Das Quadratische Reste Problem

Definition Pseudoquadrate

Sei $N = pq$ mit p, q prim. Eine Zahl a heißt *Pseudoquadrat* bezüglich N, falls

$$\left(\frac{a}{N}\right) = 1 \text{ und } a \notin QR_N.$$

Wir definieren die Sprache

$$\text{QUADRAT} := \{a \in \mathbb{Z}_N^* \mid \left(\tfrac{a}{N}\right) = 1 \text{ und } a \in QR_N\}.$$

- Für alle Pseudoquadrate a gilt: $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = (-1)$.
- D.h. die Sprache QUADRAT kann effizient entschieden werden, falls p, q bekannt sind. Im Allgemeinen ist nur N bekannt.

Quadratische Reduositätsannahme (QR-Annahme)

Es gibt keinen polynomiellen Algorithmus, der QUADRAT entscheidet.

Quadratwurzeln in $\mathbb{Z}_N^*$

Lemma

Sei $N = pq$ mit p, q prim und $p = q = 3 \bmod 4$ (sogenannte Blum-Zahl). Dann besitzt jedes $a = x^2 \in QR_N$ genau eine Quadratwurzel in QR_N, die sogenannte Hauptwurzel.

Beweis:

- Die Lösungen des Gleichungssystems $\left| \begin{array}{l} y = \pm x \bmod p \\ y = \pm x \bmod q \end{array} \right|$ liefern mittels Chinesischem Restsatz 4 Lösungen in $\mathbb{Z}_N^*$.
- Eine Lösung ist in QR_N gdw sie in $QR_p \times QR_q$ ist.
- Betrachten Lösung modulo p (analog mod q):

$$\left(\frac{x}{p}\right) = \left(\frac{(-1)(-x)}{p}\right) = \left(\frac{-1}{p}\right) \cdot \left(\frac{-x}{p}\right).$$

- Für $p = 3 \bmod 4$ gilt $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = (-1)$.
- D.h. $\left(\frac{x}{p}\right) = -\left(\frac{-x}{p}\right)$ und entweder x oder $-x$ ist in QR_p.
- Damit ist genau eine der 4 Lösungen in QR_N.

Der Blum-Blum-Shub (BBS) Pseudozufallsgenerator

Korollar

Die Abb. $f : QR_N \to QR_N, x \mapsto x^2 \bmod N$ ist eine Bijektion auf QR_N.

- (k, ℓ)-Pseudozufallsgeneratoren generieren aus k Zufallsbits eine Sequenz von $\ell > k$ Zufallsbits.
- Der (k, ℓ)-BBS Generator verwendet obige Bijektion.

Algorithmus BBS Pseudozufallsgenerator (1986)

EINGABE: $N = pq$ Blumzahl der Bitlänge $|N| = k$,
1^ℓ mit $\ell \in \mathbb{N}$ und $\ell > k$, $r \in \mathbb{Z}_N^*$

1. Setze $s_0 = r^2 \bmod N$.
2. For $i = 1$ to ℓ
   1. Setze $s_i \leftarrow s_{i-1}^2 \bmod N$. Gib $z_i = s_i \bmod 2$ aus.

AUSGABE: $(z_1, \ldots, z_\ell) \in \{0,1\}^\ell$.

Laufzeit: $\mathcal{O}(\ell \log^2 N)$, d.h. polynomiell in der Eingabelänge.

Die Sicherheit des BBS Generators

Sicherheit: Man kann die Verteilung der $(z_1, \ldots, z_\ell)$ nicht von der Gleichverteilung auf $\{0,1\}^\ell$ unterscheiden.

Man kann folgendes zeigen:

- Sei A ein polynomieller Unterscheider für $(z_1, \ldots, z_\ell)$.
- Dann gibt es einen polyn. Algorithmus B, der $s_0 \bmod 2$ berechnet.

Satz Sicherheit des BBS Generators

Die Ausgabe des BBS Generators ist von der Gleichverteilung in polynomieller Zeit ununterscheidbar unter der QR-Annahme.

