nein
(e) Anonymität: Absender/Empfänger/Kommunikationsbeziehung bleibt geheim	nein	nein	nein

Kryptosystem

- Seien P, C und K nichtleere Mengen:
  - → P: Menge der Klartexte (*plain text*)
  - → C: Menge der Geheimtexte (*cipher text*)
  - → K: Menge der Schlüssel (*keys*)
- Seien e und d Abbildungen:
  - → e: Verschlüsselungsfunktion (*encrypt*)

$$e : P \times K \rightarrow C$$

  - → d: Entschlüsselungsfunktion (*decrypt*)

$$d : C \times K \rightarrow P$$

Kryptosystem

Das Quintupel (P, C, K, e, d) heißt ein kryptografisches System (kurz: Kryptosystem), falls gilt:

$$\forall k \in K\ \exists k' \in K\ \forall x \in P : d(e(x, k), k') = x \tag{1}$$

Andere Namen für Kryptosysteme:

- Verschlüsselungsverfahren
- Chiffre
- Cipher

Symmetrische Kryptosysteme

- k' (Schlüssel zum Entschlüsseln) aus
  k (Schlüssel zum Verschlüsseln) leicht ermittelbar
- Spezialfall: k' und k sind identisch
- Andere Formulierungen:
  - $\rightarrow$ Sender und Empfänger besitzen dasselbe Geheimnis
  - $\rightarrow$ Wer verschlüsseln kann, kann auch entschlüsseln

Asymmetrische Kryptosysteme

- k' (Schlüssel zum Entschlüsseln) aus
  k (Schlüssel zum Verschlüsseln) nicht in angemessener Zeit ermittelbar
- Alternativer Name: public-key-Kryptosysteme

  $\rightarrow$ k kann vom Empfänger öffentlich bekannt gegeben werden (k' hält er aber geheim)

  $\rightarrow$ Sender und Empfänger müssen kein Geheimnis austauschen

  $\rightarrow$ Wer verschlüsseln kann, kann nicht entschlüsseln

Kerckhoffs'sches Prinzip (1883)

Die Sicherheit eines Verschlüsselungssystems darf nur

von der Geheimhaltung des Schlüssels abhängen,

nicht von der Geheimhaltung des Verschlüsselungsverfahrens!

August Kerckhoffs

Gründe für öffentliche Verschlüsselungsverfahren

- Geheimhaltung: Verschlüsselungsfunktionen schwer geheim zu halten
- Reverse-Engineering: Verfahren aus Hard- oder Software-Implementierungen rekonstruierbar
- Ersetzbarkeit: Schlüssel leichter zu ersetzen als Verschlüsselungsverfahren
- Öffentlichkeit: Fehler leichter zu entdecken
- Vertrauen: Hintertüren in nicht öffentlichen Verfahren möglich
- Erfahrung: geheime Algorithmen erwiesen sich oft als unzulänglich

Angriffsszenarien

- Ciphertext Only: Angreifer kennt einen/mehrere Geheimtexte
- Known/Probable Plaintext: Angreifer kennt einen/mehrere Geheimtexte und zugehörige Klartexte/wahrscheinliche Klartextteile
- Chosen Plaintext: Angreifer kann ausgewählte Klartexte verschlüsseln lassen
- Chosen Ciphertext: Angreifer kann ausgewählte Geheimtexte entschlüsseln lassen
- Chosen Text: Chosen Plaintext + Chosen Ciphertext
- Adaptive Varianten: Angriff über längere Zeit/mehrere Runden möglich

Überblick

Caesar-Chiffre (1. Jhd. vor Chr.)

- Ersetzung der Klar- durch Geheimtextbuchstaben
- Abbildung durch zyklische Rechtsverschiebung

