Vorlesung am 23.04.2015

4 Asymmetrische Verschlüsselung

- Öffentlicher Schlüssel (Public Key) zum Verschlüsseln
  (Jeder kann verschlüsseln)
- Geheimer Schlüssel (Secret Key) zum Entschlüsseln
  (Nur Inhaber des geheimen Schlüssels kann entschlüsseln))
- Aus öff. Schlüssel darf geh. Schlüssel nicht berechnet werden können

Beruhen auf der vermuteten Schwierigkeit mathematischer Probleme
(z.B. Faktoriserung großer Zahlen, Diskretes Logarithmusproblem)

Sehr ineffizient, daher Nutzung nur für kleine Datenmengen

Beispiel. Hybride Verschlüsselung

- Asymmetrisches Verfahren: Verschlüsselung symmetrischer Schlüssel
- Symmetrische Verfahren: Verschlüsselung der Daten

RSA-Verfahren: Entwicklet von Rivest, Shamir, Adleman am MIT 1977.

Sicherheit: Vermutete Schwierigkeit des Faktorisierens ganzer Zahlen.

Faktorisierungsproblem:
Gegeben: Zusammengesetzte natürliche Zahl $n = p \cdot q \in \mathbb{N}$, p, q prim.
Lösung: Finde die beiden Primfaktoren p und q.

Einschub Restklassenringe $(\mathbb{Z}_n, +, \cdot)$

- $(\mathbb{Z}_n, +)$ ist abelsche Gruppe, $(\mathbb{Z}_n, \cdot)$ abelsche Halbgruppe
- Für n Primzahl ist $(\mathbb{Z}_n, \cdot)$ eine Gruppe (und $(\mathbb{Z}_n, +, \cdot)$ ein Körper)
- $\mathbb{Z}_n^*$: Die bzgl. $\cdot$ invertierbaren Elemente (alle zu n teilerfremden Zahlen)

- $(\mathbb{Z}_n^*, \cdot)$ ist eine abelsche Gruppe (Übungsaufgabe)
- $\phi(n) := |\mathbb{Z}_n^*|$ heißt Eulerzahl
- n prim: $\phi(n) = n - 1$; $n = pq$ und p, q prim: $\phi(n) = (p-1)(q-1)$

Satz 4.1 (Satz von Euler)**.** *Für alle $a \in \mathbb{Z}_n^*$ gilt $a^{\phi(n)} \bmod n = 1$*

Proof. Sei $a \in \mathbb{Z}_n^*$. Die Abb. $f : \mathbb{Z}_n^* \longrightarrow \mathbb{Z}_n^*; x \mapsto ax \bmod n$ ist eine Bijektion. Also gilt

$$(\prod_{y \in \mathbb{Z}_n^*} y) \bmod n = (\prod_{y \in \mathbb{Z}_n^*} f(y)) \bmod n = (\prod_{y \in \mathbb{Z}_n^*} ay) \bmod n = (a^{\phi(n)} \prod_{y \in \mathbb{Z}_n^*} y) \bmod n$$

Kürzen ergibt die Behauptung. □

Zum RSA-Verfahren:

Schlüsselgenerierung

- Wähle Primzahlen p und q zufällig und unabhängig voneinander.
- Wähle öffentl. Exponenten $e \in \mathbb{N}$ mit $\mathrm{ggT}(e, \phi(n)) = 1$.
- Berechne geheimen Exponenten $d \in \mathbb{N}$ mit $e \cdot d = 1 \bmod \phi(n)$.
  - Ex., da e ein mult. Inverses in $\mathbb{Z}_{\phi(n)}$ besitzt
  - Berechnung mit erweitertem Euklidischen Alg.:
    Liefert $e \cdot d + y \cdot \phi(n) = \mathrm{ggT}(e, \phi(n))\ (= 1)$
- $n = p \cdot q$ heißt RSA-Modul (in $\mathbb{Z}_n$ ver- und entschl. wir)
- (n, e): öffentliche Schlüssel, (n, d): geheime Schlüssel.
- p, q und $\phi(n)$ müssen geheim bleiben:
  Damit kann aus e der geheime Exponent d berechnet werden.

