**Aufgabe:** Sei $n = p \cdot q$ und $p, q$ (Primzahlen) sind bekannt. Wie kann man feststellen, ob $x$ quadratischer Rest modulo $n$ ist, und wie kann man die Wurzeln von $x$ bestimmen?

**Antwort:** Eine Konsequenz aus dem Chin. Restsatz. $x$ muss quadratischer Rest modulo $p$ und modulo $q$ sein. Dies kann das Euler-Kriterium feststellen: $x^{\frac{p-1}{2}} \equiv 1 \pmod{p}$ und $x^{\frac{q-1}{2}} \equiv 1 \pmod{q}$.

Um die Quadratwurzeln von $x$ zu bestimmen, bestimme man die Q. Wurzeln von $x$ modulo $p$ (diese seien $u$ und $u' = p - u$), sowie die Quadratwurzeln modulo $q$ (diese seien $v$ und $q - v = v'$). Die 4 QWurzeln von $x$ modulo $n$ ergeben sich dann durch die C.R.-Transformation:

$w_1$ ○—● $(u, v)$

$w_2$ ○—● $(u', v)$

$w_3$ ○—● $(u, v')$

$w_4$ ○—● $(u', v')$

Hiebei ist das Wurzelziehen modulo $p$ (analog: modulo $q$) besonders einfach, wenn $p \equiv 3 \pmod 4$. Dann ist eine Q Wurzel von $x$ mod $p$ nämlich $x^{\frac{p+1}{4}} \bmod p$, also modexp$(x, (p+1)/4, p)$.

<u>Bem:</u> Eine Zahl $n = p \cdot q$ bei der die Primzahlen $p$ und $q$ die Eigenschaft haben: $p \equiv 3 \pmod 4$ und $q \equiv 3 \pmod 4$ heißt auch <u>Blum integer</u>.

Es gilt:

Faktorisieren (von $n$) $\equiv_{\text{prob-eff}}$ Quadratwurzel-Ziehen $\pmod n$

Beweis: Die Richtung ($\geq$) wurde gerade eben begründet: Wurzelziehen mod $p$ und mod $q$, dann Chin. Restsatz anwenden.

Die andere Richtung ($\leq$): Gegeben $n$, man möchte die Faktorisierung von $n$ finden. Hierzu wählt man (ggf. mehrfach) ein $b<n$ per Zufall, berechnet dann $a = b^2 \bmod n$ und lässt sich vom Orakel eine Quadratwurzel $w$ von $a$ bestimmen. Mit Wahrscheinlichkeit $\frac{1}{2}$ ist $w \not\equiv \pm b \pmod n$.

In diesem Fall gilt dann

$$b^2 - w^2 \equiv a - a \equiv 0 \pmod n$$

also

$$\underbrace{(b-w)}_{\not\equiv 0} \cdot \underbrace{(b+w)}_{\not\equiv 0} \equiv 0 \pmod n$$

, da $w \not\equiv \pm b$

Dann liefert $ggT(b-w, n)$ bzw $ggT(b+w, n)$ einen nicht-trivialen Faktor von $n$. $\square$

## Kleines Zahlenbeispiel:

Sei $n = 437 = \underbrace{19}_{p} \cdot \underbrace{23}_{q}$

Es gilt $p \equiv 3 \pmod 4$ und $q \equiv 3 \pmod 4$

Bestimme die Quadratwurzeln von $x = 100 \bmod n$.

Es gilt: $100 \bmod 19 = 5$

$100 \bmod 23 = 8$

Also: $100 \multimap (5,8)$

Nun Quadratwurzeln von 5 mod 19:

$$5^{\frac{p+1}{4}} \bmod 19 = 5^5 \bmod 19 = \underline{\underline{9}}$$

Die 2. QWurzel mod 19 ist dann $19 - 9 = \underline{\underline{10}}$.

