Kryptographie - Mitschrift

Sommersemester 2014

Prof. Ruppert

Autoren:

- Malte Kraus
- Christian Strate

Inhaltsverzeichnis

0 Organisatorisches

Vorlesungen: Mo: 0815 - 0945 HE
Do: 0815 - 0945 H4

www.math.fau.de/ruppert - Kryptographie

Übungen: $\approx 50\%$ bearbeitet, keine Bewertung. Abgabe erfolgt Montags in der VL.

Note:

Dieses Skript wird lediglich Abweichungen von Tafelanschrieb und bereitgestelltem Skript[1] oder für besonders wichtig erachtete Passagen festhalten.

Da es sich hierbei lediglich um eine inoffizielle Mitschrift von Studenten handelt, sind weder Garantie auf Vollständigkeit noch auf Korrektheit gewährleistet.

Notes zu aktuellen Aufgaben:

55) Fermat-Zahlen: $F_n = 2^{2^n} + 1, n \in \mathbb{N}_0, F_0 = 3, F_1 = 5, F_2 = 17, F_3 = 257, F_4 = 2^{16} + 1 = 65537$ wird oft als öffentlicher RSA-Exponent verwendet

59) Probiere Kryptographie-Vorlesung als Schlüssel.

Klausur:

- Datum: 28. Juli
- Zeit: update: 10:15 - 12:15
- Raum: H11, H12
- zugelassene Hilfsmittel: lediglich ein Taschenrechner
- Nachholklausur: 29.09.2014? Terminfindungs-Doodle
- Seitens des Professors kam der Wunsch auf doch bitte über die Umfrage Bescheid zu geben, ob man am ersten Termin teilnehmen wird.

Zu der am 15.07. veröffentlichten Probeklausur ist nun ein Lösungsvorschlag verfügbar.

Sollten Fragen aufkommen oder Fehler entdeckt werden, bitte ich um Kontaktaufnahme via Mail (Christian Strate).

[1] Da das überarbeitete, offizielle Skript gelegentlich auf sich warten lässt, orientiert sich der Inhalt großteils am alten Skript (SS12). Dementsprechend können eher triviale Passagen in dieser Mitschrift Erwähnung finden, obgleich sie im neuen, offiziellen Skript auftauchen.

1 Einführung

1.1 Ein Grundproblem der Kryptographie

1.2 Einfache Substitutionschiffren

Merkwürdige Schreibweise für XOR: $a \wedge b$

Kryptosystem CAESAR - Seite 3.
Kryptosystem MASC - Seite 4.

Bemerkung:

CAESAR ist ein Spezialfall von MASC: z.B. CAESAR-3 ist MASC-Verschlüsselung mit D. Etwas allgemeiner: MASC-Verschlüsselung mit einem Einbuchstabenwort ist eine CAESAR-Verschlüsslung.

1.3 Häufigkeitsanalyse

Tabellen für Häufigkeitsvorkommen für Englisch und Deutsch - Seite 8.
Tabellen für Häufigkeiten deutscher Bigramme, Trigramme. - Seite 9.
Häufigsten deutschen Wörter:

die, der, und, den, am, in, zu, ist, dass, es

1.4 Transpositionschiffren

Kryptosystem TRANSMAT - Seite 9. Kryptosystem TRANSSPA - Seite 9f.

Beispiel zu TRANSSPA:

Plaintext: AMDONNERSTAGENTFAELLTDIEVORLESUNG
Chiffretext: OGLLXASAVGETIUXNNDSXDARLXNETEXRFENXMTEOX

E	R	L	A	N	G	E	N
A	M	D	O	N	N	E	R
S	T	A	G	E	N	T	F
A	E	L	L	T	D	I	E
V	O	R	L	E	S	U	N
G	X	X	X	X	X	X	X

Die Spalten werden also anhand der alphabetischen Ordnung des Schlüsselwortes ausgegeben. Bei Mehrfachvorkommen einzelner Buchstaben im Schlüsselwort, wird bei den jeweils ersten Vorkommen begonnen und bei den letzten Vorkommen gestoppt.

1.5 Blockchiffren

Kryptosystem ALBC-2 - Seite 10f.
ALBC-2:

- 5. $d \in \mathbf{Z}$, alle l_x zusätzlich mit $\%N$ hinten dran? (laut Tafel, allerdings bin ich mir nicht sicher, ob das stimmt)
- bei dem Beispiel: 12: M, 17: R. Auch hier kann zur Vereinfachung mit den Klassen gerechnet werden. $\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 5 & 8 \\ 23 & 15 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 17 \\ 14 \end{pmatrix} \%26, \begin{pmatrix} x \\ y \end{pmatrix} = ZA = \begin{pmatrix} 25 \\ 0 \end{pmatrix}$
- Bemerkung: $p|N$ bedeutet Primteiler von 26 das sind 2 und 13 $\rightarrow 26^6 \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{2^2}\right) \cdot \left(1 - \frac{1}{13}\right) \cdot \left(1 - \frac{1}{13^2}\right)$

1.6 Stromchiffren

Kryptosystem STROM - Seite 13f.
Kryptosystem VERNAM CIPHER - Seite 14.

1.6.1 AUTOKEY-Verschlüsselung

Seite 15.
Der Schlüssel ist zunächst als Schlüsselwort $k_1 k_2 ... k_n$ gegeben. Die Schlüsselfolge erhält man, indem man an das Schlüsselwort den zu verschlüsselnden Text anhängt.

Text	a_1	a_2	a_3	...	a_n	a_{n+1}	a_{n+2}	...
Schlüssel	k_1	k_2	k_3	...	k_n	a_1	a_2	...

d.h. $K_{n+1} = a_i, i \geq 1$
Beispiel:

Text “SOMMERSEMESTER” Schlüsselwort “MONTAG”

Text	S	O	M	M	E	R	S	E	M	E	S	T	E	R
Schlüssel	M	O	N	T	A	G	S	O	M	M	E	R	S	E
Chiffretext	E	C	Z	F	E	...								

1.7 Angriffe-Kryptoanalyse

2 Klassische Chiffrierverfahren

2.1 Vigenère-Verschlüsselung

Kryptosystem Vigenère - Seite 1.
Kasiki Test - Seite 1f.