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZABC
```

Verschiebung +3

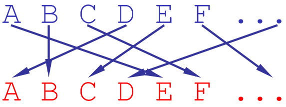
- Zahl der verschobenen Zeichen bildet den Schlüssel
- Entschlüsselung kehrt die Verschiebung um
- Monoalphabetisches Substitutionsverfahren:
  Abbildung von Klar- auf Geheimtextbuchstaben aufgrund eines einzigen Geheimtextalphabets

Jeder Klartextbuchstabe wird immer durch den selben Geheimtextbuchstaben ersetzt

Caesar-Chiffre mathematisch

- Zuordnung der Klartextbuchstaben zu den Restklassen $\mathbb{Z}_{26}$:
  A = 0, B = 1, C = 2, . . . , Z = 25
- Es ergibt sich das folgende Kryptosystem:
  - → $\mathbb{Z}_{26}$ = Klartextalphabet
    = Geheimtextalphabet
    = K (Schlüsselmenge)
  - → $(\mathbb{Z}_{26})^*$ = P (Menge der Klartexte)
    = C (Menge der Geheimtexte)
  - → Für alle $x = x_0x_1 \dots x_{n-1} \in P$ und $k \in K$ gilt
    $e(x, k) = y = y_0y_1 \dots y_{n-1} \in C$ mit $y_i = (x_i + k) \mod 26$;
    $i = 0, \dots, n - 1$
  - → Für alle $y = y_0y_1 \dots y_{n-1} \in C$ und $k \in K$ gilt
    $d(y, k) = x = x_0x_1 \dots x_{n-1} \in P$ mit $x_i = (y_i - k) \mod 26$;
    $i = 0, \dots, n - 1$

Permutations-Chiffren

- Ceasar-Chiffren: nur 26 spezielle Permutationen
- Permutations-Chiffren: erlauben alle möglichen Permutationen
- 

- Schlüssel: Angabe der vollständigen Permutation
- Entschlüsselung: Ersetzung gemäß inverser Permutation
- Das Verfahren Permutations-Chiffren ist ebenfalls ein monoalphabetisches Substitutionsverfahren

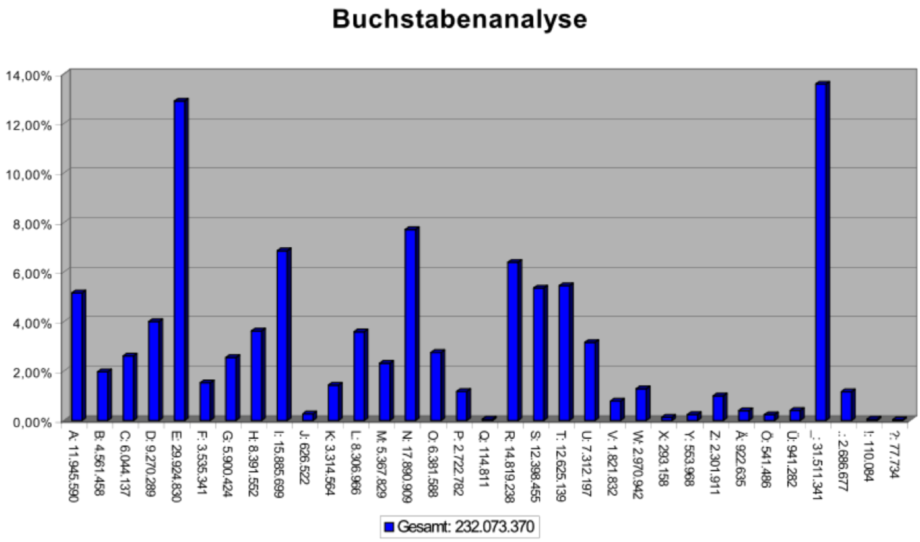
Permutations-Chiffren mathematisch

- $\Gamma = \{A, B, \ldots, Z\}$ = Klartextalphabet
  = Geheimtextalphabet
- $\{A, B, \ldots, Z\}^*$ = P (Menge der Klartexte)
  = C (Menge der Geheimtexte)
- Schlüsselmenge $K = S(\Gamma) = \{k : \Gamma \to \Gamma | k \text{ ist bijektiv}\}$ die Menge aller Permutationen von Γ
  - → Für alle $x = x_0 x_1 \ldots x_{n-1} \in P$ und $k \in K$ gilt
    $e(x, k) = y = y_0 y_1 \ldots y_{n-1} \in C$ mit $y_i = k(x_i); i = 0, \ldots, n-1$
  - → Für alle $y = y_0 y_1 \ldots y_{n-1} \in C$ und $k \in K$ gilt
    $d(y, k) = x = x_0 x_1 \ldots x_{n-1} \in P$ mit $x_i = k^{-1}(y_i); i = 0, \ldots, n-1$

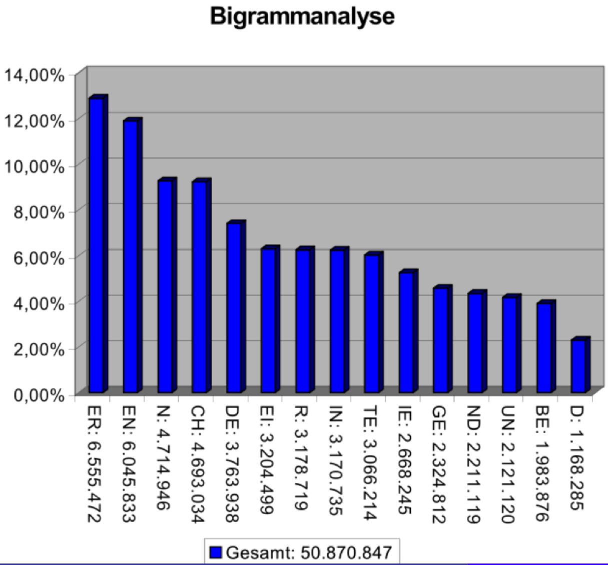
Häufigkeitsanalyse

- Das Verfahren Permutations-Chiffren besitzt eine sehr große Schlüsselmenge: $|S(\Gamma)| = |\Gamma|!$
- Beispiel:
  - $\rightarrow$ $\Gamma = \{\text{A}, \text{B}, \ldots, \text{Z}\}$, also $|\Gamma| = 26$
  - $\rightarrow$ $|K| = 26! = 403291461126605635584000000 \approx 4 \cdot 10^{26}$
- Aber:
  monoalphabetische Substitutionen leicht angreifbar, denn sprachspezifische Besonderheiten bleiben erhalten:
  - $\rightarrow$ Häufigkeitsverteilung der Buchstaben
  - $\rightarrow$ Häufigkeitsverteilung von Buchstabenkombinationen (z.B. Bigramme, Trigramme)
- Damit anfällig gegen Ciphertext-Only-Angriffe

Beispiel: Buchstabenhäufigkeit in deutschen Texten



Beispiel: Bigrammhäufigkeit in deutschen Texten



Vigenère-Chiffre

- Gleichzeitige Verwendung mehrerer Caesar-Chiffren
  - Ersetzung einzelner Klartextbuchstaben gemäß Caesar-Chiffre
  - Position des Klartextbuchstaben bestimmt gewählte Caesar-Chiffre
  - Gewähltes Schlüsselwort bestimmt Abfolge der genutzten Caesar-Chiffren
- Polyalphabetisches Substitutionsverfahren:
  Abbildung von Klar- auf Geheimtextbuchstaben mit Hilfe mehrerer Geheimtextalphabete

Jeder Klartextbuchstabe kann durch unterschiedliche Geheimtextbuchstaben ersetzt werden

Vigenère-Chiffre: Vorgehen

- Wähle ein Schlüsselwort (z.B. `GEHEIM`)
- Wiederhole das Schlüsselwort, bis die Länge des Klartextes erreicht ist
- Wähle für jeden Klartextbuchstaben die Caesar-Chiffre, die dem entsprechendem Schlüsselwortbuchstaben entspricht ($\rightarrow$ *Vigenère*-Quadrat)

Klartext:	`TREFFENUMDREI`
Schlüsselwortreihe:	`GEHEIMGEHEIMG`
Geheimtext:	`ZVLJNQTYTHZQO`

- Entwickelt von Giovan Battista Bellaso
- Benannt nach Blaise de *Vigenère*

Giovan Battista Bellaso (1553)

Blaise de *Vigenère* (1586)

Vigenère-Quadrat

Text

Schlüssel

	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

Vigenère-Chiffre mathematisch

- Zuordnung der Klartextbuchstaben zu den Restklassen $\mathbb{Z}_{26}$:
  A = 0, B = 1, C = 2, . . . , Z = 25
- Es ergibt sich das folgende Kryptosystem
  - → $\mathbb{Z}_{26}$ = Klartextalphabet
    = Geheimtextalphabet
  - → $(\mathbb{Z}_{26})^*$ = P (Menge der Klartexte)
    = C (Menge der Geheimtexte)
    = K (Menge der Schlüssel)
  - → Für alle $x = x_0 x_1 \dots x_{n-1} \in P$ und $k = k_0 k_1 \dots k_{m-1} \in K$ gilt
    $e(x,k) = y = y_0 y_1 \dots y_{n-1} \in C$ mit $y_i = (x_i + k_{i \bmod m}) \mod 26$; $i = 0, \dots, n-1$
  - → Für alle $y = y_0 y_1 \dots y_{n-1} \in C$ und $k \in K$ gilt
    $d(y,k) = x = x_0 x_1 \dots x_{n-1} \in P$ mit $x_i = (y_i - k_{i \bmod m}) \mod 26$; $i = 0, \dots, n-1$

Sicherheit der *Vigenère*-Chiffre

- Als polyalphabetisches Substitutionsverfahren deutlich sicherer als monoalphabetische Verfahren
- Sicherheit hängt wesentlich vom Schlüsselwort ab

  → Länge des Schlüsselwortes
  → Zufälligkeit der Buchstaben

- Schwachpunkt: Wiederholung des Schlüssels

  → Anfällig gegen Known-Plaintext-Angriffe
  → Schlüssellänge bekannt → Reduktion auf unsichere Caesar-Chiffren
  → Bekannte kryptoanalytische Methoden:

    - Methode von Charles Babbage (1854)/Kasiski-Test (1863)
    - Friedman-Test (1920)

  → Damit auch anfällig gegen Ciphertext-Only-Angriffe

Der Kasiski-Test

- Idee:
  Wiederholt sich eine Zeichenfolge im Klartext mit einem Abstand, der ein Vielfaches der Länge des Schlüsselworts ist, so werden diese Zeichenfolgen gleich verschlüsselt.
- Daraus resultieren Wiederholungen von Zeichenfolgen im Geheimtext, deren Abstand ein Vielfaches der Länge des Schlüsselworts ist.
- Aber:
  Wiederholungen können auch zufällig auftreten.

Der Kasiski-Test

- Schlüssel: `PLUTO`
- Klartext 1: der klartext wird zum geheimtext

```
PLUTOPLUTOPLUTOPLUTOPLUTOPLU
derklartextwirdzumgeheimtext
SPLDZPCNXLIHCKROFGZSWPCFHTIN
```

- Klartext 2: der klartext werde geheimtext

```
PLUTOPLUTOPLUTOPLUTOPLUTOP
derklartextwerdegeheimtext
SPLDZPCNXLIHYKRTRYASXXNXLI
```

(entnommen von wikipedia.de)

Der Kasiski-Test: Ein Beispiel

- Mit *Vigenère*-Chiffrierung verschlüsselter deutschsprachiger Text (die Aufteilung in Blöcke dient nur der besseren Übersicht):
- `FSGEXV EVIISA MGYFNX EJTMUR MPNYME FMPSIH`
  `EFIXUE HQFOOU PGIAVI KJSWLT IIZJIJ ELXVOT`
  `YBKMEC GYUELW RHEHOR ONIFVS EHKCJS WLFEEL`
  `JIBNTS VTIMGY JSNECT IBRQVE HXJDHF YVTSYP`
  `EEIYWX JLNRRU UYVCJC BELDHZ YVSKFE IUERXV`
  `TGCFKF IHIZOF UGYFSP IIGABV VOGYRL FGYRUL`
  `AHKVOK FEIUER XRVFTY XFHIII JGEIZU ZOIZOE`
  `LFVLAH RKFNMT IBCBIQ VUHXVS SOGOFN ILEFSY`
  `MEFSSR KBXORU TEGEEU IEDLCE NVRDHN IE`
- Zeichenfolge `MGY` tritt doppelt auf

Der Kasiski-Test: Ein Beispiel

- Mit *Vigenère*-Chiffrierung verschlüsselter deutschsprachiger Text (die Aufteilung in Blöcke dient nur der besseren Übersicht):
- ```
  FSGEXV EVIISA MGYFNX EJTMUR MPNYME FMPSIH
  EFIXUE HQFOOU PGIAVI KJSWLT IIZJIJ ELXVOT
  YBKMEC GYUELW RHEHOR ONIFVS EHKCJS WLFEEL
  JIBNTS VTIMGY JSNECT IBRQVE HXJDHF YVTSYP
  EEIYWX JLNRRU UYVCJC BELDHZ YVSKFE IUERXV
  TGCFKF IHIZOF UGYFSP IIGABV VOGYRL FGYRUL
  AHKVOK FEIUER XRVFTY XFHIII JGEIZU ZOIZOE
  LFVLAH RKFNMT IBCBIQ VUHXVS SOGOFN ILEFSY
  MEFSSR KBXORU TEGEEU IEDLCE NVRDHN IE
  ```
- `MGY` an 13. und 118. Position, Abstand $105 = 3 \cdot 5 \cdot 7$
  ```

