Dr. Dirk Frettlöh 24.4.2019

Übungen zur Vorlesung Kryptographie

Blatt 4

Aufgabe 13: (Genug Primzahlen?)
Die Aufgabe ist, zu bestimmen, wieviele 16-bit-Primzahlen es gibt, und wieviele 32-bit-Primzahlen. Genauer:

(a) Bestimmen Sie mit dem Primzahlsatz, wieviele Primzahlen es zwischen 2^{k-1} und 2^k geben sollte, jeweils für $k = 16$ und $k = 32$.
(b) Schreiben Sie ein Computerprogramm das zählt, wieviele Primzahlen es zwischen 2^{k-1} und 2^k wirklich gibt, jeweils für $k = 16$ und $k = 32$.

Aufgabe 14: (Zwei ist ein Lügner / Alle sind Lügner)
(a) Finden Sie drei Zahlen $n \in \mathbb{N} \setminus \{1\}$, die keine Primzahlen sind, und für die dennoch gilt: $2^{n-1} \equiv 1 \bmod n$.
(b) Finden Sie drei Zahlen $n \in \mathbb{N} \setminus \{1\}$, die keine Primzahlen sind, so dass dennoch für alle $a \in \mathbb{N}$ mit $2 \leq a \leq n-1$ gilt: Falls $\mathrm{ggT}(a, n) = 1$, so ist $a^{n-1} \equiv 1 \bmod n$.

Wer (b) löst, braucht (a) nicht zu lösen. Ergoogelte Lösungen zählen aber nicht.

Aufgabe 15: (Wieviele Lügner?)
Sei n keine Primzahl. Eine Zahl a heißt *Fermat-Lügner* mod n, falls $\mathrm{ggT}(a, n) = 1$ und $a^{n-1} \equiv 1 \bmod n$.

(a) Finden Sie alle Fermat-Lügner mod 15.
(b) Zeigen Sie: sind p und $2p-1$ beides Primzahlen, sowie $n = p(2p-1)$, so sind exakt die Hälfte der Elemente in Z_n^* Fermat-Lügner (und zwar alle, die quadratischer Rest mod $2p-1$ sind).

Aufgabe 16: (Fibonacciprimzahltest)
Seien f_n die Fibonaccizahlen, also $f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$ für $n \geq 2$. Zeigen Sie: eine Fibonaccizahl f_n kann nur prim sein, falls n prim ist (mit einer Ausnahme). Genauer:

(a) Zeigen Sie: Für alle $m, n \in \mathbb{N}$ (also $m \geq 0$, $n \geq 0$) gilt: $f_{m+n} = f_{m-1}f_n + f_m f_{n+1}$.
(b) Zeigen Sie: f_{mn} ist durch f_m teilbar. Wieso ist nun die Behauptung bewiesen?
(c) Könnte man daraus einen guten (probabilistischen?) Primzahltest konstruieren? Wenn ja, wie? Wenn nein, warum nicht?
(d) Welches n ist die Ausnahme? (Also: n nicht prim, aber f_n doch.)
(e) Finden Sie drei "Lügner" für diesen Kontext, also drei Primzahlen n, so dass f_n nicht prim ist.

Viele Aufgaben dürfen auch mit dem Computer gelöst werden. Falls Sie Programmcode abgeben, beschreiben Sie auf jeden Fall auch, wie Sie vorgingen.

Abgabe: Mittwoch 8.5.2019 bis 14 Uhr in Postfach 2183 in V3, oder per Email an den Tutor.
Bitte auf jeder Abgabe das Tutorium angeben! (Di oder Do)

Dienstag	Philipp Braukmann	pbraukmann@techfak.uni-bielefeld.de
Donnerstag	Oliver Tautz	otautz@techfak.uni-bielefeld.de