2.2 Playfair

Playfair-Chiffrierung - Seite 5.
Das Folgende ist eine vorangehende, weniger formale Ergänzung zum vierten Punkt der Playfair-Chiffre. Die Beispiele orientieren sich an der Tabelle, mit dem Schlüsselwort *Kryptographie*, des obig verlinktem Skriptes.

Fall a_1, a_2 stehen weder in gleicher Zeile noch in gleicher Spalte. $i_1 \neq i_2, j_i \neq j_2$:
Dann betrachtet man das Rechteck, das a_1, a_2 erzeugt.

$a_1a_2 \to b_1b_2$, wobei b_1 in der gleichen Zeile wie a_1 und in der gleichen Spalte wie a_2
wobei b_2 in der gleichen Zeile wie a_2 und in der gleichen Spalte wie a_1

Beispiel:

$SO \to LI$,	$MX \to QV$,	$ME \to LB$,	$RX \to PV$
$O - I$	$M - Q$	$E - B$	$R - P$
$L - S$	$V - X$	$L - M$	$V - X$

Fall a_1, a_2 stehen in der gleichen Zeile. $i_1 = i_2$:
Dann ist b_i der rechts neben a_i stehende Buchstabe
(steht a_i in der letzten Spalte, nimmt man für b_i den Buchstaben der ersten.)
Beispiel:
$EC \to BD$, $CE \to DB$, $DF \to FE$

Fall a_1, a_2 stehen in der gleichen Spalte. $j_1 = j_2$:
Dann ist b_i das auf a_i folgende Zeichen in der gleichen Spalte
(steht a_i in der letzten Zeile, nimmt man für b_i den Buchstaben der ersten.)
Beispiel:
$TI \to IF$, $IT \to FI$, $FZ \to ST$

komplettes Beispiel:
"HEUTE IST MONTAG" (Aus der Tabelle lesen)

HE	UT	EI	ST	MO	NT	AG
OD	ZK	FO	ZI	LG	SY	HA

2.3 Homophone Substitutionschiffrierung

Homophone Substitutionschiffrierung - Seite 1 bzw. 7
Wichtig ist hier Disjunktion der einzelnen Teilmengen $F(a) \subseteq \sum_2$ für verschiedene a's.
Also $a \neq b \Rightarrow F(a) \cap F(b) = \emptyset$

3 Grundeigenschaften der Ringe Z und Z/*n*Z

Kapitel 3 - Dieses Kapitel scheint dieses Semester nicht weiter überarbeitet zu werden.

3.1 Die Landau-Symbole

Seite 1

3.2 Grundlegende Eigenschaften der natürlichen und ganzen Zahlen

Seite 2ff.
Teilbarkeit: $\exists c \in \mathbb{Z}.\quad a = b \cdot c \Rightarrow b|a, \qquad a, b \in \mathbb{Z}$

- $a|a, \quad 1|a, \quad a|0$
- $a|b, b|c \Rightarrow a|c$
- $a|b \Leftrightarrow \pm a| \pm b$
- $a, b \neq 0 \Rightarrow (a|b, b|a \Leftrightarrow b = \pm a)$
- $a|1 \Leftrightarrow a = \pm 1$
- $a, b \in \mathbb{Z} \setminus 0 \Rightarrow \forall i. \quad \pm \prod_i p_i^{a_i} | \pm \prod_i p_i^{b_i} \Leftrightarrow a_i \leq b_i$
- [2] $a|c, b|c, ggT(a, b) = 1 \Rightarrow a \cdot b|c$
- $a|b \cdot c, ggT(a, c) = 1 \Rightarrow a|b$

Fundamentalsatz der Arithmetik. Seite 3
Naiv Primfaktorzerlegung einer natürlichen Zahl n bestimmen.
Beispiel $n = 100002$:

$$100002 = 2 \cdot 50001 = 2 \cdot 3 \cdot 16667 = 2 \cdot 3 \cdot 7 \cdot 2381$$

Dies entspricht dem "Sieb des Eratosthenes"-Verfahren, ohne Streams. Seite 3

Beispiel: $n = 300006$, es wird $2, 3$ heraus geteilt

$n = 2 \cdot 150003 = 2 \cdot 3 \cdot 50001 = 2 \cdot 3^2 \cdot 16667$
16667 ist durch 7 teilbar: $16667 = 7 \cdot 2381$
Wir müssen nun für $t = 9, 11, ...$ prüfen, ob $t|2381$ für $t^2 \leq 2381$, was äquivalent zu $t \leq \sqrt{2381} = 48.79...$ ist. Man findet, dass keine Zahl zwischen 9 und 47 die 2381 teilt. Also ist 2381 eine Primzahl. Wir erhalten insgesamt: $300006 = 2 \cdot 3^2 \cdot 7 \cdot 2381$

[2] eine Verlinkung des ggT's findet sich weiter unten.

Diese Art des Faktorisierens nennen wir "naives Faktoriesierungsverfahren"

Bemerkung:
Die Bedingung $t^2 \leq n$ ist äquivalent mit $t \leq \sqrt{n}$, also $t \leq \lfloor \sqrt{n} \rfloor$. Kann man schnell $t \leq \lfloor \sqrt{n} \rfloor$ berechnen, so kann man $t^2 > n$ durch die Bedingung $t > \lfloor \sqrt{n} \rfloor$ ersetzen.

Bemerkung:
Primzahlen. Wie lange braucht man um festzustellen, dass n eine Primzahl ist? Man muss mit t bis ungefähr $\sqrt{n}$ gehen.

Es folgen eine etwas merkwürdige Definition der Polynomial- und Exponentiallaufzeit, sowie die Definitionen gT, ggT, gV, kgV. Seite 5

Achtung:
$gT(0,0) = \mathbb{Z}$; und insbesondere $ggT(0,0) = 0$

3.3 Der euklidische Algorithmus

Seite 7ff.