Der Kasiski-Test: Ein Beispiel

- Mit *Vigenère*-Chiffrierung verschlüsselter deutschsprachiger Text (die Aufteilung in Blöcke dient nur der besseren Übersicht):
- FSGEXV EVIISA MGYFNX EJTMUR MPNYME FMPSIH
  EFIXUE HQFOOU PGIAVI KJSWLT IIZJIJ ELXVOT
  YBKMEC GYUELW RHEHOR ONIFVS EHKCJS WLFEEL
  JIBNTS VTIMGY JSNECT IBRQVE HXJDHF YVTSYP
  EEIYWX JLNRRU UYVCJC BELDHZ YVSKFE IUERXV
  TGCFKF IHIZOF UGYFSP IIGABV VOGYRL FGYRUL
  AHKVOK FEIUER XRVFTY XFHIII JGEIZU ZOIZOE
  LFVLAH RKFNMT IBCBIQ VUHXVS SOGOFN ILEFSY
  MEFSSR KBXORU TEGEEU IEDLCE NVRDHN IE
- GYF an 14. und 194. Position, Abstand $180 = 2^2 \cdot 3^2 \cdot 5$

Der Kasiski-Test: Ein Beispiel

Zeichenfolge	Position 1	Position 2	Abstand
MGY	13	118	$105 = 3 \cdot 5 \cdot 7$
GYF	14	194	$180 = 2^2 \cdot 3^2 \cdot 5$
YMEF	28	288	$260 = 2^5 \cdot 5 \cdot 13$
JSWL	56	101	$45 = 3^2 \cdot 5$
TIB	126	264	$138 = 2 \cdot 3 \cdot 23$
KFEIUERX	172	222	$50 = 2 \cdot 5^2$
IZO	189	249	$60 = 2^2 \cdot 3 \cdot 5$
GYR	207	212	5
LAH	216	256	$40 = 2^3 \cdot 5$
EFS	280	285	5

$\rightarrow$ Schlüssellänge 5

Perfekte Sicherheit - Ununterscheidbarkeit

Sicher im Sinne der Ununterscheidbarkeit

- Ein Angreifer wählt zwei beliebige Klartexte (gleicher Länge)
- Einer der beiden Texte wird (verborgen vor dem Angreifer) zufällig gewählt und verschlüsselt
- Der Angreifer erhält den Geheimtext und soll entscheiden, zu welchem der beiden Klartexte er gehört
- Ist die Wahrscheinlichkeit für eine richtige Wahl $\frac{1}{2}$, so gilt das Kryptosystem als perfekt sicher

Perfekte Sicherheit - Semantische Sicherheit

- Ein Angreifer weiß, dass eine verschlüsselte Nachricht m der Länge n übertragen wurde
  - $\rightarrow$ A priori Wahrscheinlichkeit
    Wahrscheinlichkeit, dass ein bestimmter Klartext übertragen wurde
  - $\rightarrow$ A posteriori Wahrscheinlichkeit
    Wahrscheinlichkeit, dass ein bestimmter Klartext übertragen wurde, wenn zusätzlich der Geheimtext bekannt ist
- Kryptosystem perfekt sicher:

  A priori W. = a posteriori W. für beliebige Klartexte und bei beliebigem Geheimtext

Perfekt sicheres Kryptosystem: Das One-Time-Pad

- 1918 entwickelt von Gilbert Vernam
- Basiert auf *Vigenère*-Chiffre, aber:
  - → Schlüssellänge = Klartextlänge
  - → Schlüsselbuchstaben zufällig (gemäß Gleichverteilung) gewählt
  - → Schlüssel dürfen nur einmal verwendet werden (one time)
- Deutsche Umschreibung: Einmalblock
- Herausforderungen (klassisch)
  - → Schlüsselgröße
  - → Schlüsselverteilung
- Mögliche Lösung: Quantenkryptographie

Gilbert Vernam

Überblick

Erinnerung: Eulersche Phi-Funktion

- $\varphi(n) := \left|\left\{a \in \mathbb{N} \mid 1 \leq a \leq n \wedge \mathrm{ggT}(a, n) = 1\right\}\right|$
- In Worten:

  $\varphi(n)$ ist die Anzahl der zu n teilerfremden natürlichen Zahlen zwischen 1 und n
- Beispiele
  - $\rightarrow\ \varphi(1) = 1$
  - $\rightarrow\ \varphi(2) = 1$
  - $\rightarrow\ \varphi(3) = 2$
  - $\rightarrow\ \varphi(7) = 6$
  - $\rightarrow\ \varphi(12) = 4$
  - $\rightarrow\ \varphi(26) = 12$

Erinnerung: Berechnung der Eulerschen Phi-Funktion

(1) Sei $p \in \mathbb{P}$. Dann gilt:

$$\varphi(p) = p - 1.$$

(2) Seien $p \in \mathbb{P}$ und $k \in \mathbb{N}$. Dann gilt:

$$\varphi(p^k) = p^k - p^{k-1} = p^k \left(1 - \frac{1}{p}\right).$$

(3) Seien $m, n \in \mathbb{N}$ und $\mathrm{ggT}(m, n) = 1$. Dann gilt:

$$\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n).$$

(4) Spezialfall: Seien $p, q \in \mathbb{P}$ mit $p \neq q$. Dann gilt:

$$\varphi(p \cdot q) = (p - 1) \cdot (q - 1).$$

(5) Sei $1 \neq n = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_r^{k_r} = \prod_{j=1}^{r} p_j^{k_j}$ mit $r \in \mathbb{N}$, paarweise verschiedenen $p_j \in \mathbb{P}$ und $k_j \in \mathbb{N}$ für alle $j = 1, \ldots, r$. Dann gilt:

$$\varphi(n) = n \cdot \prod_{j=1}^{r} \left(1 - \frac{1}{p_j}\right).$$

Erinnerung: Sätze von Euler und Fermat

- Satz von Euler (1760)
  Seien $a, n \in \mathbb{N}$ und sei $\text{ggT}(a, n) = 1$.
  Dann gilt: $a^{\varphi(n)} \equiv_n 1$.
  Alternative Formulierung: $a^{\varphi(n)} \bmod n = 1$.

Leonhard Euler

- Satz von Fermat (1637)
  Seien p eine Primzahl und $a \in \mathbb{N}$ mit
  $1 \leq a \leq p - 1$. Dann gilt: $a^{p-1} \equiv_p 1$.
  Alternative Formulierung: $a^{p-1} \bmod p = 1$.

Pierre de Fermat

Überblick

Motivation für asymmetrische Kryptosysteme

Nachteile bei symmetrischen Kryptosystemen:

- Voriger Austausch geheimer Schlüssel notwendig
- Stark steigende Schlüsselzahl bei vielen Kommunikationsteilnehmern:

$$n \text{ Teilnehmer} \longrightarrow \frac{n(n-1)}{2} \text{ Schlüssel}$$

- Keine Unterstützung von digitalen Signaturen

RSA im Überblick

- Verbreitetes asymmetrisches Kryptosystem
- Entwickelt 1977 von

  → Ronald Rivest
  → Adi Shamir
  → Leonard Adleman

- Verwendung zur Verschlüsselung und zur digitalen Signatur
- Jeder Teilnehmer benötigt zwei Schlüssel

  → Privater Schlüssel:
  Entschlüsseln (und Signieren der Nachricht)

  → Öffentlicher Schlüssel:
  Verschlüsseln (und Prüfen der Signatur)

RSA: Konstruktion des öffentlichen Schlüssels

- Wähle zwei (sehr, sehr) große Primzahlen p und q mit $p \neq q$.
- Setze $n := p \cdot q$.
- Berechne $\varphi(n) = (p-1) \cdot (q-1)$.
- Wähle eine zu $\varphi(n)$ teilerfremde Zahl e für die gilt $1 < e < \varphi(n)$.
- Das Zweitupel (n, e) ist der öffentliche Schlüssel, er wird allen bekanntgegeben.
- Die zur Berechnung benötigten Werte p, q und $\varphi(n)$ werden nicht veröffentlicht.