QWurzel von 8 mod 23 bestimmen:

$$8^{\frac{q+1}{4}} \bmod 23 = 8^6 \bmod 23 = \underline{\underline{13}}$$

Die 2. QWurzel mod 23 ist $23 - 13 = \underline{\underline{10}}$

Die 4 Quadratwurzeln von $x = 100$ sind somit (mittels CR):

$(9, 13) \multimapinv 427$

$(9, 10) \multimapinv 332$

$(10, 13) \multimapinv 105$

$(10, 10) \multimapinv 10$

Fortsetzung des Zahlenbeispiels, um

Faktorisieren $\leq_{prob\cdot eff}$ Quadratwurzel ziehen

zu illustrieren.

Wir wählen $b = 10$ „zufällig" und berechnen $a = b^2 = 100$. Das Orakel soll eine Quadratwurzel $w$ von 100 liefern mit $w \not\equiv \pm b$. Wir haben „Glück" und erhalten $w = 105$ als Antwort. Dann ergibt sich

$$ggT(b-w, n) = ggT(10-105, 437) = 19$$

$$ggT(b+w, n) = ggT(10+105, 437) = 23$$

# Public Key-Verfahren von Rabin

(eher von akademischen Interesse ?)

Jeder Teilnehmer wählt in der Initialisierungsphase 2 große, geheime Primzahlen $p, q$.
(Diese sind der geheime Schlüssel.)

Der Teilnehmer berechnet $n = p \cdot q$ und dies ist der öffentliche Schlüssel

Protokoll (A sendet an B Nachricht m)

| A | | B |
|---|---|---|
| besorgt sich B's öffentlichen Schlüssel $n$ | | |
| $c := m^2 \bmod n$ | | |
| | $\xrightarrow{c}$ | |
| | | Berechnet die 4 Quadratwurzeln von $c$ modulo $n$ unter Zuhilfenahme von $p$ und $q$. Entscheidet, welche der 4 Quadratwurzeln die richtige Nachricht $m$ ist. |

(Akademischer?) Vorteil von Rabin gegenüber RSA: „Rabin knacken" ($\hat{=}$ Wurzelziehen mod $n$) ist genauso schwierig wie $n$ faktorisieren. Bei RSA weiß man nur die eine Richtung:

$$\text{„RSA knacken"} \leq_{\text{eff}} n \text{ faktorisieren}$$

---

Eine kleine Abschweifung:

Sei $n$ $(= p \cdot q)$ öffentlich, aber $p, q$ geheim.

Der <u>Blum-Blum-Shub Pseudozufalls-generator</u> ist kryptographisch sicher (sofern man $n$ nicht faktorisieren kann):

$x_0$ beliebig (seed)

$x_{i+1} = (x_i)^2 \mod n$

**Beispiel:** Sei $n = 23 * 29$ und $x_0 = 2$.
$= 667$

Dann erhält man folgende Folge von Quadratzahlen mod n (also BBS-Pseudozufallszahlen):

2, 4, <u>16</u>, 256, 170, 219, 604, 634, 422, 662,
25, 625, 430, 141, 538, 633, 489, 335, 169,
547, 393, 372, 315, 509, 285, 518, 190, 82,
54, 248, 140, 257, <u>16</u>, ... (Periodenlänge: 30)

**Bem:** Insgesamt könnte man maximal

$$\frac{p-1}{2} \cdot \frac{q-1}{2} = 11 \cdot 14 = 154$$ Quadratzahlen

durchlaufen, bis sich die Folge wiederholt.

RSA und Rabin verwenden Schwierigkeit, eine Zahl $n\ (=p\cdot q)$ zu faktorisieren.

Ein Public Key-Verfahren, das auf dem Problem, den Diskreten Logarithmus zu bestimmen, beruht stammt von <u>El Gamal</u>.

Herleitung über das Diffie-Hellman-Schlüsselvereinbarungsprotokoll: In der Initialisierungsphase wird eine große Primzahl $n$ und eine Primitivwurzel $a$ modulo $n$ festgelegt. Alle Teilnehmer kennen $n$ und $a$. Alle Rechnungen sind fortan modulo $n$.

<u>Diffie-Hellman-Protokoll:</u>

| A | | B |
|---|---|---|
| Wählt $x$ zufällig | | Wählt $y$ zufällig |
| $\tilde{x} := a^x$ | | $\tilde{y} = a^y$ |
| | $\xleftarrow{\tilde{y}}$ $\quad$ $\xrightarrow{\tilde{x}}$ | |
| $s := (\tilde{y})^x$ | | $s := (\tilde{x})^y$ |

Bem: DH-Problem: berechne $\tilde{x}, \tilde{y} \mapsto s$

Dieses Protokoll <u>asymmetrisch</u> machen:

A will Nachricht $m$ an Empfänger B senden.