3.4 Fibonacci-Zahlen

Seite 10ff.
Laufzeit des euklidischen Algorithmus:
$ggT(a,b), a, b \in \mathbb{N}, a > b$ benötigt maximal 4.785 $log_{10}\ a$ Divisionen mit Rest

3.5 Der erweiterte euklidische Algorithmus

Seite 12ff.
Beispiel zum ersten Satz $\rightsquigarrow$ Satz 2 Mathe3-Skript:
$ggT(15,9) = ggT(3 \cdot 5, 3^2) = 3 = (-1) \cdot 15 + 2 \cdot 9$

Für "kleine" Zahlen kann man auch manchmal folgendes Verfahren finden: Schreibe a_n im euklidischen Algorithmus von "unten nach oben" bis man a_n als Linearkombination von a_0 und a_1 erhält.

Beispiel: $ggT(12345, 987)$

$$\begin{aligned} 3 &= 15 - 2 \cdot 6 = (501 - 1 \cdot 486) - 2 \cdot (486 - 32 \cdot 15) = 501 - 3 \cdot 486 + 64 \cdot 15 = \\ &= 501 - 3 \cdot 486 + 64 \cdot (501 - 1 \cdot 486) = 65 \cdot 501 - 67 \cdot 486 = \\ &= 65 \cdot (12345 - 12 \cdot 987) - 67 \cdot (987 - 1 \cdot 501) = \\ &= 65 \cdot 12345 - 847 \cdot 987 + 67(12345 - 12 \cdot 987) = 132 \cdot 12345 - 1651 \cdot 987 \end{aligned}$$

$x_i = x_{i-2} - q_{i-2} \cdot x_{i-1}, \qquad y_i = y_{i-2} - q_{i-2} \cdot y_{i-1}$

3.6 Die Gleichung $ax + by = c$

Seite 14f.

Vorbemerkung zum ersten Satz:

Sind $x_0, y_0 \in \mathbf{Z}$ mit $ggT(a,b) = x_0 a + y_0 b$,
so gilt für alle $k \in \mathbf{Z}$: $ggT(a,b) = (x_0 + kb) \cdot a + (y_0 - ka) \cdot b$

erster Teil des besagten Satzes:

$a, b \in \mathbb{Z}, (a,b) \neq (0,0) \Rightarrow$ $ax + by = c$ ist genau dann Lösbar, wenn $ggT(a,b) | c$

essenzielle Folgerung:

$a, b \in \mathbb{Z}, ggT(a,b) = 1 \Rightarrow$ $\exists x_0, y_0 \in \mathbb{Z}. \quad ax_0 + by_0 = 1,$
$\{(x,y) \in \mathbb{Z} \times \mathbb{Z} : ax + by = 1\} = \{(x_0 + bm, y_0 - am) : m \in \mathbb{Z}\}$

$a \cdot x + b \cdot y = c$, seien $a = 4, b = 2, c = 1, x = 1, y = -1.5 \Rightarrow 4 - 3 = 1$ aber $ggT(4,2) = 2$ und $a|1 \Leftrightarrow a = \pm 1 \neq 2$

3.7 Kongruenzen

Seite 15ff.

Verträglichkeit der Äquivalenzrelation mit Addition und Multiplikation.
$\overline{a} + \overline{b} = \overline{a + b}$ $\overline{a}\overline{b} = \overline{ab}$

Vorbemerkung zum ersten Satz:

Seien $m_1, m_2 \in \mathbf{N}$ mit $ggT(m_1, m_2) = 1$ und $a_1, a_2 \in \mathbf{Z}$. Mit dem erweiterten euklidischen Algorithmus findet man $u, v \in \mathbf{Z}$ mit $um_1 + vm_2 = 1$. Dann löst $a = a_2 u m_1 + a_1 v m_2$ das Gleichungssystem $x \equiv a_1 \ mod \ m_1, x \equiv a_2 \ mod \ m_2$
$1 = um_1 + vm_2$
$a = a_2 u m_1 + a_1 v m_2$

Beispiel:

Bestimme die kleinste Lösung von $x \equiv 2 \ mod \ 25, x \equiv 5 \ mod \ 52$ in natürlichen Zahlen. Es ist $ggT(25, 52) = 1$, das Gleichungssystem ist lösbar, Eindeutigkeit modulo $25 \cdot 52 = 1300$

Wir wenden den euklidischen Algorithmus auf 52 und 25 an:

$$52 = 2 \cdot 25 + 2$$
$$25 = 12 \cdot 2 + 1$$
$$2 = 2 \cdot 1 + 0$$

$$1 = 25 - 12 \cdot 2 \quad = \quad 25 - 12 \cdot (52 - 2 \cdot 25) \quad = \quad (-12) \cdot 52 + 25 \cdot 25$$

Also: $1 = (-12) \cdot 52 + 25 \cdot 25$
Sei $a = 2 \cdot (-12) \cdot 52 + 5 \cdot 25 \cdot 25 = 1877$
a löst die Kongruenzgleichungen. $1877 - 1300 = 577$ ist die kleinste Lösung in den natürlichen Zahlen.

Nicht jedes Kongruenzgleichungssystem ist lösbar.

Beispiel: $x \equiv 4 \ mod \ 45, x \equiv 5 \ mod \ 54$
Hier ist $ggT(45, 54) = 9$. Wir betrachten die Gleichungen *modulo* 9, dem ggT.

$$x \equiv 4 \ mod \ 45 \qquad \Rightarrow x \equiv 4 \ mod \ 9$$
$$x \equiv 5 \ mod \ 54 \qquad \Rightarrow x \equiv 5 \ mod \ 9$$

Das geht nicht gleichzeitig!