Beispiel

Wähle $p := 7$ und $q := 11$. Dann ist

$$n = p \cdot q = 7 \cdot 11 = 77 \qquad \text{und}$$
$$\varphi(n) = \varphi(p \cdot q) = (p-1) \cdot (q-1) = 6 \cdot 10 = 60.$$

Wähle $e := 13$.

Öffentlicher Schlüssel: $(n, e) = (77, 13)$.

RSA: Konstruktion des privaten Schlüssels

- Berechne $1 < d < \varphi(n)$ als das multiplikative Inverse zu e in $\mathbb{Z}_{\varphi(n)}$.
  - $\rightarrow$ Es muss also gelten: $e \cdot d \equiv_{\varphi(n)} 1$
  - $\rightarrow$ Verwende zur Berechnung den erweiterten euklidischen Algorithmus.
- Das Zweitupel (n, d) ist der private Schlüssel.
- Der private Schlüssel wird stets geheimgehalten und niemandem mitgeteilt.
- Die Werte p, q und $\varphi(n)$ können nach der Berechnung des privaten Schlüssels gelöscht werden.

Weiter im Beispiel von Folie 40

Multiplikatives Inverse zu $[13]_{60}$ mit erweitertem Euklidischen Algorithmus berechnen:

a	b	q	s	t
60	13	4	5	-23
13	8	1	-3	5
8	5	1	2	-3
5	3	1	-1	2
3	2	1	1	-1
2	1	2	0	1
1	0		1	0

Es ergibt sich:

$$[13]_{60}^{-1} = [-23]_{60} = [37]_{60}.$$

Also ist $d = 37$.

Privater Schlüssel: $(n, d) = (77, 37)$.

RSA: Ver- und Entschlüsselung

- Szenario: Bob möchte Alice eine verschlüsselte Nachricht senden
- Bob codiert seine Nachricht als Zahlenfolge
  (z. B. Uni- oder ASCII-Code)
- Die Nachrichtenzahlenfolge wird in Blöcke einer maximalen Größe
  ($< p \cdot q = n$) eingeteilt$^{*)}$
- Jeden Block m verschlüsselt Bob mit Alices öffentlichem Schlüssel e wie folgt:

$$c := m^e \mod n$$

- Alice entschlüsselt die erhaltenen Blöcke mit ihrem privaten Schlüssel d:

$$m = c^d \mod n$$

Weiter im Beispiel von Folie 40 und 41

Wähle die Nachricht $m := 42$. Dann ergibt sich die verschlüsselte Nachricht

$$c := m^e \mod n = 42^{13} \mod 77 \underset{\text{(Berechnung Folie 43)}}{=} 14.$$

$*)$ In der Praxis wird ein etwas komplexeres Vorgehen gewählt

Berechnung großer Potenzen

- Berechne $g^k \mod n$ für große $k \in \mathbb{N}$
- Naiver Ansatz benötigt sehr viele Multiplikationen $\underbrace{g \cdot g \cdot \ldots \cdot g}_{k-1 \text{ Multiplikationen}}$
- Square-and-Multiply-Algorithmus
  - → Zerlege k in seine Binärdarstellung:
  $$k = \sum_{j=0}^{\ell} b_j 2^j \text{ mit } b_j \in \{0,1\} \text{ und } \ell = \lfloor \log_2(k) \rfloor$$
  - → Berechne die Zweierpotenzen von g modulo n:
  $g \mod n, g^2 \mod n, g^4 \mod n = (g^2)^2 \mod n, g^8 \mod n = (g^4)^2 \mod n, \ldots$
  - → Multipliziere die entsprechenden Ergebnisse miteinander, für die $b_i \neq 0$ ist.

Berechnung zu Folie 42: $42^{13} \mod 77 = 14$

Zerlegung: $k = 13 = 8 + 4 + 1 = 2^3 + 2^2 + 2^0$. Es gilt:

$$\begin{aligned}
42^{2^0} &\mod 77 = 42 \\
42^{2^1} &\mod 77 = 70 \\
42^{2^2} &\mod 77 = 70^2 \mod 77 = 49 \\
42^{2^3} &\mod 77 = 49^2 \mod 77 = 14
\end{aligned}$$

$\rightarrow 42^{13} \mod 77 = 14 \cdot 49 \cdot 42 \mod 77 = 14$

Berechnung großer Potenzen

Aufgabe

Nach den Folien 40 und 41 gilt:
$p := 7$, $q := 11$, $n = 77$, $\varphi(n) = 60$, $e := 13$ und $d = 37$.
Entschlüsseln Sie die Nachricht $c := 14$.

Lösung

Es muss $\quad c^d \mod n = 14^{37} \mod 77 \quad$ berechnet werden.

Zerlegung: $k = 37 = 2^5 + 2^2 + 2^0$. Es gilt:

$$
\begin{aligned}
14^{2^0} &\mod 77 = 14,\\
14^{2^1} &\mod 77 = 42,\\
14^{2^2} &\mod 77 = 42^2 \mod 77 = 70,\\
14^{2^3} &\mod 77 = 70^2 \mod 77 = 49,\\
14^{2^4} &\mod 77 = 49^2 \mod 77 = 14,\\
14^{2^5} &\mod 77 = 14^2 \mod 77 = 42.
\end{aligned}
$$

Es ergibt sich: $14^{37} \mod 77 = 42 \cdot 70 \cdot 14 \mod 77 = 42 = m$.

Warum funktioniert RSA?

- Alice erhält die verschlüsselte Nachricht
$$c := m^e \mod n,$$
wobei $m \in \mathbb{N}$ mit $m < n$.
- Alice berechnet selbst $c^d \mod n$:
$$c^d \mod n = (m^e)^d \mod n = m^{e \cdot d} \mod n.$$
- Wir zeigen: $m^{e \cdot d} \mod n = m$.

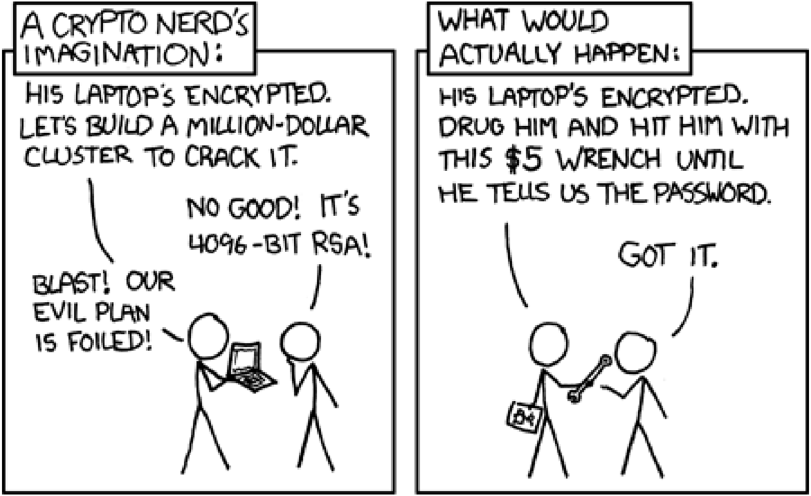
Warum ist RSA (bisher) sicher?

- Nachrichten können entschlüsselt werden, wenn der private Schlüssel d bekannt ist.
- Für die Berechnung des privaten aus dem öffentlichen Schlüssel muss $\varphi(n)$ bekannt sein. Die Berechung von $\varphi(n)$ ist äquivalent zur Primfaktorzerlegung von n.
- Es ist bislang kein effizientes Verfahren bekannt, sehr große Zahlen ($\approx$ 4096 Bit) zu faktorisieren.

Aber:

- Es ist bisher nicht bekannt, ob man auch ohne Kenntnis des privaten Schlüssels entschlüsseln kann $\rightsquigarrow$ RSA-Annahme.
- Es gibt Angriffsverfahren auf RSA-Verschlüsselung, die in der Regel auf einer schlechten Implementierung beruhen oder nur unter künstlichen Bedingungen effizient eingesetzt werden können.
- Für langfristige Archivierungen werden aktuell Schlüssel der Länge 15360 Bit empfohlen (Quelle: Europäische Agentur für Netz- und Informationssicherheit ENISA).
- Für die praktische Anwendung ist diese Schlüssellänge nicht mehr effizient.

Warum ist RSA (bisher) sicher?



