FGI-2 – Formale Grundlagen der Informatik II
Prozesse und Nebenläufigkeit

Aufgabenblatt 10 : Kryptographie

Abgabe am 15.1.2007 Besprechung am 17.1.2007.

Präsenzaufgabe 10 : RSA

Um die Faktorisierung $n = p \cdot q$ praktisch undurchführbar zu machen, sollen folgende drei Bedingungen erfüllt sein:

(1) $n > 10^{160}$

(2) p und q unterscheiden sich in der Länge ihrer Dezimalzahldarstellung um etliche Stellen.

(3) Weder p noch q sind klein, oder sind aus einer Primzahltafel genommen, oder sind von spezieller Form.

Geben Sie Gründe für diese Bedingungen an. Denken Sie bei (2) an die Formel $p \cdot q = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$.

Übungsaufgabe 10.1 :

a) Zeigen Sie, dass mit $(e, n) = (d, n) = (9, 55)$ ein RSA-Kyptosystem (im Sinne von Abschnitt 6.2.3) gegeben ist! (Geben Sie $p, q, \varphi(n)$ an, und prüfen Sie die zugehörigen Bedingungen!)

b) Verschlüsseln und entschlüsseln Sie „FGI“ für $F = 06, G = 07, I = 09$ (die Angabe für F genügt, jedoch Zwischenergebnisse notieren)!

c) Durch ungünstige Werte kann ein RSA-Schlüssel gebrochen werden. Beweisen Sie, dass p oder q gleich $ggt((a^{\frac{r}{2}} - 1) \ mod \ n, n)$, falls $a^r = 1 \ mod \ n$ für eine gerade natürliche Zahl $r > 0$ gilt! (Hinweis: Wenden Sie die 3. binomische Formel auf $a^r - 1 = 0$ an!)

d) Zeigen Sie c) für $a = 6$ in dem Beispiel unter a)!

von
5

Übungsaufgabe 10.2 :

Die folgende probabilistische Version von Quicksort (PQS) hat die *erwartete Zeitkomplexität* Exp-Time$_{PQS}(n) = O(n \cdot \log n)$ und ist deshalb effektiv.

a) Geben Sie einen Ablauf auf der Eingabe $[2, 8, 7, 3, 6, 5, 1, 4]$ an (mit Angabe der jeweiligen Zufallsentscheidung)!

b) Ist die Ausgabe immer eine korrekt sortierte Folge? Ist die Anzahl der möglichen Abläufe des Algorithmus linear, polynomiell oder exponentiell in Abhängigkeit einer festen Eingabe der Länge n? (mit Begründung)

c) Wieviele Vergleichsoperationen macht der Algorithmus minimal und maximal?[a]

d) Was berechnet der folgenden probabilistische Algorithmus **PSEL**(A,k) allgemein? Wenden Sie ihn auf $(A, 4)$ an, wobei A wie in a) und das erste zufällig gewählte Element 2 sei! Er benötigt $O(n)$ zu erwartende Schritte. Wieviel Schritte sind nötig, wenn das gleiche Problem mit einem deterministischen (d.h. nicht probabilistischen) Sortieralgorithmus gelöst wird?

von
5

[a] Das eingangs erwähnte Ergebnis beruht auf der Tatsache, dass mit der Wahrscheinlichkeit $0,75$ die Mengen $A_<$ und $A_>$ etwa gleich groß sind.

Algorithmus **PQS**(A)

Eingabe: Eine linear geordnete Menge von n verschiedenen Elementen.
Schritt 1: Falls $|A| = 1$ und $A = \{a\}$, dann **Ausgabe:** „a“. **Stopp**.
Falls $|A| > 1$, dann wähle zufällig ein Element $a \in A$.
Schritt 2: Setze
$A_< := \{b \in A | b < a\}$ und
$A_> := \{c \in A | c > a\}$
Schritt 3: Ausgabe: „$\mathbf{PQS}(A_<), a, \mathbf{PQS}(A_>)$“.

Algorithmus **PSEL**(A,k)

Eingabe: Eine linear geordnete Menge von n verschiedenen
Elementen und eine natürliche Zahl $1 \leq k \leq n$.
Schritt 1: Falls $|A| = 1$ und $A = \{a\}$, dann **Ausgabe:** „a“. **Stopp**.
Falls $|A| > 1$, dann wähle zufällig ein Element $a \in A$.
Schritt 2: Setze
$A_< := \{b \in A | b < a\}$ und
$A_> := \{c \in A | c > a\}$
Schritt 3: Falls $|A_<| > k$, dann $\mathbf{PSEL}(A_<, k)$,
sonst [falls $|A_<| = k - 1$, dann **Ausgabe:**„a“
sonst $\mathbf{PSEL}(A_>, k - |A_<| - 1)$];