Einheit $\overline{a} \in \mathbb{Z}/m\mathbb{Z}$, also ein Element von $(\mathbb{Z}/m\mathbb{Z})^*$[3]:
$(\mathbb{Z}/m\mathbb{Z})^* = \{\overline{a} : 0 \leq a \leq m-1, ggT(a, m) = 1\}$

Eulersche φ-Funktion Definition - Seite 17, Eigenschaften - Seite 18 bzw. Seite 42

3.8 Der chinesische Restsatz

Seite 18ff.
Satz: (Zusammenfassung von Kongruenzgleichungen)
Seien $m_1, \ldots, m_r$ paarweise teilerfremd, $a_1, \ldots, a_r \in \mathbb{Z}$ und $a \in \mathbb{Z}$ mit $a \equiv a_i \ mod \ m_i$ für $i = 1, \ldots, r$ (wie im chinesischen Restsatz). Für $x \in \mathbb{Z}$ sind äquivalent: $x \equiv a_i \ mod \ m_i \Leftrightarrow x \equiv a \ mod \ m_1 \ldots m_r$

Beweis:
"$\Rightarrow$"
Sei $x \in \mathbb{Z}$ mit $x \equiv a_i \ mod \ m_i$ für $i = 1, \ldots, r$. Wegen $a \equiv a_i \ mod \ m_i$, folgt $x \equiv a_i \equiv a \ mod \ m_i, \quad \Rightarrow m_i | x - a, \quad \Rightarrow m_1 \ldots m_r | x - a, \quad \Rightarrow x \equiv a \ mod \ m_1 \ldots m_r$

"$\Leftarrow$"
$x \equiv a \ mod \ m_1 \ldots m_r \Rightarrow x \equiv a \ mod \ m_i \overset{a \equiv a_i \ mod \ m_i}{\Rightarrow} x \equiv a_i \ mod \ m_i.$

Beweisskizze für $\phi(mn) = \phi(m)\phi(n)$ im Fall $ggT(m, n) = 1$.

[3] Nochmal zur Erinnerung: $\mathbb{Z}/m\mathbb{Z}$ bezeichnet den Restklassenring modulo m

Man zeigt, dass die Abbildung

$$\{0 \leq a \leq mn-1 : ggT(mn,a) = 1\} \to$$
$$\{0 \leq b \leq m-1 : ggT(m,b) = 1\} \times \{0 \leq c \leq n-1 : ggT(n,c) = 1\}$$
$$a \to (a \ mod \ m, a \ mod \ n)$$

wohldefiniert und injektiv ist; der chinesiche Restsatz liefert, dass sie surjektiv ist. Da die Abbildung also bijektiv ist, sind die Mächtigkeiten von Definitionsmenge und Bild gleich, ergo $\phi(mn) = \phi(m)\phi(n)$.

In der Algebra taucht der chinesische Restsatz manchmal in folgender Form auf:
Satz:

Seien $m_1, \ldots, m_r \in \mathbb{N}$ paarweise teilerfremd.
Dann definiert

$$\mathbb{Z}/_{m_1 \ldots m_r \mathbb{Z}} \to \mathbb{Z}/_{m_1 \mathbb{Z}} \times \ldots \times \mathbb{Z}/_{m_r \mathbb{Z}}$$

einen Ringisomorphismus.

Beispiel:

$\mathbb{Z}/_{60\mathbb{Z}} = \mathbb{Z}/_{(4\ 3\ 5)\mathbb{Z}} \cdots$

4 Primzahltests

Kapitel 4

4.1 Kleine Teiler

Seite 1f.

4.2 Die Sätze von Fermat und Euler

Seite 2ff.

Die Ordnung einer Gruppe $G = (\mathbb{Z}/n\mathbb{Z})^*$ wird repräsentiert durch $\#(\mathbb{Z}/n\mathbb{Z})^*$

Lemma:
Für eine Primzahl p und $a, b, \in \mathbb{Z}$ gilt $(a+b)^p \equiv a^p + b^p \ mod\ p$

Kleiner Satz von Fermat:
Für eine Primzahl p und $a \in \mathbb{Z}$ gelten die folgenden zwei Aussagen, sowie die dritte unmittelbar hieraus gefolgerte:

- $a^p \equiv a \ mod\ p$
- $gcd(a,p) = 1 \Rightarrow a^{p-1} \equiv 1 \ mod\ p$
- $a^{n-1} \not\equiv 1 \ mod\ n \Rightarrow n$ zusammengesetzt

Satz von Euler:
Seien $n \in \mathbb{N}$ und $a \in \mathbb{Z}$ gegeben mit $gcd(a,n) = 1$, so gilt $a^{\varphi(n)} \equiv 1 \ mod\ n$

4.3 Schnelles Potenzieren - Die square-and-multiply-Methode

Seite 4ff.

4.4 Der Fermatsche Primzahltest

Seite 6ff.
Definitionen einer *Pseudoprimzahl* (Seite 7) sowie einer *wahrscheinlichen Primzahl* (Seite 8) und insbesondere (Seite 11).

zusätzliche Bemerkung:

(2) Es gilt $\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{log\ x}} = 1$, d.h. $\pi(x) \sim \frac{x}{log\ x}$ (Primzahlsatz)
Man kann zeigen: $\lim_{x\to\infty} \frac{\pi_{F,a}(x)}{\pi(x)} = 0$
Im Verhältnis zu Primzahlen gibt es wenige Pseudoprimzahlen.

(3) Überlegung: n bestehe den Fermat-Test zur Basis a, d.h. $a^{n-1} \equiv 1 \; mod \; n$
Dann gibt es zwei Möglichkeiten:

i) n prim

ii) n Fermat-Pseudoprimzahl zur Basis a

$\Rightarrow$ Fall ii) ist unwahrscheinlich wegen (2). n ist also wahrscheinlich prim.

$\Rightarrow$ (2) Skript-Bemerkung der Seite 8

Eigenschaften der Carmicheael-Zahlen: Seite 1f.

- sind quadratfrei, d.h. $p|n \Rightarrow p^2 \nmid n$, haben mindestens drei Primteiler und sind ungerade.
- Es gibt unendlich viele Carmichael-Zahlen
- n Carmichael-Zahl $\Leftrightarrow \forall a \in \mathbb{Z}. \quad a^n \equiv a \; mod \; n$

Satz (Korselt-Kriterium): Seite 1f.
$n \in \mathbb{N}$ ist genau dann eine Carmichael-Zahl, wenn gilt:

1. n zusammengesetzt
2. n quadratfrei
3. $p|n \Rightarrow p-1|n-1$ für alle Primteiler p von n

Beispiel:
$n = 561 = 3 \cdot 11 \cdot 17$, 1, 2 ✓
3: $2|560, \quad 10|560, \quad 16|560$ ✓

Folgerung:
Sei $n \in \mathbb{N}, \quad p_1 = 6u+1, \quad p_2 = 12u+1, \quad p_3 = 18u+1$
$n = p_1 p_2 p_3 = 1296u^3 + 396u^2 + 36u + 1$
Sind p_1, p_2, p_3 Primzahlen, so ist n eine Carmichael-Zahl.

