

Hausübungen zur Vorlesung

Kryptanalyse

WS 2012/2013

Blatt 2 / 23. Oktober 2012 / Abgabe bis spätestens 30. Oktober 2012, 8:30
Uhr in dem Kasten auf NA 02

AUFGABE 1 (5 Punkte):

Alice feiert eine weitere Party und schickt eine Einladung m an Bob und Berta (Birte ist im Urlaub). Dabei verschlüsselt Alice m mit den öffentlichen RSA-Schlüsseln (N, e_1) und (N, e_2) von Bob und Berta, wobei e_1 und e_2 teilerfremd sind.

Wieder ist Eve nicht zur Party eingeladen. Helfen Sie Eve auch diesmal und zeigen Sie, dass man aus den Chiffretexten die Einladung m effizient berechnen kann.

AUFGABE 2 (10 Punkte):

Sei $f(x)$ ein Polynom vom Grad $n-1$ wobei n eine Zweierpotenz ist. Im Folgenden bezeichne mod das modulo auf dem Ring der Polynome in x . Wir setzen voraus, dass man $g(x) \text{ mod } h(x)$ für beliebige Polynome des Grades höchstens $n-1$ in Zeit $O(n \log n)$ berechnen kann.

Gegeben seien $f(x)$ und n Stellen x_0, \dots, x_{n-1} . Zu berechnen ist $f(x_i)$ für $i = 0, \dots, n-1$. Für $0 \leq i \leq j \leq n-1$ definieren wir $p_{ij}(x) = \prod_{m=i}^j (x - x_m)$ und $q_{ij}(x) = f(x) \text{ mod } p_{ij}(x)$.

- (a) Zeigen Sie, dass $f(x) \text{ mod } (x - z) = f(z)$ für alle z .
- (b) Zeigen Sie, dass $q_{kk}(x) = f(x_k)$ und $q_{0,n-1}(x) = f(x)$.
- (c) Zeigen Sie, dass für $i \leq k \leq j$ gilt $q_{ik}(x) = q_{ij}(x) \text{ mod } p_{ik}(x)$ und $q_{kj}(x) = q_{ij}(x) \text{ mod } p_{kj}(x)$.
- (d) Konstruieren Sie einen Algorithmus, der in Zeit $O(n \log^2 n)$ die Werte $f(x_0), \dots, f(x_{n-1})$ berechnet. Beweisen Sie *Korrektheit* und Laufzeit.

Hinweis: Realisieren Sie eine Divide-and-Conquer Strategie. Benutzen Sie (a) - (c).

AUFGABE 3 (5 Punkte):

Sei (N, e) ein öffentlicher RSA-Schlüssel. Ein *Fixpunkt* dieser RSA-Verschlüsselung ist ein $x \in \mathbb{Z}_N$ mit $x^e \equiv x \pmod{N}$. Sei $N = pq$ die Primfaktorzerlegung von N . Zeigen Sie, dass es genau

$$(\text{ggT}(e-1, p-1) + 1) \cdot (\text{ggT}(e-1, q-1) + 1)$$

Fixpunkte gibt.

AUFGABE 4 (5 Punkte):

Alice hat eine Einladung zu ihrer Geburtstagsparty an 17 Freunde verschickt. Diese besitzen die paarweise teilerfremden RSA-Moduln N_1, N_2, \dots, N_{17} und verwenden alle den öffentlichen Exponenten $e = 17$. Die Einladung wurde symmetrisch mit einem 80 Bit Schlüssel k verschlüsselt, welcher in einer (für alle Gäste identischen) Nachricht m asymmetrisch mit RSA verschlüsselt wurde. Die Nachricht m (Padding + Schlüssel) ist ein gültiger Klartext für alle Moduln, d.h. es gilt $m < \min\{N_1, \dots, N_{17}\}$. Eve ist nicht zur Party eingeladen, konnte aber die Chiffretexte c_1, \dots, c_{17} mitschneiden.

Implementieren Sie einen Broadcast-Angriff (siehe Präsenzübung Blatt 2, Aufgabe 2) in *sage*. Benutzen Sie hierzu die Dateien `N.sobj` und `c.sobj`¹. Sie können die Liste der Moduln beispielsweise durch den Befehl `N=load('N')` einlesen, mit `N[i]` können Sie sich dann den $(i-1)$ -ten Modul ausgeben lassen. Alternativ können Sie die Daten auch per Copy & Paste aus der Datei `crt.txt` einlesen. Wie lautet der Schlüssel k (die letzten 80 Bit von m)? Geben Sie den Quelltext Ihres Programms mit ab.

Hinweis: Sie können in *sage* mit der Funktion `crt` den verallgemeinerten Chinesischen Restsatz berechnen². Zur Berechnung von $\sqrt[n]{a}$ über den ganzen Zahlen können Sie z.B. mit `(x^n-a).roots()` alle Nullstellen der Funktion $f(x) = x^n - a$ bestimmen und die ganzzahlige Lösung herausfiltern. Zur Bestimmung der letzten 80 Bit von m können Sie mit `hex` die Nachricht hexadezimal darstellen.

¹Sie finden die zu den Implementierungsaufgaben gehörigen Daten (z.B. Public Keys oder abgefangene Chiffretexte) auf der Webseite zur Vorlesung <http://www.cits.rub.de/lehre/ws1213/kryptanal12.html>

²<http://www.sagemath.org/doc/reference/sage/rings/arith.html>