Programmierpraktikum
Thema: Kryptographie
Sommersemester 2004

4. Übungsblatt

1. (*) Es soll ein mit RSA verschlüsselter Klartext m per Low-Exponent-Attacke dekodiert werden. Der Verschlüsselungsexponent sei $e = 3$. Die drei abgefangenen Chiffretexte c'_1, c'_2, c'_3 und ihre entsprechenden RSA-Moduln n_1, n_2, n_3 kann man auf der Webseite einsehen (`chiffretext4.txt`). Wie gehabt wurden nur die Buchstaben a-z verschlüsselt, diese werden durch die Werte 0-25 dargestellt.

   Beim Entschlüsseln sollte man folgendermaßen vorgehen:

   - Leer- und Satzzeichen im Chiffretext entfernen, ihre Positionen für das spätere Wiedereinfügen merken
   - Chiffretexte in BigInteger Zahlen c_1, c_2, c_3 übersetzen. Die Formel für einen Text $y_0 y_1 \dots y_k$ der Länge $k + 1$:

     $$y = \sum_{i=0}^{k} y_i N^{k-i}$$

     N ist die Anzahl der Zeichen des benutzten Alphabets, bei RSA-Modul n ist $k = \lfloor log_N n \rfloor$
     Hinweis 1: Es wurde nur ein Block verschlüsselt.
     Hinweis 2: Eine Methode in der Klasse `RSA` berechnet k.
   - finde die eindeutige Lösung $\mod n$ der drei simultanen Kongruenzen

     $$x \equiv c_1 \mod n_1, \quad x \equiv c_2 \mod n_2, \quad x \equiv c_3 \mod n_3$$

     mit dem chinesischen Restsatz, dabei ist $n = n_1 n_2 n_3$
   - es gilt $x = m^3$, ziehe die dritte Wurzel aus x mit einer Methode in der Klasse `CrpytMath` im package `crypto.protocols`
   - die Leer- und Satzzeichen in m wieder einfügen, überflüssige Symbole (68) am Ende entfernen

2. (Alexander/Oleg/Yves) Erweitert die Methode `friedman`, so dass nach der Berechnung der Schlüssellänge auch ein möglicher Schlüssel geraten wird. Dieser soll in der Variablen `String key` gespeichert werden. Das Verfahren zur Berechnung des kompletten Vigenère-Schlüssels ist im vierten Teil des Skripts beschrieben.

3. (David/Pavel) Integriert die Kasiski- und Friedmanmethoden der Klasse `Analyze` in das Vigenère-Dechiffrierfenster. Für das Substitutionsverfahren gibt es nun eine Möglichkeit, einen partiellen Schlüssel auf den Chiffretext anzuwenden. Dabei werden alle Buchstaben, die noch nicht zugeordnet sind, als "?" kodiert. Beispiel:

bekannte Zuordnungen	`E - q, N - b, S - w, T - e`
Darstellung Schlüssel	`????q????????b????we??????`

   Wenn man diesen String per Methode `stringToKey` in einem Objekt vom Typ Substitution speichert, wird das Flag `partialKey` gesetzt. Damit werden beim Entschlüsseln nur die bekannten Zuordnungen aufgelöst. Die entschlüsselten Zeichen werden beim Rückgabestring durch Großbuchstaben dargestellt. Verschlüsseln ist mit einem partiellen Schlüssel nicht möglich.

Abgabe: bis Dienstag, 29.6.2004