RUHR UNIVERSITÄT BOCHUM

RUB

Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Gottfried Herold

Hausübungen zur Vorlesung

Zahlentheorie

SS 2013

Blatt 11 / 21. Juni 2013 / Abgabe bis spätestens 1. Juli 2013, 12:00 Uhr in dem Kasten auf NA 02 oder am Anfang der Vorlesung

Geben Sie bitte die Aufgaben zur Vereinfachung der Korrektur folgendermassen nach Aufgaben getrennt ab:

- Aufgabe 1 in Kasten A
- Aufgabe 2 in Kasten B
- Aufgaben 3,4 in Kasten C

Die Kästen auf NA 02 sind entsprechend beschriftet. Wenn Sie in der Vorlesung abgeben, machen sie einfach 3 getrennte Stapel. Schreiben Sie auf alle 3 Abgaben jeweils Ihre(n) Namen und/oder Matrikelnummer(n).
Bitte schreiben Sie auf Ihre Abgaben eine Sollrückgabestelle (Übungsgruppe, Zentralübung, persönlich in NA5/74).

AUFGABE 1 (4 Punkte)**:**
Sei p Primzahl und $n = p^k$ Primzahlpotenz mit $k > 1$. Zeigen Sie dass es dann a mit $\mathrm{ggT}(a, n) = 1$ gibt mit

$$(X + a)^n \not\equiv X^n + a \mod n$$

im Polynomring $\mathbb{Z}[X]$.

Hinweis: Betrachten Sie den Koeffizienten von X^p.

AUFGABE 2 (5 Punkte)**:**
Sei $n > 3$ zusammengesetzte Zahl. Nehmen wir an wir kennen vier paarweise verschiedene Zahlen $1 \leq a_1, a_2, a_3, a_4 < n$, für die gilt:

$$a_1^3 \equiv a_2^3 \equiv a_3^3 \equiv a_4^3 \mod n$$

Geben Sie einen effizienten Algorithmus an, der (bei Eingabe n, a_1, a_2, a_3, a_4 mit obiger Eigenschaft) einen nicht-trivialen Faktor von n liefert.
Effizient soll dabei heissen, dass die Laufzeit höchstens $\mathcal{O}(\log(n)^k)$ für ein $k \in \mathbb{N}$ ist.
Beweisen Sie, dass Ihr Algorithmus sowohl korrekt ist (d.h. wirklich einen nicht-trivialen Faktor liefert) und effizient ist.

– bitte wenden –

AUFGABE 3 (5 Punkte)**:**
Faktorisieren Sie die zusammengesetzte Zahl $n = 7519$ mit Hilfe der Fermat-Faktorisierungsmethode.

AUFGABE 4 (6 Punkte)**:**
Faktorisieren Sie die Zahl $n = 7519$ mit Hilfe des Morrison-Brillhart Kettenbruch-Algorithmus.
Wählen Sie als Faktorbasis $B = \{-1, 2, 3, 5\}$.

Hinweis: Sie benötigen nicht unbedigt 4 B-glatte Zahlen. Es genügt, die Kettenbruchentwicklung $\sqrt{n} = [a_0, a_1, a_2, a_3, \ldots]$ bis a_3 zu berechnen.