B erzeugt in der (individuellen) Initialisierungsphase einen geheimen + einen öffentlichen Schlüssel:

B wählt Zufallszahl $y$

$\tilde{y} := a^y$

($y \hat{=}$ geheimer Schlüssel,

$\tilde{y} \hat{=}$ öffentl. Schlüssel)

| A | B |
|---|---|
| Besorgt sich B's öffentlichen Schlüssel $\tilde{y}$ | |
| Wählt $x$ zufällig | |
| $\tilde{x} := a^x$ | |
| $s := (\tilde{y})^x$ | |
| $c := m \cdot s$ (↑ Nachricht) $\xrightarrow{(\tilde{x}, c)}$ | |
| | $s := (\tilde{x})^y$ |
| | $m := c \cdot s^{-1}$ |

Nicht überraschend ist die Tatsache:

$$\underbrace{\text{Diffie-Hellman „knacken"}}_{\text{das sog. „Diffie-Hellman-Problem"}} \equiv_{\text{eff}} \text{ElGamal „knacken"}$$

Ähnlich wie bei RSA besteht nur eine Reduktion in eine Richtung, wenn man mit dem Diskreten Log-Problem vergleicht:

$$\text{El Gamal „knacken"} \leq_{\text{eff}} \text{Diskreten Log. berechnen.}$$

Vergleich von RSA mit ElGamal:

| | RSA | ElGamal |
|---|---|---|
| Länge der gesendeten Chiffre | etwa so lang wie m | etwa doppelt so lang wie m |
| Sicherheit | Faktorisieren | Diskr. Log. |
| Art | deterministisch (bei derselben Nachricht m immer dieselbe Chiffre) | probabilistisch (immer andere Chiffren $(\tilde{x}, c)$, Vorteil: Randomisierung) |
| Laufzeit für Ver/Entschlüsselung | modulare Exponentiation $O(m^3)$ evtl. bei speziell gewähltem Expon: $O(m^2)$ | modulare Expon. (+ Inverse bestimmen) $O(m^3)$ |
| Verschlüsselung und Entschlüsselung kommutieren? (wichtig für Signatur) | ja | nein, es gibt aber spezielles ElGamal-Signaturverfahren, das auf ElGamal-Infrastruktur aufsetzt. |

# Elektronische Unterschrift

Dokument m wird von A unterschrieben:

A erzeugt: (m, A's Unterschrift an m)

Ziele/Eigenschaften:

- nur A kann diese Unterschrift erzeugen (→ fälschungssicher).

- jeder Teilnehmer kann überprüfen, dass A das Dokument m unterschrieben hat.

- von A nicht abstreitbar, dass er das Dokument unterschrieben hat.

- Die „Unterschrift" hängt von m ab. Man kann sie nicht an ein anderes Dokument m' anfügen:

  (m', A's Unterschrift an m)

  Unzulässigkeit ist überprüfbar.

Gegeben ein Public Key-System $E(\,)$, $D(\,)$ mit öffentlichen Schlüsseln $k_X$ und geheimen Schlüsseln $k'_X$ für jeden Teilnehmer $X$.

A unterschreibt Dokument $m$:

$$(m, \tilde{m}) \qquad \text{wobei } \tilde{m} = D(k'_A, m)$$

Überprüfen der Unterschrift:

$$E(k_A, \tilde{m}) \stackrel{?}{=} m$$

Dies setzt voraus, dass

$$E(k_A, D(k'_A, m)) = m$$

umgekehrte Reihenfolge: normalerweise verwendet man:

$$D(k'_A, E(k_A, m)) = m$$

Voraussetzung: $E(\,)$ und $D(\,)$ müssen <u>kommutieren</u>

Bei RSA ist das der Fall:

$$(m^e \bmod n)^d \bmod n = m^{de} \bmod n = (m^d \bmod n)^e \bmod n$$