Diffie-Hellman Schlüsselaustausch (1976)

Öffentliche Parameter: Primzahl p, Generator g von $\mathbb{Z}_p^*$

Protokoll Diffie-Hellman Schlüsselaustausch

EINGABE: p, g

1. Alice wählt $\alpha \in_R \mathbb{Z}_{p-1}$ und schickt $g^\alpha \bmod p$ an Bob.
2. Bob wählt $\beta \in_R \mathbb{Z}_{p-1}$ und schickt $g^\beta \bmod p$ an Alice.
3. Alice berechnet $\left(g^\beta\right)^\alpha = g^{\alpha\beta}$, Bob analog $\left(g^\alpha\right)^\beta = g^{\alpha\beta}$.

Gemeinsamer geheimer DH-Schlüssel: $g^{\alpha\beta}$.

- Angreifer Eve erhält g, g^α, g^β.
- **Sicherheit:** Eve kann $g^{\alpha\beta}$ nicht von g^y, $y \in_R \mathbb{Z}_{p-1}$ unterscheiden.

Definition Decisional Diffie-Hellman (DDH)

Sei p prim, g Generator von $\mathbb{Z}_p^*$. Wir definieren die Sprache

$$\mathrm{DDH} := \{(g^\alpha, g^\beta, g^y) \mid g^y = g^{\alpha\beta}\}.$$

Das ElGamal Kryptosystem (1984)

Parameter des ElGamal Kryptosystems:

öffentlich: p prim, g Generator von $\mathbb{Z}_p^*$, g^a

geheim: $a \in \mathbb{Z}_{p-1}$

Algorithmus ElGamal Ver- und Entschlüsselung

- Verschlüsselung von $m \in \mathbb{Z}_p$ unter Verwendung von p, g, g^a.
  - Wähle $r \in_R \mathbb{Z}_{p-1}$.
  - Berechne $e(m) = (\gamma, \delta) = (g^r, m \cdot (g^a)^r) \in \mathbb{Z}_p^* \times \mathbb{Z}_p$.
- Entschlüsselung von $e(m)$ unter Verwendung von p, a.
  - Berechne $d(m) = \frac{\delta}{\gamma^a} = \frac{m \cdot g^{ar}}{g^{ar}} = m$.

Laufzeit:

- Verschlüsselung: $\mathcal{O}(\log r \cdot \log^2 p) = \mathcal{O}(\log^3 p)$
- Entschlüsselung: $\mathcal{O}(\log a \cdot \log^2 p) = \mathcal{O}(\log^3 p)$

Sicherheit von ElGamal

Intuitiv: Eve soll $\delta = m \cdot g^{ab}$ nicht von $x \in_R \mathbb{Z}_p$ unterscheiden können.

Protokoll Unterscheider

EINGABE: p, g, g^a

1. Eve wählt $m \in \mathbb{Z}_p^*$ und schickt m an Alice. (Man beachte: $m \neq 0$.)
2. Alice wählt $b \in \{0, 1\}$:
   - Falls $b = 0$: Sende Verschlüsselung $e(m) = (g^r, m \cdot g^{ar})$ an Eve zurück.
   - Falls $b = 1$: Sende $(g^r, x) \in_R \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ an Eve zurück.

Eves AUSGABE: $b' \in \{0, 1\}$

- Eve gewinnt das Spiel gdw $b' = b$.
- D.h. Eve muss δ von einer Zufallszahl x unterscheiden.

Definition Sprache ElGamal

Sei p prim und g ein Generator von $\mathbb{Z}_p^*$. Wir definieren

$$\text{ELGAMAL} := \{g^a, g^r, m, x \mid x = m \cdot g^{ar} \bmod p\}.$$

Sicherheitsbeweis per Reduktion

Satz Sicherheit von ElGamal unter DDH

Das ElGamal Kryptosystem ist sicher gegen polynomielle Angreifer unter der Annahme, dass DDH nicht effizient entscheidbar ist.

