RSA Key Generation

Luc Spachmann

Friedrich-Schiller-Universität Jena

11.01.2024

RSA Schlüssel

- Schlüssel (e, n) und (d, n) mit

$$ed = 1 \mod \varphi(n)$$

  und n Produkt zweier Primzahlen p und q
- φ ist Eulersche φ-Funktion
- Für Produkt zweier Primzahlen pq gilt

$$\varphi(pq) = (p - 1)(q - 1)$$

Schlüsselgenerierung

- Generiere 2 ausreichend große Primzahlen p, q
- Beide Primzahlen sollten nicht zu nahe aneinander liegen
- Erzeuge dazu zufälliges z in der gewünschten Größe
- Teste $30z + i$ für $i \in \{1, 7, 11, 13, 17, 19, 23, 29, 30 + 1, \ldots\}$
- Erzeuge zufälliges e Teilerfremd zu $\varphi(pq)$ (oft $2^{16} + 1$)
- Berechne d mit $de \equiv 1 \mod \varphi(pq)$ mit eEA
- Schlüssel sind dann $(e, pq), (d, pq)$

Primzahltest von Miller-Rabin

- Nichtdeterministischer Primzahltest
- Sehr schnell auch für große Zahlen
- W'Keit von 'False Negative' ist Null
- W'Keit von 'False Positive' $< \frac{1}{4}$
- Durch wiederholte Anwendung kann Fehlerwahrscheinlichkeit verringert werden

Primzahltest von Miller-Rabin

```
Require: n ∈ ℕ
Bestimme ungerades m mit n − 1 = 2^k · m
Wähle zufälliges 2 ≤ a < n
b = a^m mod n
if b ≡ 1 mod n then
    return Prim
end if
for i = 1 to k do
    if b ≡ −1 mod n then
        return Prim
    else
        b = b^2 mod n
    end if
end for
return Zusammengesetzt
```

Erweiterter euklidischer Algorithmus

- Berechnet $\mathrm{ggT}(a, b) = sa + tb = c$

Require: a, b

$k = 0, r_0 = a, r_1 = b, s_0 = 1, s_1 = 0, t_0 = 0, t_1 = 1$
repeat
    Erhöhe k um 1
    $q_k = \frac{r_{k-1}}{r_k}$
    $r_{k+1} = r_{k-1} - q_k \cdot r_k$
    $s_{k+1} = s_{k-1} - q_k \cdot s_k$
    $t_{k+1} = t_{k-1} - q_k \cdot t_k$
until $r_{k+1} = 0$
return r_k, s_k, t_k

Schlüsselgenerierung

- Generiere 2 ausreichend große Primzahlen p, q
- Beide Primzahlen sollten nicht zu nahe aneinander liegen
- Erzeuge dazu zufälliges z in der gewünschten Größe
- Teste $30z + i$ für $i \in \{1, 7, 11, 13, 17, 19, 23, 29, 30 + 1, ...\}$
- Miller-Rabin mehrmals ausfühen zum Verifizieren!
- Erzeuge zufälliges e Teilerfremd zu $\varphi(pq)$ (oft $2^{16} + 1$)
- Falls nicht Teilerfremd, mit anderem e Wiederholen
- Berechne d mit $de \equiv 1 \mod \varphi(pq)$ mit eEA
- Schlüssel sind dann $(e, pq), (d, pq)$

Aufgaben

- Implementiert die RSA Schlüsselgenerierung
- Programmname [Länge] [Output privat] [Output öffentlich] [benutzte Primzahlen]
- Länge: Gewünschte Bitlänge des Schlüssels (ca, als Zahl)
- Schlüssel: Zwei Zeilen
  - Erste Zeile: e bzw d
  - Zweite Zeile: n (beides Dezimal)
- Benutzte Primzahlen als Datei in zwei Zeilen