- Annahme: $\exists$ polyn. Unterscheider A für den BBS Generator.
- Sei B ein Algorithmus, der $s_0 \bmod 2$ berechnet.
- Zeigen, dass dann ein polyn. Algorithmus für QUADRAT existiert. (Widerspruch zur Quadratischen Residuositätsannahme)

Entscheiden der Sprache QUADRAT

Algorithmus für QUADRAT

EINGABE: $N, a \in \mathbb{Z}_N^*$ mit $\left(\frac{a}{N}\right) = 1, 1^\ell$

1. Setze $s_0 \leftarrow a \bmod N$.
2. Berechne $(z_1, \ldots, z_\ell)$ mittels BBS Generator.
3. Berechne $z_0 \leftarrow B(z_1, \ldots, z_\ell)$.
4. Falls $z_0 = (a \bmod 2)$, Ausgabe "$x \in QR_N$".
   Sonst Ausgabe "$x \notin QR_N$".

Laufzeit: $\mathcal{O}(\ell \cdot \log^2 N + T(B))$

Korrektheit:

- Wegen $\left(\frac{a}{N}\right) = 1$ ist entweder a oder $(-a) = N - a$ in QR_N.
- D.h. a oder $(-a)$ ist eine Hauptwurzel von $s_1 = a^2 \bmod N$.
- Genau eine der beiden Zahlen $a, (-a)$ ist gerade.
- z_0 ist das unterste Bit der Hauptwurzel von $s_1 = a^2 \bmod N$.
- D.h. a ist eine Hauptwurzel gdw z_0 und $a \bmod 2$ übereinstimmen.

Probabilistische Verschlüsselung

Algorithmus Goldwasser-Micali Kryptosystem (1984)

1. Schlüsselgenerierung: Wähle Blumzahl $N = pq$. Wähle $z \in_R \mathbb{Z}_N^*$, so dass z ein Pseudoquadrat ist. Setze den öffentlichen Schlüssel $pk = (N, z)$ und den privaten Schlüssel $sk = (p, q)$.
2. Verschlüsselung: Für $m \in \{0, 1\}$ wähle $x \in_R \mathbb{Z}_N^*$ und berechne
$$c \leftarrow z^m x^2 \bmod N.$$
3. Entschlüsselung: Für einen Chiffretext c berechne
$$m = \begin{cases} 0 & \text{falls } c \in QR_N, \text{ d.h. falls } \left(\frac{c}{p}\right) = 1. \\ 1 & \text{falls } c \notin QR_N, \text{ d.h. falls } \left(\frac{c}{p}\right) = (-1). \end{cases}$$

Sicherheit des Goldwasser-Micali Kryptosystems

Korrektheit

- Falls $m = 0$ ist $c = x^2$ ein zufälliger quadratischer Rest in $\mathbb{Z}_N^*$.
- Falls $m = 1$ ist $c = z \cdot x^2$ ein zufälliges Pseudoquadrat.
- Es gilt $\left(\frac{c}{N}\right) = \left(\frac{z^m x^2}{N}\right) = \left(\frac{z}{N}\right)^m \cdot \left(\frac{x^2}{N}\right) = 1$.
- D.h. entweder $\left(\frac{c}{p}\right) = \left(\frac{c}{q}\right) = 1$ oder $\left(\frac{c}{p}\right) = \left(\frac{c}{q}\right) = (-1)$.
- Im ersten Fall ist $c \in QR_N$, im zweiten Fall gilt $c \notin QR_N$.

Laufzeit:

- Verschlüsselung: $\mathcal{O}(\log^2 N)$
- Entschlüsselung: $\mathcal{O}(\log^2 N)$

Satz Sicherheit des Goldwasser-Micali Kryptosystems

Das GM Kryptosystem ist sicher unter der QR-Annahme.

Beweisidee:

- Unterscheiden von Verschlüsselungen von 0 und 1 ist äquivalent zum Entscheiden der Sprache QUADRAT.

Bit Commitments

Szenario informal:

1. **Commitment-Phase:**
   - Alice platziert ein Bit $b \in \{0,1\}$ in einem Safe, der in Bob's Zimmer steht. Bob besitzt keinen Safeschlüssel.
   - Bob kann den Safe nicht einsehen, lernt also nichts über b. **(Conceiling Eigenschaft)**
2. **Revealing-Phase:**
   - Alice öffnet den Safe und zeigt Bob das Bit b.
   - Alice kann ihr Bit dabei nicht ändern. **(Binding Eigenschaft)**

Mathematische Modellierung

- Commitment mittels $f : \{0,1\} \times X \to Y$ für endliche Mengen X, Y.
- Commitment (sog. Blob): Wähle $x \in X$ und sende $f(b,x)$ an Bob.
- Öffnen des Commitments: Sende b und x an Bob.