Logik des Beweises:

- Zeigen: DDH $\leq_p$ ELGAMAL
- D.h. jeder polynomielle Algorithmus für ELGAMAL liefert einen polynomiellen Algorithmus für DDH.
- **Annahme:** Es existiert ein polynomieller Angreifer A, der Verschlüsselungen von Zufallszahlen unterscheiden kann.
- Dann gibt es einen Algorithmus, der in polynomieller Zeit DH-Schlüssel $g^{\alpha\beta}$ von Zufallszahlen unterscheidet.
- **Widerspruch:** Nach Annahme gibt es keinen effizienten Algorithmus zum Entscheiden von DH-Schlüsseln $g^{\alpha\beta}$.
- Daher kann es auch keinen polynomiellen Angreifer A geben.

Reduktion f

Algorithmus M_f

EINGABE: $g^\alpha, g^\beta, g^y \in \mathbb{Z}_p^*$

1. Setze $g^a \leftarrow g^\alpha$ und $g^r \leftarrow g^\beta$.
2. Wähle $m \in_R \mathbb{Z}_p^*$.
3. Berechne $x = m \cdot g^y \bmod p$.

AUSGABE: g^a, g^r, m, x

Laufzeit:

- Eingabelänge: $\Omega(\log p)$
- Gesamtlaufzeit: $\mathcal{O}(\log^2(p))$

Korrektheit Reduktion: $w \in \text{DDH} \leq_p f(w) \in \text{ElGamal}$

Sei $(g^\alpha, g^\beta, g^y) \in \text{DDH}$.

- Dann gilt $g^y = g^{\alpha\beta} = g^{ar}$.
- Damit ist $x = m \cdot g^y = m \cdot g^{ar}$ korrekte Verschlüsselung von m.
- D.h. $(g^a, g^r, m, x) \in \text{ElGamal}$

Sei $f(g^\alpha, g^\beta, g^y) = (g^a, g^r, m, x) \in \text{ElGamal}$.

- Dann ist $x = m \cdot g^y$ eine korrekte Verschlüsselung von m.
- D.h. $d(m) = \frac{m \cdot g^y}{g^{ar}} = m$ und damit $g^y = g^{ar} = g^{\alpha\beta}$.
- Dann ist $(g^\alpha, g^\beta, g^y) \in \text{DDH}$.

Brechen von ElGamal ist nicht schwerer als DDH

Satz

ELGAMAL $\leq_p$ DDH

Beweis: Wir definieren die folgende Reduktion f.

Algorithmus M_f

EINGABE: $g^a, g^r, m, x \in \mathbb{Z}_p^*$

1. Setze $g^\alpha \leftarrow g^a$ und $g^\beta \leftarrow g^r$.
2. Berechne $g^y = \frac{x}{m}$.

AUSGABE: g^α, g^β, g^y

Laufzeit:

- Eingabelänge: $\Omega(\log p)$
- Laufzeit: $\mathcal{O}(\log^2 p)$

Korrektheit von f: $w \in$ ElGamal $\Leftrightarrow f(w) \in$ DDH

Sei $(g^a, g^r, m, x) \in$ ElGamal.

- Dann ist $x = m \cdot g^{ar}$ korrekte Verschlüsselung von m.
- Damit gilt $\frac{x}{m} = g^{ar} = g^{\alpha\beta} = g^y$.
- D.h. $(g^\alpha, g^\beta, g^y) \in$ DDH.

Sei $f(g^a, g^r, m, x) = (g^\alpha, g^\beta, g^y) \in$ DDH.

- Dann gilt $g^y = g^{\alpha\beta} = g^{ar}$.
- Damit folgt $x = m \cdot g^y = m \cdot g^{ar}$ ist Verschlüsselung von m.
- D.h. $(g^a, g^r, m, x) \in$ ElGamal.