Beweis:
Seien $p_1 p_2 p_3$ Primzahlen.
Korselt-Kriterium: 1, 2 ✓
3:

$$n - 1 = 2^4 \cdot 3^4 \cdot u^3 + 2^2 \cdot 3^2 \cdot 11 \cdot u^2 + 2^2 \cdot 3^2 \cdot u$$
$$p_1 - 1 = 6u, \quad p_2 - 1 = 12u, \quad p_3 - 1 = 18u$$
$$p_1 - 1|n-1, \quad p_2 - 1|n-1, \quad p_3 - 1|n-1. \quad ✓$$

Korselt-Kriterium ist erfüllt, also handelt es sich bei n um eine Carmichael-Zahl.

Mit der Folgerung kann man praktisch leicht Carmichael-Zahlen konstruieren:
$u = 1, \quad p_1 = 7, \quad p_2 = 13, \quad p_1 9, \quad n = 1729 \qquad u = 10^{100} + 289351$
und die zugehörigen p_1, p_2, p_3 sind prim, also das zugehörige n eine Carmichael-Zahl.

4.5 Der Miller-Rabin-Test

Seite 9ff.

Lemma:

Seien p eine Primzahl, a eine Zahl mit $a^2 \equiv 1 \ mod \ p$ und $a \not\equiv 1 \ mod \ p \Rightarrow a \equiv -1 \ mod \ p$

Lemma:

p ungerade prim, $p-1 = 2^l q, \quad q \equiv 1 \ mod \ 2 \quad$ und $\quad b \equiv a^q \ mod \ p, \quad gcd(a,p) = 1$
$\Rightarrow b \equiv 1 \ mod \ p \ \vee \ \left(\exists i \in \{0, \ldots, l-1\}. \quad b^{2^i} \equiv -1 \ mod \ p\right)$

Definition *starke Pseudoprimzahl zur Basis a* Seite 10.
Erweiterte Riemansche Vermutung[4] auf Seite 12.

[4] Rieman Vermutung macht Aussagen über NS der meromorphe Fortsetzung von $I(s) = \sum_{n \geq 1} \frac{1}{n^s}, \quad s \in \mathbb{C}$

5 Public-Key-Kryptosysteme

Kapitel 5

5.1 Erinnerung

Seite 1

5.2 Eine Idee

Seite 1f
Dieses Unterkapitel erklärt im Grunde die Idee und die notwendigen Eigenschaften der asymmetrischen Verschlüsselungsverfahren.

5.3 RSA

Seite 2ff

5.3.1 Mathematische Grundlagen

Seite 2
$p, q \in \mathbb{P}, p \neq q, \quad N = pq, \quad ed \equiv 1 \ mod \ (p-1)(q-1), e, d \in \mathbb{N} \quad a, b \in \mathbb{Z}$

- $b \equiv a^e \ mod \ N \Rightarrow a \equiv b^d \ mod \ N$
- $\forall a \in \mathbb{Z}. \quad a^{ed} \equiv a \ mod \ N$

5.3.2 Das RSA-Kryptosystem

Seite 2ff.

- $e_A > 1, gcd(e_A, (p_A - 1)(q_A - 1)) = 1$
- public key: (N_A, e_A), private key: (N_A, d_A)
- $b_i = a_i^{e_A} \ mod \ N_A$ (B sendet an A)
- $a_i = b_i^{d_A} \ mod \ N_A$

Beispiele auf Seite 4f

5.4 Ein Angriff auf RSA mit dem chinesischen Restsatz

Seite 5f

5.5 Faktorisierung einer RSA-Zahl N bei Kenntnis des öffentlichen und privaten Schlüssels

Seite 6ff
Der Pseudocode für die Anwendung unter bestimmten Gegebenheiten ist auf Seite 8 beschrieben.

5.6 Die Fermatsche Faktorisierungsmethode

Seite 8ff
Die ungerade zu faktorisierende natürliche Zahl N wird als Differenz zweier Quadratzahlen dargestellt.
Der Fall $N = pq$ als RSA-Zahl wird auf Seite 10f behandelt.
Seite 11ff betrachtet bis zum Ende des Unterkapitels die Laufzeiten dieses Verfahrens.

RSA-Zahlen sind mit der Fermatschen Faktorisierungsmethode günstig zu faktorisieren, wenn $\Delta = |p - q| \lessapprox N^{\frac{1}{4}}$ und ungünstig, wenn $q \geq \lambda p, \lambda > 1$. Für große RSA-Zahlen genügt $q \geq 1.1p$.

6 Die Pollardsche rho-Methode zur Faktorisierung

Kapitel 6 entfällt aus Zeitgründen.

7 Kryptographische Anwendungen diskreter Logarithmen

Kapitel 7

Vorbemerkung:

Die Sicherheit von RSA beruht darauf, dass man praktisch ganz schlecht faktorisieren kann, dass man andererseits leicht und schnell potenzieren kann, (wahrscheinliche) Primzahlen erzeugen kann, ggT und $ax \equiv m$ berechnen kann, ...

Frage:

Gibt es auch andere Stellen in der Mathematik, wo Ähnliches passiert?