Überblick

Wähle sehr, sehr große Primzahlen ...

- Gibt es überhaupt so große Primzahlen?
  $\rightarrow$ Satz von Euklid: Anzahl Primzahlen $= \infty$
- Gibt es genügend Primzahlen der gewünschten Größe?
  $\rightarrow$ pi-Funktion: $\pi(n) := |\{p \in \mathbb{P} \mid p \leq n\}|$

  $\rightarrow$ Primzahlsatz: $\pi(n) \approx \frac{n}{\ln(n)}$
- Wie findet man geeignet große Primzahlen?
  - Zunächst raten,
  - dann testen

Fermat-Test: Idee

- Formalisierung (Satz von Fermat):
  Für alle $n \in \mathbb{N}$ und Zahlen $a \in \mathbb{N}$ mit $1 \leq a \leq n-1$ gilt:
  $$n \in \mathbb{P} \Rightarrow a^{n-1} \equiv_n 1.$$
- Kontraposition (Satz von Fermat):
  Für alle $n \in \mathbb{N}$ und Zahlen $a \in \mathbb{N}$ mit $1 \leq a \leq n-1$ gilt:
  $$a^{n-1} \not\equiv_n 1 \Rightarrow n \notin \mathbb{P}.$$
- Die Umkehrung gilt im Allgemeinen nicht, d.h.:
  Gilt für natürliche Zahlen $n, a \in \mathbb{N}$ und $1 \leq a \leq n-1$
  $$a^{n-1} \equiv_n 1,$$
  so braucht n keine Primzahl zu sein!

Fermat-Test: Idee

Fermatsche Pseudoprimzahlen

- Sei $n \in \mathbb{N}$ und zusammengesetzt, und sei $a \in \{2, \ldots, n-1\}$
- n heißt fermatsche Pseudoprimzahl zur Basis a, falls

$$a^{n-1} \equiv_n 1.$$

Beispiel: 91 pseudoprim zu $3, 4, 9$, aber nicht zu $2, 5, 6, 7$

Denn: $91 = 13 \cdot 7$,

$3^{90} \equiv_{91} 1, 4^{90} \equiv_{91} 1, 9^{90} \equiv_{91} 1$ und

$2^{90} \equiv_{91} 64, 5^{90} \equiv_{91} 64, 6^{90} \equiv_{91} 64, 7^{90} \equiv_{91} 77.$

Fermat-Test: Algorithmus

- Eingabe: Ungerade natürliche Zahl n
- Ausgabe: „$n \notin \mathbb{P}$“ oder „wahrscheinlich $n \in \mathbb{P}$“
  1. *Wähle zufällig a mit* $1 < a < n$
  2. *If* $ggT(a, n) \neq 1$ *then return* „$n \notin \mathbb{P}$“
  3. *If* $a^{n-1} \mod n \neq 1$ *then return* „$n \notin \mathbb{P}$“ *(Satz von Fermat)*
  4. *else return „wahrscheinlich* $n \in \mathbb{P}$“
- Wahrscheinlichkeit, dass eine fermatsche Pseudoprimzahl gewählt wurde, ist höchstens $\frac{1}{2}$ (Fehlerwahrscheinlichkeit ist $\frac{1}{2}$)
- Wiederhole den Test mit weiteren (zufälligen!) a, um die Fehlerwahrscheinlichkeit zu senken

  $\rightsquigarrow$ Fehlerwahrscheinlichkeit nach k Schritten ist dann $\left(\frac{1}{2}\right)^k$.

Miller-Rabin-Test: Idee

- Problem beim Fermat-Test: Carmichael-Zahlen
- Carmichael-Zahl:
  $n \in \mathbb{N}$ heißt Carmichael-Zahl, wenn sie eine fermatsche Pseudoprimzahl zu allen Basen $a \in \{2, \dots, n-1\}$ mit $\mathrm{ggT}(a, n) = 1$ ist.

  Beispiel: $561 = 3 \cdot 11 \cdot 17$ ist Carmichael-Zahl.
- Idee von Gary Miller/Michael Rabin/John Selfridge:

  $\rightarrow$ Wenn $n \in \mathbb{P}$ und $a^{n-1} \equiv_n 1$ dann

  $$a^{\frac{n-1}{2}} \equiv_n 1 \text{ oder } a^{\frac{n-1}{2}} \equiv_n -1,$$

  also

  $$a^{\frac{n-1}{2}} \bmod n = 1 \text{ oder } a^{\frac{n-1}{2}} \bmod n = n-1.$$

  $\rightarrow$ Falls $\frac{n-1}{2}$ gerade und $a^{\frac{n-1}{2}} = 1 \mod n$,
  dann auch $a^{\frac{n-1}{4}} = \pm 1 \mod n$ usw.

Miller-Rabin-Test: Algorithmus

- Eingabe: Ungerade natürliche Zahl n
- Ausgabe: „$n \notin \mathbb{P}$“ oder „wahrscheinlich $n \in \mathbb{P}$“

1. *Wähle zufällig a mit* $1 < a < n$
2. *If* $ggT(a, n) \neq 1$ *then return* „$n \notin \mathbb{P}$“
3. *else berechne*
   $r, s \in \mathbb{N}$ *mit* $n - 1 = 2^r \cdot s$ *und* s *ungerade,*
   $x_0 := a^{2^0 \cdot s} \bmod n,\ x_1 := a^{2^1 \cdot s} \bmod n,\ \ldots,\ x_r := a^{2^r \cdot s} \bmod n,$
   bzw. rekursiv $x_0 := a^s \bmod n$ *und* $x_j = x_{j-1}^2 \bmod n$ *für* $j = 1, \ldots, r$.
4. *if* $x_r \neq 1$ *oder*
   $(x_0, \ldots, x_r) = (\ldots, *, 1, \ldots, 1)$ *mit* $* \neq 1$ *und* $* \neq n - 1$
   then return „$n \notin \mathbb{P}$“
5. *else return* „wahrscheinlich $n \in \mathbb{P}$“

Miller-Rabin-Test

Beispiel

Welches Ergebnis liefert der Miller-Rabin-Test für $n := 13$?

Zerlegung: $n - 1 = 12 = 2^2 \cdot 3 \rightsquigarrow r = 2$ und $s = 3$.

Wähle $a := 11$. Dann ist $\text{ggT}(a, n) = \text{ggT}(11, 13) = 1$. Weiter gilt:

$$
\begin{aligned}
x_0 &:= a^{2^0 \cdot s} \mod n = 11^{2^0 \cdot 3} \mod 13 = 11^3 \mod 13 = 5 \\
x_1 &:= a^{2^1 \cdot s} \mod n = 11^{2^1 \cdot 3} \mod 13 = 11^6 \mod 13 = 5^2 \mod 13 = 12\,(\equiv_n -1) \\
x_2 &:= a^{2^2 \cdot s} \mod n = 11^{2^2 \cdot 3} \mod 13 = 11^{12} \mod 13 = (-1)^2 \mod 13 = 1
\end{aligned}
$$

Es ergibt sich $(x_0, x_1, x_2) = (5, 12, 1)$, damit ist wahrscheinlich $13 \in \mathbb{P}$.

Wähle nun $a := 10$. Dann ist $\text{ggT}(a, n) = \text{ggT}(10, 13) = 1$. Weiter gilt:

$$
\begin{aligned}
x_0 &:= a^{2^0 \cdot s} \mod n = 10^{2^0 \cdot 3} \mod 13 = 10^3 \mod 13 = 12\,(\equiv_n -1) \\
x_1 &:= a^{2^1 \cdot s} \mod n = 10^{2^1 \cdot 3} \mod 13 = (-1)^2 \mod 13 = 1 \\
x_2 &:= a^{2^2 \cdot s} \mod n = 10^{2^2 \cdot 3} \mod 13 = 1^2 \mod 13 = 1
\end{aligned}
$$

Es ergibt sich $(x_0, x_1, x_2) = (12, 1, 1)$, damit ist wahrscheinlich $13 \in \mathbb{P}$.

Miller-Rabin-Test

Aufgabe

Untersuchen Sie für die Basen $a = 7$ und $a = 11$, ob $n = 25$ eine Primzahl ist.

Lösung

Zerlegung: $n - 1 = 24 = 2^3 \cdot 3 \rightsquigarrow r = 3$ und $s = 3$.