Bit Commitment via Goldwasser-Micali Kryptosystem

Öffentliche Parameter:

- Blumzahl N, Pseudoquadrat $z \in \mathbb{Z}_N^*$
- $X = Y = \mathbb{Z}_N^*$

Algorithmus Goldwasser-Micali Bit Commitment

1. Commitment-Phase
   - Wähle $x \in_R \mathbb{Z}_N^*$.
   - Sende Blob $f(b,x) = z^b x^2 \bmod N$ an Bob.
2. Revealing-Phase
   - Sende b, x an Bob.
   - Bob überprüft die Korrektheit von $f(b,x) = z^b x^2 \bmod N$.

Conceiling Eigenschaft:

- Unter der QR-Annahme lernt Bob nichts über das Bit $b \in \{0,1\}$.

Binding Eigenschaft

Satz

Goldwasser-Micali Commitments besitzen die Binding Eigenschaft.

Beweis:

- **Annahme:** Alice kann Blob $f(b,x)$ für $b = 0$ und $b = 1$ öffnen.
- D.h. Alice kann $x_1, x_2 \in \mathbb{Z}_N^*$ berechnen mit

$$f(b,x) = z^0 x_1^2 = z^1 x_2^2 \bmod N.$$

- Daraus folgt $z = \left(\frac{x_1}{x_2}\right)^2 \bmod N$, d.h. $\frac{x_1}{x_2}$ ist Quadratwurzel von z. (Widerspruch: z ist ein Pseudoquadrat in $\mathbb{Z}_N^*$.)

Münzwurf über das Telefon

- Bit Commitments haben zahlreiche Anwendungen in kryptographischen Protokollen.
- Exemplarisch hier ein Protokoll für einen fairen Münzwurf.

Algorithmus Münzwurf via Internet

1. Alice sendet Bob Commitment für Bit $b \in \{0, 1\}$.
2. Bob rät ein Bit $b' \in \{0, 1\}$.
3. Alice öffnet ihr Bit. Bob gewinnt gdw $b' = b$.

- Conceiling-Eigenschaft verhindert, dass Bob etwas über b lernt.
- Binding-Eigenschaft verhindert, dass Alice b in $1 - b'$ ändert.

Berechnen von Quadratwurzeln modulo p

Satz Quadratwurzeln mod p

Sei p prim, $p = 3 \bmod 4$ und $a \in QR_p$. Dann sind die beiden Quadratwurzeln von a von der Form

$$x = \pm a^{\frac{p+1}{4}} \bmod p, \quad \text{wobei } a^{\frac{p+1}{4}} \in QR_p.$$

- Es gilt

$$x^2 = a^{\frac{p+1}{2}} = a^{\frac{p-1}{2}} \cdot a = \left(\frac{a}{p}\right) \cdot a = a \bmod p.$$

- Ferner gilt $a^{\frac{p+1}{4}} \bmod p \in QR_p$ wegen

$$\left(\frac{a^{\frac{p+1}{4}}}{p}\right) = \left(\frac{a}{p}\right)^{\frac{p+1}{4}} = 1.$$

D.h. Quadratwurzeln können in Zeit $\mathcal{O}(\log^3 p)$ berechnet werden.