7.1 Die multiplikative Ordnung einer Zahl $a \in \mathbb{Z}\ mod\ m \in \mathbb{N}$

Seite 1

Ist $a \in \mathbb{Z}$ mit $ggT(a,m) = 1$, so gilt $a^{\varphi(m)} \equiv 1\ mod\ m$. (Satz von Euler)

Daher ist folgende Definition sinnvoll:

$ord_m(a) = min\{n \in \mathbb{N} : a^n \equiv 1\ mod\ m\}$ (Ordnung von $a\ mod\ m$)

Beispiele:

1. $m = 5, a = 2;$ $2^1 \equiv 2\ mod\ 5,$ $2^2 \equiv 4\ mod\ 5,$ $2^3 \equiv 3\ mod\ 5,$ $2^4 \equiv 1\ mod\ 5$
   $\Rightarrow ord_5(2) = 4$

2. $m = 8, a = 5;$ $5^1 \equiv 1\ mod\ 8,$ $5^2 \equiv 1\ mod\ 8$
   $\Rightarrow ord_8(5) = 2$

Lemma:

$m \in \mathbb{N}, a \in \mathbb{Z}$ mit $ggT(a,m) = 1$

1. Für $n \in \mathbb{N}_0$ sind äquivalent: $a^n \equiv 1\ mod\ m \Leftrightarrow ord_m(a)|n$
2. Für $n_1, n_2 \in \mathbb{N}_0$ sind äquivalent: $a^{n_1} \equiv a^{n_2}\ mod\ m \Leftrightarrow n_1 \equiv n_2\ mod\ ord_m(a)$

Beweis:

1. "$\Rightarrow$"
   Teile n durch $ord_m(a)$ mit Rest r, $n = k \cdot ord_m(a) + r$ mit $k, r \in \mathbb{N}_0$ und $0 \leq r \leq ord_m(a) - 1$.

Daher $1 \equiv a^n \equiv a^{k \cdot ord_m(a)+r} \equiv \left(a^{ord_m(a)}\right)^k \cdot a^r \equiv a^r \ mod\ m$.
Aus $0 \leq r < ord_m(a)$ folgt $r = 0$, also $n = k \cdot ord_m(a)$, d.h. $ord_m(a)|n$

"$\Leftarrow$"

$ord_m(a)|n \Rightarrow n = k \cdot ord_m(a)$ für ein $k \in \mathbb{N}_0$, also $\Rightarrow a^n \equiv \left(\underbrace{a^{ord_m(a)}}_{\equiv 1}\right)^k \equiv 1 \ mod\ m$

2. O.E. $n_2 \geq n_1, n_2 = n_1 + n$. Dann $a^{n_1} \equiv a^{n_2} \ mod\ m \Leftrightarrow \underbrace{a^{n_1}}_{\text{teilerfremd zu } m} \equiv a^{n_1} \cdot a^n \ mod\ m \Leftrightarrow 1 \equiv a^n \ mod\ m \overset{1}{\Leftrightarrow} ord_m(a)|n \Leftrightarrow ord_m(a)|n_2 - n_1 \Leftrightarrow n_1 \equiv n_2 \ mod\ ord_m(a)$

Insbesondere: $n_1 \equiv n_2 \ mod\ ord_m(a) \Rightarrow a^{n_1} \equiv a^{n_2} \ mod\ m$, d.h. im Exponenten von a darf man $mod\ ord_m(a)$ rechnen. Was weiß man über die Ordnung $ord_m(a)$?

1. $ord_m(a)|\varphi(m)$ (Folgt aus $a^{\varphi(m)} \equiv 1 \ mod\ m$)
2. Ist p Primzahl, so gilt $ord_p(a)|p-1$ (Folgt aus kleinem Satz von Fermat $a^{p-1} \equiv 1 \ mod\ m$)

Aus 2 folgt: $ord_p(a) \leq p-1$
$a \in \mathbb{Z}$ heißt Primitivwurzel modulo p, wenn $ord_p(a) = p-1$ ist.

Satz:
Zu jeder Primzahl p gibt es Primitivwurzeln modulo p.
Beispiele:
g_p sei kleinste Primitivwurzel modulo p mit $g_p \in \mathbb{N}$

p	2	3	5	7	11	13	17	19	23	29	31	37	41
g_p	1	2	2	3	2	2	3	2	5	2	3	2	6

7.2 Die multiplikative Gruppe des Körpers F_p - Diskrete Logarithmen[5]

Seite 1ff.
Naive Methode zur Berechnung diskreter Logarithmen Seite 2.

- gesucht: $log_g(a) \ mod\ p$
- $2 \leq g, \quad a \leq p-1$
- für $x \in \{0, \ldots p-1\}$ ausprobieren, ob $g^x \equiv a \ mod\ p$ gilt. $min\{x|g^x \equiv a \ mod\ p\}$
  - $g^x = a \Rightarrow x = log_g(a) \ mod\ p$

[5] F_p bezeichnet hierbei eine andere Schreibweise für $\mathbb{Z}/p\mathbb{Z}$, wobei p eine Primzahl

- $g^x = 1 \Rightarrow$ es existiert keine Lösung

Bestimmung der Ordnung eines Elements aus F_p^* Seite 3.

7.3 Schlüsselaustausch nach Diffie-Hellmann

Seite 4

- $p \in \mathbb{P}, \quad g \in \{2, \ldots, p-2\}, \quad e_A, e_B \in \{0, \ldots, p-2\}$
- $f_A = g^{e_A} \ mod \ p, \quad f_B = g^{e_B} \ mod \ p$
- $k_{AB} = g^{e_A e_B} \ mod \ p = f_B^{e_A} \ mod \ p = f_A^{e_B} \ mod \ p$

7.4 Die ElGamal-Verschlüsselung

Seite 5f.
Das Verfahren selbst ist nicht weiter spannend. Der interessante Teil beschränkt sich auf die Verwendung des DH-Austauschs.

Wichtig für die *Sicherheit der ElGamal-Verschlüsselung* ist insbesondere die erste Überlegung auf Seite 5, die nicht 100% äquivalent zu der Bemerkung auf Seite 6f. ist.