Wähle $a := 7$. Dann ist $\mathrm{ggT}(a, n) = \mathrm{ggT}(7, 25) = 1$. Weiter gilt:

$$
\begin{aligned}
x_0 &:= 7^{2^0 \cdot 3} \mod 25 = 7^3 \mod 25 = 18 \\
x_1 &:= 18^2 \mod 25 = 24\,(\equiv -1) \\
x_2 &:= (-1)^2 \mod 25 = 1 \\
x_3 &:= 1^2 \mod 25 = 1
\end{aligned}
$$

Es ergibt sich $(x_0, x_1, x_2, x_3) = (18, 24, 1, 1)$, damit ist wahrscheinlich $25 \in \mathbb{P}$.

Wähle $a := 11$. Dann ist $\mathrm{ggT}(a, n) = \mathrm{ggT}(11, 25) = 1$. Weiter gilt:

$$
\begin{aligned}
x_0 &:= 11^{2^0 \cdot 3} \mod 25 = 11^3 \mod 25 = 6 \\
x_1 &:= 6^2 \mod 25 = 11 \\
x_2 &:= 11^2 \mod 25 = 21 \\
x_3 &:= 21^2 \mod 25 = 16
\end{aligned}
$$

Es ergibt sich $(x_0, x_1, x_2, x_3) = (6, 11, 21, 16)$, damit ist $25 \notin \mathbb{P}$.

Miller-Rabin-Test: Korrektheit

- Problem bei Miller-Rabin: Starke Pseudoprimzahlen
- Starke Pseudoprimzahlen bestehen für manche Basen a auch den Miller-Rabin-Test
- Aber: Jede Zahl n ist für höchstens ein Viertel der Basen kleiner als n stark pseudoprim
- Erhöhe Zuverlässigkeit des Test durch wiederholte Ausführung mit anderen Basen
- Die Fehlerwahrscheinlichkeit bei k verschiedenen, zufällig gewählten Basen ist $\left(\frac{1}{4}\right)^k$
- Es ist keine starke Pseudoprimzahl bekannt, die den Miller-Rabin-Test für alle teilfremden Basen besteht. Falls die sogenannte Riemannsche Vermutung korrekt ist, besteht tatsächlich keine zusammengesetzte Zahl n diesen Test für sämtliche a kleiner als $70 \cdot \ln(n)^2$. In diesem Fall könnte man aus dem probabilistisch Primzahltest einen deterministischen Test machen, der auch noch sehr effektiv wäre.

Überblick

Definition: Diskreter Logarithmus

- Sei p eine Primzahl und $g \in \mathbb{Z}$ ein erzeugendes Element der multiplikativen Gruppe $\mathbb{Z}_p^\times$, d.h. $\langle [g]_p \rangle = \mathbb{Z}_p^\times$.
  Dafür schreiben wir auch kurz: $\langle g \rangle = \mathbb{Z}_p^\times$.
- Sei $y \in \{1, \ldots, p-1\}$. Die kleinste Zahl $x \in \mathbb{N}_0$ mit
$$g^x \mod p = y$$
  heißt der diskrete Logarithmus von y zur Basis g.
  Notation: $x := \mathrm{dlog}_g(y)$

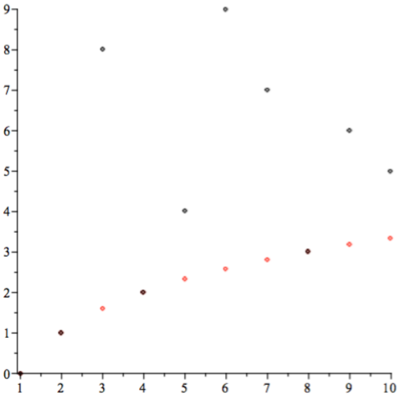
Beispiel

In $\mathbb{Z}_{11}^\times$: $\mathrm{dlog}_2(3) = ?$ oder $\mathrm{dlog}_6(7) = ?$
2 und 6 sind erzeugende Elemente in $\mathbb{Z}_{11}^\times$:

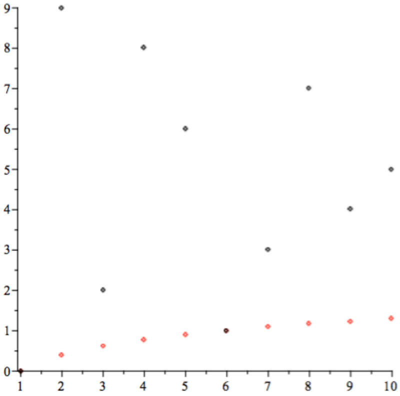
$\langle 2 \rangle = \mathbb{Z}_{11}^\times$		$\langle 6 \rangle = \mathbb{Z}_{11}^\times$	
$2^0 \mod 11 = 1$,	$2^1 \mod 11 = 2$	$6^0 \mod 11 = 1$,	$6^1 \mod 11 = 6$
$2^2 \mod 11 = 4$,	$2^3 \mod 11 = 8$	$6^2 \mod 11 = 3$,	$6^3 \mod 11 = 7$
$2^4 \mod 11 = 5$,	$2^5 \mod 11 = 10$	$6^4 \mod 11 = 9$,	$6^5 \mod 11 = 10$
$2^6 \mod 11 = 9$,	$2^7 \mod 11 = 7$	$6^6 \mod 11 = 5$,	$6^7 \mod 11 = 8$
$2^8 \mod 11 = 3$,	$2^9 \mod 11 = 6$	$6^8 \mod 11 = 4$,	$6^9 \mod 11 = 2$

Es ergibt sich: $\mathrm{dlog}_2(3) = 8$ und $\mathrm{dlog}_6(7) = 3$.

Graphik: Diskreter Logarithmus in $\mathbb{Z}_{11}$



$\circ$ $\log_2(y)$ und $\circ$ $\mathrm{dlog}_2(y)$



$\circ$ $\log_6(y)$ und $\circ$ $\mathrm{dlog}_6(y)$

Diffie-Hellman-Key-Exchange

- Verfahren zum Austausch eines symmetrischen Schlüssels über ein unsicheres Medium
- Basiert auf der in der Praxis zu aufwendigen Berechnung von diskreten Logarithmen
- 1976 veröffentlicht von

  Martin Hellman,
  Whitfield Diffie und
  Ralph Merkle
- Ähnliches Verfahren bereits Anfang der 1970er von James Ellis, Clifford Cocks und Malcom J. Williamson entwickelt, aber nicht veröffentlicht

Diffie-Hellman-Key-Exchange: Ablauf des Verfahrens

- Alice und Bob wählen öffentlich eine große Primzahl p und ein erzeugendes Element g von $\left(\mathbb{Z}_p^\times, \otimes\right)$.
- Alice wählt geheime Zahl $a \in \{1, \ldots, p-1\}$, bildet $x := g^a \mod p$ und schickt x an Bob.
- Bob wählt geheime Zahl $b \in \{1, \ldots, p-1\}$, bildet $y := g^b \mod p$ und schickt y an Alice.
- Gemeinsamer privater Schlüssel: $g^{a \cdot b} \mod p =: z$.
- Alice berechnet gemeinsamen privaten Schlüssel

$$y^a \mod p = (g^b)^a \mod p = g^{a \cdot b} \mod p = z.$$

- Bob berechnet gemeinsamen privaten Schlüssel als

$$x^b \mod p = (g^a)^b \mod p = g^{a \cdot b} \mod p = z.$$

- Lauscher kennt nur p, g, x und y. Daraus lässt sich z nur mit sehr großem Aufwand berechnen.

Diffie-Hellman-Key-Exchange

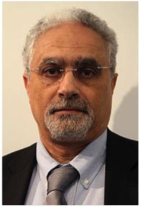
Beispiel

1. Alice und Bob einigen sich öffentlich auf $p = 13$ und $g = 2$.
2. Alice wählt die Zufallszahl $a = 5$. Bob wählt die Zufallszahl $b = 8$.
3. Alice berechnet $x = g^a \mod p = 6$ und sendet x an Bob.
4. Bob berechnet $y = g^b \mod p = 9$ und sendet y an Alice.
5. Alice berechnet $z = y^a \mod p = 3$.
6. Bob berechnet $x^b \mod p = 3 = z$.
7. Beide erhalten den Schlüssel $z = 3$.

Bemerkung

Lauscher Eve muss also a bzw. b aus $x = g^a \mod p$ bzw. $y = g^b \mod p$ berechnen. Dies führt auf die Berechnung von $\mathrm{dlog}_g(x)$ bzw. $\mathrm{dlog}_g(y)$ in $\mathbb{Z}_p^\times$.