Quadratische Reste

Definition Quadratischer Rest

Sei $n \in \mathbb{N}$. Ein Element $a \in \mathbb{Z}_n$ heißt *quadratischer Rest* in $\mathbb{Z}_n$, falls es ein $b \in \mathbb{Z}_n$ gibt mit $b^2 = a \bmod n$. Wir definieren

$$QR_n = \{a \in \mathbb{Z}_n^* \mid a \text{ ist ein quadratischer Rest } \} \text{ und } QNR_n = \mathbb{Z}_n^* \setminus QR.$$

Lemma Anzahl quadratischer Reste in primen Restklassen

Sei $p > 2$ prim. Dann gilt $|QR_p| = \frac{|\mathbb{Z}_p^*|}{2} = \frac{p-1}{2}$.

- Sei $a \in QR_p$. Dann gilt $a = b^2 = (-b)^2$.
- D.h. jeder quadratische Rest besitzt ≥ 2 Quadratwurzeln.
- Da $\mathbb{F}_p$ ein Körper ist, besitzt das Polynom $p(x) = x^2 - a$ hat höchstens zwei Nullstellen in $\mathbb{F}_p$. D.h. a hat ≤ 2 Quadratwurzeln.
- Damit bildet $f : \mathbb{Z}_p^* \to QR, x \mapsto x^2 \mod p$ jeweils genau zwei Elemente $\pm b$ auf einen quadratischen Rest $a \in QR$ ab.
- D.h. genau die Hälfte der Elemente in $\mathbb{Z}_p^*$ ist in QR.

Das Legendre Symbol

Definition Legendre Symbol

Sei $p > 2$ prim und $a \in \mathbb{N}$. Das *Legendre Symbol* ist definiert als

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{falls } p|a \\ 1 & \text{falls } (a \bmod p) \in QR_p \\ -1 & \text{falls } (a \bmod p) \in QNR_p. \end{cases}$$

Berechnung des Legendre Symbols

Satz

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

- Für $p|a$ sind beide Seiten Null. Gelte also $p \not| a$.
- Da $a^{p-1} = 1 \bmod p$, folgt $a^{\frac{p-1}{2}} = \pm 1$.
- Sei g Generator von $\mathbb{Z}_p^*$ und $a = g^j$ für ein $j \in \mathbb{Z}_{p-1}$.
- Es gilt für die linke Seite $a \in QR_p$ gdw. j gerade ist.
- Andererseits $a^{\frac{p-1}{2}} = g^{\frac{j(p-1)}{2}} = 1$ gdw $p-1$ teilt $\frac{j(p-1)}{2}$.
- Damit ist die rechte Seite ebenfalls 1 gdw j gerade ist.

Das Legendresymbol lässt sich in Zeit $\mathcal{O}(\log a \log^2 p)$ berechnen.

Eigenschaften des Legendre Symbols

Eigenschaften Quadratischer Reste

1. Multiplikativität: $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$
2. $(QR, \cdot)$ ist eine multiplikative Gruppe.
3. $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1 & \text{für } p = \pm 1 \bmod 8 \\ -1 & \text{für } p = \pm 3 \bmod 8. \end{cases}$

1. $\left(\frac{ab}{p}\right) = (ab)^{\frac{p-1}{2}} \bmod p = a^{\frac{p-1}{2}} \bmod p \cdot b^{\frac{p-1}{2}} \bmod p = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$
2. Übungsaufgabe
3. ohne Beweis (nicht-trivial)

Das Quadratische Reziprozitätsgesetz

Satz Quadratisches Reziprozitätsgesetz (Gauß)

Seien $p, q > 2$ prim. Dann gilt

$$\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)(q-1)}{4}} \left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{p}{q}\right) & \text{für } p = q = 3 \bmod 4 \\ \left(\frac{p}{q}\right) & \text{sonst.} \end{cases}$$

ohne Beweis (nicht-trivial)

- Liefert alternativen Algorithmus zur Berechnung des Legendre Symbols.
- **Bsp:**
$$\begin{aligned} \left(\frac{6}{11}\right) &= \left(\frac{3}{11}\right) \cdot \left(\frac{2}{11}\right) = -\left(\frac{11}{3}\right) \cdot (-1) \\ &= -\left(\frac{2}{3}\right) \cdot (-1) = -(-1) \cdot (-1) = (-1). \end{aligned}$$
- D.h. 6 ist quadratischer Nichtrest in $\mathbb{Z}_{11}^*$.
- Benötigen Primfaktorzerlegung, um das QR-Gesetz anzuwenden.