- public key: (p_A, g_A, f_A), $f_A = g_A^{e_A} \ mod \ p_A$,
  Zufallszahl $z_i \in \{0, \ldots, p_A - 2\} \ni e_A, \quad g_A \in \{2, \ldots, p-2\}$
- $b_i = g_A^{z_i} \ mod \ p_A, \quad k_i = f_A^{z_i} \ mod \ p_A = b_i^{e_A} \ mod \ p_A, \quad c_i = a_i \cdot k_i \ mod \ p_A$
- Chiffretext: (b_i, c_i)
- $a_i = c_i \cdot b_i^{-e_A} \ mod \ p_A = b_i^{p_A - 1 - e_A} \cdot c_i \ mod \ p_A$

7.5 Massey-Omura-Verschlüsselung

Seite 7f.
Hierbei dürfte es sich m.E. wohl um ein lediglich für die Theorie interessantes Verfahren handeln, da Mechanismen eingesetzt werden müssten, um MITM-Angriffen vorzubeugen, welche sämtliche Vorteile gegenüber üblich praktizierten Verfahren zunichte machen.

- $p \in \mathbb{P}, \quad gcd(e_A, p-1) = 1, \quad d_A e_A \equiv 1 \ mod \ p-1, \quad e_A, d_A$ beide geheim!
- $b_i = a^{e_A} \ mod \ p$
- $c_i = b_i^{e_B} \ mod \ p$
- $d_i = c_i^{d_A} \ mod \ p$
- $e_i = d_i^{d_B} \ mod \ p = a_i \ mod \ p$

8 Kryptographische Hashfunktionen

Kapitel 8 Dieses Kapitel scheint lediglich erwähnt worden zu sein.

9 Digitale Signaturen

Kapitel 9

9.1 Einführung

Seite 1 Eigenschaften von Signaturen.

9.2 Allgemeine Verfahren

Seite 1f.
Unterschreiben mit einem Public-Key-Datenverschlüsselungssystem.
Unterschreiben mit einem Public-Key-Verfahren unter Verwendung einer Hash-Funktion.
Digitale Signatur mit Verschlüsselung.

9.3 Die RSA-Signatur

Seite 3

(1) Schlüsselerzeugung:
   a) A konstruiert sich ein RSA-Schlüsselpaar: (N_A, e_A) öffentlich, (N_A, d_A) geheim ($y \equiv x^{e_A} \ mod \ N_A \Leftrightarrow x \equiv y^{d_A} \ mod \ N_A$)
   b) A legt sich auf eine Hashfunktion h fest. (Praktisch: Die Hashwerte sollten $< N_A$ sein)

(2) Unterschreiben/Signieren eines Dokuments M:
   a) A berechnet den Hashwert $h(M)$
   b) Mit seinem privaten Schlüssel (N_A, d_A) berechnet A die Zahl $\boxed{s = h(M)^{d_A} \ mod \ N_A}$ Die Zahl s ist die RSA-Signatur von A für das Dokument M

(3) Signaturüberprüfung: Wie testet man, ob eine Signatur s für M von A stammt?
   a) Man berechnet den Hashwert $h(M)$ und mit dem öffentlichen Schlüssel (N_A, e_A) von A die Zahl $\boxed{h' = s^{e_A} \ mod \ N_A}$.
   b) Gilt $h' = h(M)$, so wird die Signatur als gültig anerkannt, andernfalls nicht $\boxed{s \equiv h(M)^{d_A} \ mod \ N_A \Rightarrow h(M) \equiv s^{e_A} \ mod \ N_A}$

9.4 Das ElGamal-Signatur-Verfahren

Seite 3ff.
Auch hier findet das DH-Verfahren wieder seine Verwendung. Daher ist das Prinzip ganz ähnlich der ElGamal-Verschlüsselung.

- $ggT(z, p-1) = 1$

- $c \equiv z^{-1}(h - be) \ mod \ p - 1$ (2)

Zu (3), der *Signaturprüfung*: Warum muss man $1 \leq b \leq p_A - 1$ überprüfen?
Weil man sonst Unterschriften fälschen kann! Sei M ein Dokument mit dem Hashwert $h(M)$ und (p_A, g_A, f_A) der öffentliche Schlüssel von A. Sei weiterhin $b = (p_A - 1) \cdot (p_A - g_A), \quad c = h(M)$. Es ist $b \equiv (-1) \cdot (-g_A) \equiv g_A \ mod \ p_A$ und $b \equiv 0 \ mod \ (p_A - 1)$ (Im Exponenten rechnet man $mod \ p_A - 1$, in der Basis $mod \ p_A$) Dann: $f_A^b \cdot b^c \equiv f_A^b \cdot g_A^c \equiv f_A^0 \cdot g_A^c \equiv g_A^{h(M)} \ mod \ p_A$ d.h. (b, c) erfüllt die zweite Bedingung der Signaturüberprüfung, obwohl (b, c) nicht von A stammt. Wegen $b > p_A - 1$ ist die Bedingung $1 \leq b \leq p_A - 1$ wichtig.

Überlegung zur Sicherheit: Öffentlicher Schlüssel: (p, g, f), privat e mit $f = g^e \ mod \ p$

(2) Sei (b, c) gültige Signatur, dann gilt: $\boxed{eb + zc \equiv h \ mod \ (p - 1)}$ und $\boxed{b = g^z \ mod \ p}$.

a) Könnte man diskrete Logarithmen berechnen, so könnte man z bestimmen und dann aus der Kongruenzgleichung e

b) Sonst hat man eine Gleichung mit zwei Unbekannten e, z, was ungenügend ist.

c) z muss auf jeden Fall geheim bleiben, sonst eine Gleichung, eine Unbekannte $e \Rightarrow$ lösbar.

d) Selbiges gilt bei Mehrfachverwendung gleicher Zufallszahlen.

Folglich braucht man einen guten Zufallsgenerator

(3) Hier lediglich der Beweis dafür, dass die Zufallszahlen z_1, z_2 identisch sind, der Rest steht im offiziellen Skript. Angenommen, wir erhalten zwei Dokumente mit den Hashwerten h_1, h_2, den Signaturen $(b_1, c_1), (b_2, c_2)$ und es gilt $b_1 = b_2$

a) Es ist $b_1 \equiv g^{z_1} \ mod \ p, \quad b_2 \equiv g^{z_2} \ mod \ p$, also, da g Primitivwurzel $mod \ p$ ist, folgt aus $g^{z_1} \equiv g^{z_2} \ mod \ p$ sofort $z_1 \equiv z_2 \ mod (p - 1)$, also o.E. $z = z_1 = z_2$

b) vgl. Skript Seite 5

(4) Rest siehe Skript.