ElGamal-Kryptosystem



Taher ElGamal

- Asymmetrisches Verschlüsselungsverfahren
  - $\rightarrow$ basiert auf der Idee des Diffie-Hellman-Key-Exchange
  - $\rightarrow$ 1985 entwickelt von Taher ElGamal
- Öffentliche Parameter
  - $\rightarrow$ Primzahl p
  - $\rightarrow$ Erzeugendes Element g von $(\mathbb{Z}_p^\times, \otimes)$
  - $\rightarrow$ Nachrichtenverschlüsselung: Multiplikation (oder XOR-Verknüpfung) mit Diffie-Hellman-Schlüssel

ElGamal: Ablauf

- Szenario: Bob sendet eine Nachricht $m < p$ an Alice (Ist $m > p$: Einteilung in Blöcke).
- Alice wählt geheimes $a \in \{1, \ldots, p-1\}$.
- Alice veröffentlicht öffentlichen Schlüssel $x := g^a \mod p$.
- Verschlüsselung:
  Bob wählt zufälliges $b \in \{1, \ldots, p-1\}$, bildet $y := g^b \mod p$ und $c := m \cdot x^b \mod p = m \cdot g^{a \cdot b} \mod p$ und sendet (y, c) an Alice.
- Entschlüsselung:
  Alice berechnet: $y^{p-1-a} \cdot c \mod p = m$ (beachte: $p - 1 - a \geq 0$).

Bemerkung

Es wird der Schlüssel $z := g^{a \cdot b} \mod p$ verwendet, und die Verschlüsselung erfolgt durch Multiplikation mit z modulo p.

ElGamal: Ablauf

Beispiel

1. Alice und Bob einigen sich öffentlich auf $p := 23$ und $g := 7$.
2. Alice wählt geheim $a := 5$ und berechnet den öffentlichen Schlüssel
$$x := g^a \mod p = 7^5 \mod 23 = 17.$$
3. Bob möchte die Nachricht $m := 8$ an Alice verschlüsselt versenden. Dazu wählt Bob geheim $b := 3$.
   - (3.1) Bob verschlüsselt die Nachricht
   $$c := m \cdot x^b \mod p = 8 \cdot 17^3 \mod 23 = 20.$$
   - (3.2) Damit Alice entschlüsseln kann, berechnet Bob
   $$y := g^b \mod p = 7^3 \mod 23 = 21.$$
4. Bob sendet das Paar $(y, c) = (21, 20)$ an Alice.
5. Alice entschlüsselt die Nachricht
$$y^{p-1-a} \cdot c \mod p = 21^{17} \cdot 20 \mod 23 = 5 \cdot 20 \mod 23 = 8 = m.$$

Warum funktioniert ElGamal?

- Es gilt $y = g^b \mod p$, also $[y]_p = [g^b]_p$.
- Nach dem Satz von Fermat gilt $[y]_p^{p-1} = [1]_p$.
- Wir berechnen in $\mathbb{Z}_p$:

$$[y^{p-1-a}]_p = [y]_p^{p-1-a} = [y]_p^{p-1} \otimes [y]_p^{-a} = [1]_p \otimes [g^b]_p^{-a} = [g]_p^{-a \cdot b}.$$

Damit folgt:

$$\begin{aligned} [y^{p-1-a} \cdot c]_p &= [y^{p-1-a}]_p \otimes [c]_p = [g]_p^{-a \cdot b} \otimes \left([m]_p \otimes [g]_p^{a \cdot b}\right) \\ &= [g]_p^{-a \cdot b + a \cdot b} \otimes [m]_p = [g]_p^0 \otimes [m]_p = [m]_p. \end{aligned}$$

Wegen $m < p$ folgt

$$y^{p-1-a} \cdot c \mod p = m.$$

Baby-Step-Giant-Step-Algorithmus

- Algorithmus zur effizienteren Berechnung des diskreten Logarithmus.
- Entwickelt 1971 von Daniel Shanks
- Problemstellung
  - → Gegeben: Primzahl p, erzeugendes Element g von $(\mathbb{Z}_p^\times, \otimes)$
  - → Gesucht: $x = \mathrm{dlog}_g(y)$, d.h. ein $x \in \mathbb{N}_0$ mit $g^x \mod p = y$
- Ansatz
  - → Nutze Darstellung $x = q \cdot s + r$ mit $s = \left\lceil \sqrt{p-1} \right\rceil$
  - → Teile Berechnung auf in

    Giant-Steps (für $q \cdot s$) und Baby-Steps (für r)

Baby-Step-Giant-Step: Herleitung

- Berechne $s := \left\lceil \sqrt{p-1} \right\rceil$
  (kleinste ganze Zahl größer oder gleich $\sqrt{p-1}$)
- Ansatz: $x = q \cdot s + r$ mit $q, r \in \{0, \ldots, s-1\}$
- Dann ist $y = g^x \mod p = g^{q \cdot s + r} \mod p$.
- Damit gilt: $g^{q \cdot s} \mod p = y \cdot g^{-r} \mod p$.
  Es ist $g^{-r} = \left(g^{-1}\right)^r$, wobei $g^{-1} \in \mathbb{Z}$ einen Repräsentanten von $[g]_p^{-1}$ bezeichne.
- Berechne Giant-Steps: $g^{q \cdot s} \mod p$.
  $\rightarrow$ Speichere die berechneten Werte in einer Tabelle.
- Berechne Baby-Steps: $y \cdot g^{-r} \mod p$.
  $\rightarrow$ Bis ein Wert einem Giant-Step-Wert entspricht.
- Ermittle Ergebnis $x = q \cdot s + r$ mit r aus dem Baby-Step und q aus dem zugehörigen Giant-Step.

Baby-Step-Giant-Step

Beispiel

Gegeben seien $p := 31$ und $g := 17$.

Wir möchten $x = \mathrm{dlog}_g(y) = \mathrm{dlog}_{17}(23) = q \cdot s + r$ berechnen.

Es ist $s = \lceil \sqrt{p-1} \rceil = \lceil \sqrt{30} \rceil \underset{\sqrt{30} \approx 5.47}{=} 6$. Also sind $q, r \in \{0, \ldots, 5\}$.

Giant-Steps-Tabelle:

q	0	1	2	3	4	5
$g^{q \cdot s} \mod p$ $= (17^6)^q \mod 31 = 8^q \mod 31$	1	8	2	16	4	1

Baby-Steps-Tabelle:

r	0	1	2	3	4	5
$y \cdot g^{-r} \mod p$ $= 23 \cdot (17^{-1})^r \mod 31 = 23 \cdot 11^r \mod 31$	23	5	24	16		

Also ist $x = \mathrm{dlog}_g(y) = \mathrm{dlog}_{17}(23) = q \cdot s + r = 3 \cdot 6 + 3 = 21$.

Probe: $g^x \mod p = 17^{21} \mod 31 = 23 = y$.

Baby-Step-Giant-Step

Aufgabe

Eve erlauscht im Zuge eines DH-Schlüsselaustauschs die berechnete Zahl $y = 17$ von Bob. Außerdem hat sie Zugang zu den öffentlichen Zahlen $p = 29$ und $g = 15$. Berechnen Sie die geheime Zahl b von Bob.

Lösung

Wir möchten $b = \mathrm{dlog}_g(y) = \mathrm{dlog}_{15}(17) = q \cdot s + r$ berechnen.
Es ist $s = \lceil\sqrt{p-1}\rceil = \lceil\sqrt{28}\rceil \underset{\sqrt{28}\approx 5.29}{=} 6$. Also sind $q, r \in \{0, \ldots, 5\}$.

Giant-Steps-Tabelle:

q	0	1	2	3	4	5
$g^{q \cdot s} \mod p$ $= (15^6)^q \mod 29 = 5^q \mod 29$	1	5	25	9	16	22

Baby-Steps-Tabelle:

r	0	1	2	3	4	5
$y \cdot g^{-r} \mod p$ $= 17 \cdot (15^{-1})^r \mod 29 = 17 \cdot 2^r \mod 29$	17	5				

Also ist $b = \mathrm{dlog}_g(y) = \mathrm{dlog}_{15}(17) = q \cdot s + r = 1 \cdot 6 + 1 = 7$.

