Prof. J. Rothe Düsseldorf, 03.11.2009

Übung zur Vorlesung
Kryptologie 2

Blatt 4, Abgabe: 12.11.2009 bis 15:00 Uhr

Aufgabe 1 (5 Punkte): Bestimmen Sie mit dem Chinesischen Restesatz jeweils ein x, das die folgenden Kongruenzensysteme löst:

(a)

$$\begin{aligned} x &\equiv 2 \mod 4 \\ x &\equiv 1 \mod 3 \end{aligned}$$

(b)

$$\begin{aligned} x &\equiv 7 \mod 16 \\ x &\equiv 24 \mod 25 \end{aligned}$$

Aufgabe 2 (10 Punkte): Berechnen Sie

- $a = \log_2 10 \mod 13$ und
- $b = \log_3 267 \mod 401$

mit dem Pohlig-Hellman-Algorithmus.

Aufgabe 3 (15 Punkte):

(a) Führen Sie das ElGamal-Protokoll konkret mit den Werten $p = 401$, $\gamma = 3$, $a = 10$, $b = 15$ und $m = 323$ durch. Entscheiden Sie bei allen Zahlen, ob diese öffentlich oder geheim sind.

(b) Seien im ElGamal-Protokoll die Werte $p = 53$, $\gamma = 2$, $\beta = 14$ und $(\alpha_1, \alpha_2) = (17, 33)$ bekannt. Bestimmen Sie den Klartext m.

(c) Betrachten Sie nun das ElGamal-Protokoll für digitale Signaturen. Führen Sie dieses Protokoll konkret mit den Werten $p = 401$, $\gamma = 3$, $b = 11$, $s = 63$ und $m = 315$ durch. Geben Sie auch an, wie Alice kontrollieren kann, dass die Nachricht tatsächlich von Bob signiert wurde.

Aufgabe 4 (freiwillig): Schreiben Sie ein Java-Programm, das bei Eingabe von n die primitven Elemente von $\mathbb{Z}_n^*$ berechnet. Benutzen Sie die Klasse `BigInteger`, um große Zahlen darstellen zu können.
Für diese Aufgabe können Bonuspunkte vergeben werden – beachten Sie dazu Folgendes:

- Ihr Programm wird automatisch anhand einiger Beispielzahlen getestet. Das Programm soll n als Kommandozeilenparameter einlesen. Geben Sie die primitven Elemente von $\mathbb{Z}_n^*$ *zeilenweise* und *der Größe nach geordnet* (beginnend mit dem kleinsten) auf dem Bildschirm aus. Sollte es keine primitiven Elemente in $\mathbb{Z}_n^*$ geben, so ist eine entsprechende Meldung auf dem Bildschirm auszugeben.

- Wird kein Parameter übergeben, so soll das Programm automatisch Ihre Matrikelnummer für n wählen.

- Da es sich um eine Master-Vorlesung und eine Bonusaufgabe handelt, sind Sie beim Programmieren auf sich allein gestellt und bekommen keine Hilfestellung aus unserer Arbeitsgruppe.

- Es gibt keine Teilpunkte für "fast fertige" oder "teilweise lauffähige" Programme. Ein Programm kann nur dann mit Bonuspunkten prämiert werden, wenn es für *alle* Testzahlen die korrekte Ausgabe (in korrekter Formatierung) liefert.

- Die Programme werden mit natürlichen Zahlen ≥ 3 als Parameter getestet. Abfragen, ob dieser Parameter eine natürliche Zahl ist, sind unnötig (dürfen aber gerne implementiert werden).

- Es gilt das gleiche Prinzip, wie für die restlichen Übungsaufgaben: Abschreiben wird mit 0 Punkten bestraft.