Das Jacobi Symbol

Definition Jacobi Symbol

Sei $n = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k} \in \mathbb{N}$ ungerade und $a \in \mathbb{N}$. Dann ist das *Jacobi Symbol* definiert als

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_i} \cdot \ldots \cdot \left(\frac{a}{p_k}\right)^{e_k}.$$

- **Warnung:** $\left(\frac{a}{n}\right) = 1$ impliziert nicht, dass $a \in QR_n$ ist.
- Bsp: $\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right) \cdot \left(\frac{2}{5}\right) = (-1)(-1) = 1$.
- D.h. $2 \in QNR_3$ und $2 \in QNR_5$. Damit besitzt $x^2 = 2$ weder Lösungen modulo 3 noch modulo 5.
- Nach CRT besitzt $x^2 = 2 \mod 15$ ebenfalls keine Lösung.

Verallgemeinerungen für das Jacobi Symbol

Satz

Für alle ungeraden m, n gilt

1. $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$.
2. $\left(\frac{m}{n}\right) = (-1)^{\frac{(m-1)(n-1)}{4}} \left(\frac{n}{m}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \text{für } m = n = 3 \bmod 4 \\ \left(\frac{n}{m}\right) & \textit{sonst}. \end{cases}$.

Wir beweisen hier nur das Analog des Reziprozitätsgesetzes.

- Falls $\mathrm{ggT}(m, n) > 1$, sind beide Seiten 0. Sei also $\mathrm{ggT}(m, n) = 1$.
- Schreiben Primfaktorzerlegung $m = p_1 \dots p_r$ und $n = q_1 \dots q_s$. (p_i's und q_j's können dabei jeweils mehrmals auftreten)
- Wandeln $\left(\frac{m}{n}\right) = \prod_{i,j} \left(\frac{p_i}{q_j}\right)$ zu $\left(\frac{n}{m}\right) = \prod_{i,j} \left(\frac{q_j}{p_i}\right)$ durch rs-malige Anwendung des Reziprozitätsgesetzes.
- Anzahl (-1) entspricht Anzahl Paare (i, j) mit $p_i = q_j = 3 \bmod 4$.
- D.h. $\left(\frac{m}{n}\right) = -\left(\frac{n}{m}\right)$ gdw. ungerade viele p_i, g_j kongruent 3 mod 4.
- Es gibt ungerade viele $p_i, g_j = 3 \bmod 4$ gdw. $m = n = 3 \bmod 4$ ist.

Rekursive Berechnung des Jacobi Symbols

Algorithmus Jacobi-Symbol

EINGABE: m, n mit n ungerade

1. Falls $ggT(m,n) > 1$, Ausgabe 0.
2. Sei $m = 2^k m'$ mit m' ungerade.
3. Ausgabe $(-1)^{\frac{k(n^2-1)}{8}} \cdot (-1)^{\frac{(m'-1)(n-1)}{4}} \cdot \text{Jacobi-Symbol}(n \bmod m', m')$

AUSGABE: $\left(\frac{m}{n}\right)$

Bsp: $\left(\frac{14}{15}\right) = \left(\frac{2}{15}\right) \cdot \left(\frac{7}{15}\right) = (-1) \cdot \left(\frac{15 \bmod 7}{7}\right) = (-1).$

- **Laufzeit:** Analog zum Euklidischen Algorithmus: $\mathcal{O}(\log \max\{m,n\})$ rekursive Aufrufe.
- Jeder Aufruf kostet $\mathcal{O}(\log^2 \max\{m,n\})$.
- **Korrektheit:** Für ungerades n gilt

$$\left(\frac{m}{n}\right) = \left(\frac{2}{n}\right)^k \cdot \left(\frac{m'}{n}\right) = \left(\frac{2}{n}\right)^k \cdot (-1)^{\frac{(m'-1)(n-1)}{4}} \left(\frac{n \bmod m'}{m'}\right).$$