Verschlüsselung einer Nachricht $m < n$: berechne $c = m^e \bmod n$.

Entschlüsselung eines Geheimtextes $c < n$: berechne $m = c^d \bmod n$.

Beispiel.
- $p = 5, q = 11$, also $n = 55$ und $\phi(n) = 40$.

- Kandidaten für e : 3,7,9,... Wir wählen 7.
- $d = 23 : 7 \cdot 23 = 161 = 1 \bmod 40$.
- Verschl. von $m = 3$: $c = m^e = 3^7 = 2187 = 42 \bmod 55$.
- Entschl. von $c = 42$: $c^d = 42^{23}$.
- Trick: Binäre Exponentiation (square and multiply):
  - $23 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$.
  - $23 = ((2 \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 1$.
  - Also $42^{23} = (((42^2)^2 \cdot 42)^2 \cdot 42)^2 \cdot 42$.
  - Ausrechnen:
    - $42^2 = 1764 = 4 \bmod 55$, $4^4 = 16 \bmod 55$.
    - $16 \cdot 42 = 672 = 12 \bmod 55$, $12^2 = 144 = 34 \bmod 55$.
    - $34 \cdot 42 = 1428 = 53 \bmod 55$, $53^2 = 2809 = 4 \bmod 55$.
    - $4 \cdot 42 = 168 = 3 \bmod 55$.

Satz 4.2. *Sind n, e und d wie oben gewählt, dann gilt für alle $m \in \mathbb{Z}_n$*

$$(m^e)^d \bmod n = m.$$

Beweisskizze. Es gilt $e \cdot d = 1 \bmod \phi(n)$, also ex. $k \in \mathbb{N}$ mit $e \cdot d - k \cdot \phi(n) = 1$, d.h. $e \cdot d = 1 + k \cdot \phi(n)$, also $(m^e)^d = m^{ed} = m^{1+k\phi(n)} = m \cdot (m^{\phi(n)})^k$.

Fall 1. m, n teilerfr.: Nach Euler: $m^{\phi(n)} = 1 \bmod n$, also $(m^e)^d = m \bmod n$.

Fall 2. m, n nicht teilerfr.: Etwas komplizierter. □

Schlüssellänge Beurteilung der Sicherheit von RSA.
Alle kryptoanalytischen Methoden (nicht nur Schlüsselexhaustion)

Erinnerung: Wir fordern Sicherheitsniveau von 100 Bit,
Angreifer benötigt ca. 2^{100} Versuche, die Entschlüsselung zu berechnen.

Gesucht: Untere Schranke für die Schlüssellänge.

Dazu: Betrachte drei Probleme:

P1: Gegeben öff. Schl. (n, e) und Cipher $c = m^e \bmod n$. Berechne m.

P2: Gegeben öff. Schl. (n, e). Berechne d mit $ed = 1 \bmod \phi(n)$.

P3: Gegeben Modul $n = pq$. Berechne p und q. (Faktorisierungsproblem)

Offensichtlich gilt: P3 lösbar $\Longrightarrow$ P2 lösbar $\Longrightarrow$ P1 lösbar.

- P3 lösbar $\Longrightarrow$ P2 lösbar
  - Berechne p, q.
  - Berechne d mit $ed = 1 \mod \phi(n)$ ($\phi(n) = (p-1)(q-1)$.
- Problem 2 lösbar. Öff. Schlüssel (n, e) bekannt.
  - Berechne d.
  - Berechne $m = c^d \bmod n$.

Frage: Sind Probleme äquivalent, d.h. gilt auch
P1 lösbar $\Longrightarrow$ P2 lösbar $\Longrightarrow$ P3 lösbar?

- Bekannt: P2 und P3 sind äquivalent.
- Bisher nicht bekannt, ob P1 und P2 auch äquivalent sind.
  D.h. Entschlüsselung könnte leichter sein als d zu berechnen.

Derzeit schnellster Faktorisierungsalgorithmus: Laufzeit $\mathcal{O}(e^{\sqrt{\ln n \ln\ln n}})$.

Das bedeutet: $n \approx 2^{2048}$. Selbst nachrechnen!