Satz:

$a, b \in \mathbb{Z}, m \in \mathbb{N}. (u, v \in \mathbb{Z}$ mit $au + mv = gcd(a, m))$

1. $ax \equiv b \ mod \ m$ lösbar $\Leftrightarrow gcd(a, m) | b$
2. Sei $gcd(a, m) | b$ erfüllt, seien weiterhin $x_0 = \frac{b}{gcd(a,u)} u$ und $x_i = x_0 + i \cdot \frac{m}{gcd(a,m)}$ für $i = 0, \ldots, gcd(a, m) - 1$. Dann sind $x_0, \ldots, x_{gcd(a,m)-1}$ alle Lösungen von $ax \equiv b \ mod \ n$ (und paarweise verschieden).

In gpg Version 1.2.3, konnte man die ElGamal-Signatur verwenden. Damit das Signieren schneller geht, wurde aber e_A und die Zufallszahl z deutlich kleiner als p gewählt, natürlich nicht so klein, dass man die Logarithmenberechnung erhalten konnte ($b = g_A^z \; mod \; _A, f_A = g_A^{e_A} \; mod \, p_A$) Nguyen fand, dass man dann aus $cz+be_A \equiv h(M) \; mod \; (p_A-1)$ mit Gitermethoden e_A und z schnell berechnen kann: $cz + be_A \equiv h(M) \; mod \; (p_A - 1)$.
Gilt $g_A | p_A - 1$, so ist die ElGamal-Signatur nicht sicher.

Hier die Beweisidee des ersten Satz des Anhangs *Unsicherheit bei ElGamal-Signatur* Seite 11f..
Fälle der Beweisidee:

$f^b \equiv 1 \; mod \; p$:
$f^b b^c \equiv g^h \mod p \Leftrightarrow b^c \equiv 2^h \; mod \; p \Leftrightarrow 2^{\frac{p-3}{2} \cdot c} \equiv 2^h \; mod \; p \Leftrightarrow \frac{p-3}{2} \cdot 2 \equiv h \; mod \; ord_p(2) \overset{2 \text{ Primitivwurzel } mod \; p}{\Leftrightarrow} \frac{p-3}{2} \cdot c \equiv h \; mod \; (p-1)$ Dies ist eine Gleichung des Typs $ax \equiv b \; mod \; m$, die man gut lösen kann.

$f^b \equiv -1 \; mod \; p$:
Analog

Es gibt verschiedene Weiterentwicklungen der ElGamal-Signatur, die auch praktisch von Bedeutung sind: DSA und ECDSA . Die Frage: Wie kommt man von der ElGamal-Signatur zu DSA wird im offiziellen Skript behandeln.

Wie kann man die ElGamal-Signatur verallgemeinern?

(1) Ein Problem: b kommt in der Basis vor (Rechnen $mod \; p_A$), aber auch im Exponenten (Rechnen $mod \; (p_A - 1)$). Bei der Basis rechnen wir in der Gruppe $(\mathbb{Z}/p_A\mathbb{Z})^*$. Wir definieren $l : (\mathbb{Z}/p_A\mathbb{Z})^* \to \mathbb{Z}$ mit $\overline{x} \mapsto x$ mit $0 \leq x \leq p-1$ (Auswahl eines Repräsentanten). Die Gleichungen schreiben sich nun $\underbrace{b = g^z \; mod \; p_A}_{\text{Rechnen in } (\mathbb{Z}/p_A\mathbb{Z})^*}$;

$\underbrace{c = \frac{1}{z}(h(M) - l(b)e_A) \; mod \; (p_A - 1)}_{\text{Rechnen mit ganzen Zahlen}}; \quad \underbrace{f_A^{l(b)} b^c \equiv g_A^{h(M)} \; mod \; p_A}_{\text{Rechnen in } (\mathbb{Z}/p_A\mathbb{Z})^*}$

(2) Wir versuchen $(\mathbb{Z}/p_A\mathbb{Z})^*$ durch eine Gruppe zu ersetzen:

Sei G multiplikativ geschriebene Gruppe (Bsp.: $Q^*, Gh_n(Q), \ldots$) $g \in G$ mit $ord_g(g) = n, n \in \mathbb{N}$. Wähle $e_A \in \mathbb{N}$ geheim, $f_A = g^{e_A} \in G$ öffentlich. (G, g, f_A)

Unterschreiben eine Nachricht M mit Hashwert $h(M)$:

Wähle Zufallszahl z mit $gcd(z, n) = 1$, berechne $b = g^z \in G$,
$c = \frac{1}{z} \underbrace{(h(M) - l(b)e_A)}_{zc = h(M) - l(b)e_A \; mod \; n} \; mod \; n.$

Signatur:

$$(b, c) \in G \times \mathbb{Z}$$

Signatur-Test:

$$f_A^{l(b)} b^c = (g^{e_A})^{l(b)} (g^z)^c = g^{e_A l(b) + zc} \overset{ord(g)=n}{=} g^{h(M)}$$

(3) Man kann (2) auch mit additiver Gruppe $(G, +)$ aufschreiben. Ein Beispiel sind elliptische Kurven über endliche Körper $\rightarrow$ ECDSA

9.5 DSA - Digital Signature Algorithm

Seite 6ff.
Inhaltlich findet man die Tafelanschrift prinzipiell in dem Handout und dem Skript.

- $q, p \in \mathbb{P}, q \in \{2^{n-1} - 1, \ldots, 2^n + 1\}, \quad p \equiv 1 \ mod \ q, p \in \{2^{l-1} - 1, \ldots, 2^l + 1\}$
- bestimme $g \in \{2, \ldots, p-1\}, ord_p(g) = q$
- wähle Zufallszahl $z \in \{1, \ldots, q-1\} \ni e_A$
- $f_A = g^{e_A} \ mod \ p_A$
- public key: (f_A, p, q), private key: e_A
- $r = (g^z \ mod \ p) \ mod \ q$
- $s \equiv z^{-1} \cdot (h + e_A r) \ mod \ q, r \neq 0 \neq s$
- Signatur (r, s)