Das Blum-Goldwasser Kryptosystem

Algorithmus Blum-Goldwasser Kryptosystem (1985)

1. Schlüsselgenerierung: Wähle Blumzahl $N = pq$.
   Setze $pk = N$ und $sk = (p, q)$.
2. Verschlüsselung: Für $m = (m_1, \ldots, m_\ell) \in \{0,1\}^\ell$:
   - Wähle $r \in_R \mathbb{Z}_N^*$.
   - $(z_1, \ldots, z_\ell) \leftarrow$ BBS Generator auf $s_0 = r^2 \bmod N$.
   - For $i = 1$ to ℓ: Berechne $c_i = m_i + z_i \bmod 2$.
   - Berechne $s_{\ell+1} = s_0^{2^{\ell+1}} \bmod N$.
   - AUSGABE: Chiffretext $c = (c_1, \ldots, c_\ell, s_{\ell+1}) \in \{0,1\}^\ell \times \mathbb{Z}_N^*$.
3. Entschlüsselung von c mittels $sk = (p, q)$:
   - Berechne $s_0 \in \mathbb{Z}_N^*$ als Lösung von $\left| \begin{array}{rcll} s_0 & = & s_{\ell+1}^{\left(\frac{p+1}{4}\right)^{\ell+1}} & \bmod p \\ s_0 & = & s_{\ell+1}^{\left(\frac{p+1}{4}\right)^{\ell+1}} & \bmod q \end{array} \right|$.
   - $(z_1, \ldots, z_\ell) \leftarrow$ BBS Generator auf $s_0 = r^2 \bmod N$.
   - For $i = 1$ to ℓ: Berechne $m_i = c_i + z_i \bmod 2$.

Laufzeit und Korrektheit

Korrektheit:

- $(z_1, \ldots, z_\ell)$ wird als One-Time Pad für m verwendet.
- Entschlüsselung berechnet $\ell + 1$-malig die Hauptwurzel von $s_{\ell+1}$.
- Dies rekonstruiert die Saat s_0 des BBS Generators.

Laufzeit:

- Verschlüsselung: $\mathcal{O}(\ell \cdot \log^2 N)$
- Entschlüsselung: $\mathcal{O}(\log^3 N + \ell \cdot \log^2 N)$.

Satz Sicherheit des BG-Kryptosystems

Das Blum Goldwasser Kryptosystem ist sicher unter der Annahme, dass Blumzahlen $N = pq$ schwer zu faktorisieren sind.

(ohne Beweis)

Elliptische Kurven

Definition Elliptische Kurve

Sei $p \neq 2, 3$ prim, $f(x) = x^3 + ax + b \in \mathbb{Z}_p[x]$, $4a^3 + 27b^2 \neq 0 \bmod p$.
Wir definieren für $f(x)$ eine *elliptische Kurve* E als

$$\{(x, y) \in \mathbb{Z}_p \mid y^2 = x^3 + ax + b\} \cup \{\mathbf{O}\},$$

wobei **O** der Punkt im Unendlichen heißt.

Anmerkungen:

- Die Bedingung $4a^3 + 27b^2$ ist äquivalent zu der Forderung, dass $f(x)$ in $\mathbb{Z}_p^*$ keine mehrfachen Nullstellen besitzt. (Übung)
- Für jeden Punkt $P = (x, y)$ auf E liegt auch $(x, -y)$ auf E.
- Wir definieren $-P = (x, -y)$.
- Für $P = \mathbf{O}$ definieren wir $-P = \mathbf{O}$ und $P + Q = Q$ für alle Q auf E.

Addition von Punkten

Algorithmus Addition von Punkten auf E

EINGABE: $P = (x_1, y_1)$, $Q = (x_2, y_2)$ auf E mit $P, Q \neq \mathbf{O}$

1. Falls $x_1 = x_2$ und $y_1 = -y_2$, Ausgabe $\mathbf{O}$.
2. Setze $\alpha := \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{für } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} & \text{für } x_1 = x_2 \end{cases}$. Setze $\beta = y_1 - \alpha x_1$.
3. Berechne $x_3 = \alpha^2 - x_1 - x_2$ und $y_3 = -(\alpha x_3 + \beta)$.

AUSGABE: $P + Q = (x_3, y_3)$

Anmerkungen:

- Sei $P \neq Q$. Wir betrachten die Gerade G durch P, Q.
- Falls $Q = -P$, so liegt G parallel zur y-Achse. Wir definieren
$$P + (-P) = \mathbf{O}.$$
- Sonst ist G definiert durch $y = \alpha x + \beta$ mit Steigung $\alpha = \frac{y_2 - y_1}{x_2 - x_1}$.
- Für $P = Q$ besitzt die Tangente im Punkt P Steigung $\alpha = \frac{3x_1^2 + a}{2y_1}$.