Baby-Step-Giant-Step: Aufwand

- Giant-Steps
  - $\rightarrow$ $s \approx \sqrt{p}$ Berechnungen
  - $\rightarrow$ Speicherbedarf somit ebenfalls $\approx \sqrt{p}$
  - $\rightarrow$ Im Voraus berechenbar, da unabhängig von y
- Baby-Steps
  - $\rightarrow$ Zwischen 1 und s Berechnungen (im Mittel $\frac{s}{2}$)
  - $\rightarrow$ Kein zusätzlicher Speicherbedarf
  - $\rightarrow$ Kein nennenswerter zusätzlicher Aufwand durch Abgleich mit Tabelle bei geeigneter Datenstruktur (Hash-Tabelle)
- Gesamt
  - $\rightarrow$ Rechenzeit $O(\sqrt{p})$
  - $\rightarrow$ Speicherbedarf $O(\sqrt{p})$
  - $\rightarrow$ Ab etwa $p > 2^{160}$ in der Praxis nicht mehr einsetzbar (Stand 2003)

Verallgemeinerung

- Alle Verfahren dieses Kapitels lassen sich auch mit einer beliebigen zyklischen Gruppe $(G, \circ)$ anstelle von $(\mathbb{Z}_p^\times, \otimes)$ durchführen.
- Ist $G = \langle g \rangle$, so wird auch in diesem Fall allgemein der diskrete Logarithmus $x = \mathrm{dlog}_g(y)$ eines Elements $y \in G$ zur Basis g definiert als kleinste Zahl $x \in \mathbb{N}_0$ mit $g^x = y$.
  (Erinnerung: $g^n = \underbrace{g \circ \cdots \circ g}_{n-\text{mal}}$ für $n \in \mathbb{N}$ und $g^0 := 0$).
- Anforderungen: Die Potenzen $y = g^x$ sollen möglichst effizient berechnet werden können, die diskreten Logarithmen $x = \mathrm{dlog}_g(y)$ hingegen möglichst schwierig.
- Besonders geeignet sind hierbei sogenannte elliptische Kurven $E(K)$ über einem Körper K.
- Praxis: Verwende $E(\mathbb{Z}_p) \rightsquigarrow$ ECC (Elliptic Curve Cryptography).
- Vorteil gegenüber RSA: deutlich kürzere Schlüssel führen zur gleichen Sicherheit.
  Beispiel: 512 Bit ECC $\approx$ 15360 Bit RSA.

Überblick

Erreichte Schutzziele

- Vertraulichkeit:
  Ja, abhängig von der Stärke des verwendeten Kryptosystems
- Integrität:
  Sehr begrenzt, gezielte Veränderungen werden nicht sicher entdeckt
- Authentizität:
  - → Symmetrische Verfahren
    (Ja): Ein gemeinsamer, geheimer Schlüssel wird verwendet. Aber: Es muss immer die vollständige Nachricht verschlüsselt werden.
  - → Asymmetrische Verfahren
    Nein, Schlüssel zum Verschlüsseln ist öffentlich.
- Nicht-Abstreitbarkeit:
  - → Symmetrische Verfahren
    Nein: Auch der Empfänger kann verschlüsseln
  - → Asymmetrische Verfahren
    Nein: Jeder kann mit öffentlichen Schlüssel verschlüsseln

Hash-Funktionen

- Sei Σ ein Alphabet, aus dessen Buchstaben Zeichenfolgen gebildet werden, z.B.:
  - $\rightarrow$ $\Sigma = \{0, 1\}$
  - $\rightarrow$ $\Sigma = \{0, 1, 2, \ldots, D, E, F\}$
- Eine Hash-Funktion $h : \Sigma^* \rightarrow \Sigma^n$ bildet Zeichenfolgen beliebiger Länge in Zeichenfolgen fester Länge ab.
- Anwendungsbereich
  - $\rightarrow$ Verwaltung großer Datenbestände (Hashtabellen)
  - $\rightarrow$ Prüfsummen
  - $\rightarrow$ Kryptologie

Kryptologische Hash-Funktionen

- Wichtiges Einsatzgebiet: Digitaler Fingerabdruck
  - $\rightarrow$ Sei m eine Nachricht.
  - $\rightarrow$ Ihr Hash-Wert $h(m)$ ist ihr digitaler Fingerabdruck.
- Anforderungen an kryptologische Hash-Funktion h
  - $\rightarrow$ Effizient berechenbar (aber nicht zu effizient!)
  - $\rightarrow$ Einwegfunktion: m aus $h(m)$ praktisch nicht rekonstruierbar.
  - $\rightarrow$ Schwache Kollisionsresistenz: Zu m lässt sich praktisch kein m' finden mit $m \neq m'$ und $h(m) = h(m')$.
  - $\rightarrow$ Starke Kollisionsresistenz: Es lassen sich praktisch keine m und m' finden mit $m \neq m'$ und $h(m) = h(m')$.
- Bekannte kryptologische Hash-Funktionen

  MD5 (128-Bit-Hashwert), SHA-1 (160-Bit-Hashwert) $\rightsquigarrow$ veraltet!
  SHA-2 (224-, 256-, 384- oder 512-Bit)
  SHA-3 (256-, 384- oder 512-Bit)

Message Authentication Code

- Ablauf:
  - $\rightarrow$ Sender und Empfänger
    - – tauschen geheimen Schlüssel k aus
    - – einigen sich auf schlüsselabhängige Hash-Funktion h
  - $\rightarrow$ Sender berechnet Message Authentication Code (MAC): $MAC := h(m, k)$ der zu sendenden Nachricht m
  - $\rightarrow$ Sender übermittelt MAC zusammen mit m
  - $\rightarrow$ Empfänger berechnet MAC der erhaltenen Nachricht m'
  - $\rightarrow$ Falls $h(m, k) = h(m', k)$ ist, dann wurde m nicht verändert
- Vorteile:
  - $\rightarrow$ Integrität bei guter kryptologischer Hash-Funktion gewährleistet
  - $\rightarrow$ Integritätsprüfung auch ohne Verschlüsselung der Nachricht möglich

Digitale Signatur am Beispiel RSA

Szenario: Bob sendet Alice Nachricht m

Ablauf:

- Alice und Bob einigen sich auf eine öffentliche, schlüssellose kryptologische Hash-Funktion
- Bob ermittelt Hash-Wert $h(m)$ der Nachricht m
- Bob berechnet $s := h(m)^d \mod n$ mit seinem privaten Schlüssel d
- Bob schickt signiertes Dokument (m, s) an Alice
- Alice erhält (m', s')
- Alice bestimmt zur erhaltenen Nachricht m' den Hash-Wert $h(m')$
- Alice berechnet $(s')^e$ mit Bobs öffentlichem Schlüssel e
- Falls $(s')^e \mod n = h(m')$ ist, dann haben wir die Integrität und Authentizität gewährleistet

RSA-Signaturen

- Umgekehrte Reihenfolge der Schlüsselanwendung
  - → ist möglich, da Ver- und Entschlüsselung kommutativ
- Verschlüsselung und Signierung kombinierbar
  - → Verschlüsselung: Schlüssel des Empfängers
  - → Signierung: Schlüssel des Senders
- Vorteile durch Verwendung von Hash-Funktionen
  - → effizienter (nur der Hash-Wert muss ver- und entschlüsselt werden)
  - → größere Sicherheit gegenüber bestimmten Angriffsarten

Digitale Signaturen allgemein

- Digitale Signaturen grundsätzlich auch mit anderen asymmetrischen Verfahren möglich
- Beispiele
  - → ElGamal-Signaturen
  - → Digital Signatur Standard (DSS)
- Schutzziel Nicht-Abstreitbarkeit durch digitale Signatur erfüllt
  - → Versender kann Versand der signierten Nachricht nicht abstreiten
  - → Nur Versender besitzt notwendigen privaten Schlüssel

Public-Key-Infrastructure

- Angriffsszenario von Public-Key-Systemen:
  Fälschung der öffentlichen Schlüssel
- Lösungsansatz: Public-Key-Infrastructure (PKI)
  - → Vertrauenswürdiges Managementsystem zur sicheren Verwaltung der öffentlichen Schlüssel
  - → Digitale Zertifikate gewährleisten Echtheit öffentlicher Schlüssel
  - → Bestandteile einer PKI
    - – Zertifizierungsstelle (Certification Authority, CA)
    - – Registrierungsstelle (Registration Authority, RA)
    - – Validierungsstelle (Validation Authority, VA)
- Grundsätzliche Ansätze
  - → Hierarchische PKI
  - → Web of Trust (z.B. Pretty Good Privacy (PGP))

Vielen Dank für Ihre Aufmerksamkeit