

Vertiefungskurs Mathematik

Kryptographie

Kryptographie ist die Wissenschaft der Verschlüsselung von Informationen.

Diffie-Hellman Schlüsselaustausch

Alice und Bob vereinbaren eine Primzahl p und eine Generatorzahl $g \in \{1, 2, \dots, p-1\}$, am besten eine Primitivwurzel in \mathbb{Z}_p . Alice wählt geheim eine Zahl a aus, Bob geheim eine Zahl b mit $a, b \in \{1, 2, \dots, p-1\}$. Alice berechnet $A = g^a \bmod p$, $B = g^b \bmod p$. Dann tauschen beide A und B aus. Das bedeutet, p, g, A, B sind öffentlich bekannt, a kennt nur Alice, b nur Bob.

Beide können nun den gemeinsamen Schlüssel K berechnen:

$$\text{Alice: } B^a \equiv (g^b)^a \equiv g^{ba} \equiv K \bmod p$$

$$\text{Bob: } A^b \equiv (g^a)^b \equiv g^{ab} \equiv K \bmod p$$

Beispiel: $p = 13, g = 2$

Alice: $a = 5, \quad A \equiv 2^5 \equiv 6 \pmod{13}$

Bob: $b = 8, \quad B \equiv 2^8 \equiv 9 \pmod{13}$

Schlüssel berechnen:

Alice: $B^a \equiv 9^5 \equiv 3 \pmod{13} \Rightarrow K = 3$

Bob: $A^b \equiv 6^8 \equiv 3 \pmod{13} \Rightarrow K = 3$

Nebenrechnungen (mod 13):

$$2^1 \equiv 2$$

$$6^1 \equiv 6$$

$$2^2 \equiv 4$$

$$6^2 \equiv 36 \equiv -3$$

$$2^4 \equiv 3$$

$$6^4 \equiv 9 \equiv -4$$

$$6^8 \equiv 16 \equiv 3$$

$$9^1 \equiv -4$$

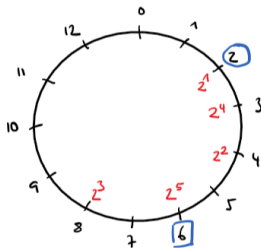
$$9^2 \equiv 16 \equiv 3$$

$$9^4 \equiv -4$$

Angriff auf Diffie-Hellman

Eve versucht aus den öffentlich bekannten Zahlen p, g, A, B den Schlüssel K zu berechnen. Eve weiß: $2^a \equiv 6 \pmod{13}$ und $2^b \equiv 9 \pmod{13}$.

Die Beschaffung des Exponenten a oder b heißt 'Berechnung des diskreten Logarithmus'. Bei kleinen Zahlen ist dies durch Ausprobieren möglich. g sollte Primitivwurzel sein, damit Angreifer möglichst viel ausprobieren muss.



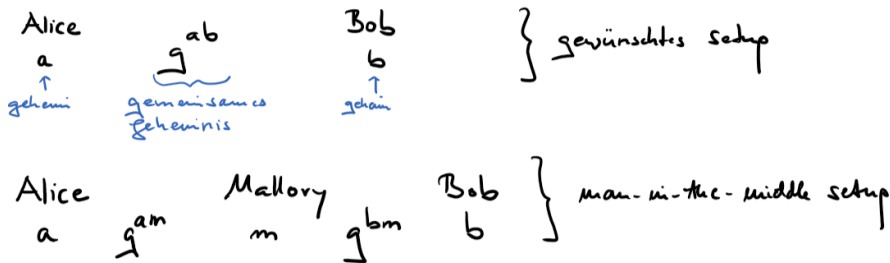
In der Praxis ist p eine Primzahl mit ca. 300 Stellen (eine Zahl größer als die Anzahl Atome im Weltall). Es ist kein effektives Verfahren zur Berechnung des diskreten Logarithmus bekannt.

Unterschied Logarithmus in \mathbb{Z}_p und Logarithmus in \mathbb{R}

Beim Rechnen in \mathbb{R} kann man den Wert des Exponenten abschätzen.

Beispiel: Gesucht ist $a \in \mathbb{R}$ mit $2^a = 12$. Wegen $2^3 = 8$ und $2^4 = 16$ folgt: $3 < a < 4$. Diese Folgerung ist im diskreten Fall nicht möglich. Die Potenzen der Generatorzahl springen wie zufällig in dem Restklassenring herum.

Man-in-the-middle-Angriff auf Diffie-Hellman



Mallory kontrolliert das Netzwerk. Er gibt sich gegenüber Alice als Bob aus und gegenüber Bob als Alice. Mit beiden vereinbart er getrennte Schlüssel g^{am} und g^{bm} .

RSA-Verfahren

Alice wählt zwei Primzahlen p und q und berechnet $m = p \cdot q$ und $\tilde{m} = (p - 1)(q - 1)$. Alice wählt Verschlüsselungsexponent e mit $1 < e < \tilde{m}$ und $\text{ggT}(e, \tilde{m}) = 1$ und berechnet Entschlüsselungsexponent d mit $\bar{d} = \frac{1}{e}$ in $\mathbb{Z}_{\tilde{m}}$.

Dann ist der öffentliche Schlüssel (m, e) und der private Schlüssel (m, d) .

Für die zu verschlüsselnde Nachricht muss gelten: $0 < n < m$.

Bob verschlüsselt n : $N = n^e \bmod m$

Alice entschlüsselt N : $n = N^d \bmod m$

Beispiel: $p = 7, q = 13 \Rightarrow m = 91$ (RSA-Modul), $\tilde{m} = 72$
 $e = 11, \text{ggT}(e, \tilde{m}) = 1$.

Berechnung von d : $\bar{d} = \frac{1}{\bar{e}}$ in $\mathbb{Z}_{\tilde{m}}$, d.h. $d \cdot e \equiv 1 \pmod{\tilde{m}}$, also
 $d \cdot e - 1 = k \cdot \tilde{m}$ für ein $k \in \mathbb{Z}$.

Wir lösen die diophantische Gleichung $d \cdot 11 - k \cdot 72 = 1$ mit dem
Erweiterten Euklidischen Algorithmus und erhalten $d = 59$. Der öffentliche
Schlüssel ist $(91, 11)$, der private Schlüssel ist $(91, 59)$.

Bob verschlüsselt $n = 10$: $N = 10^{11} \pmod{91} = 82$

Alice entschlüsselt $N = 82$: $n = 82^{59} \pmod{91} = 10$

Nebenrechnungen (mod 91)

$$10^1 \equiv 10$$

$$10^2 \equiv 9$$

$$10^4 \equiv -10$$

$$10^8 \equiv 100 \equiv 9$$

$$10^{11} \equiv 9 \cdot 9 \cdot 10 \equiv -100 \equiv 82$$

$$82^1 \equiv -9$$

$$82^2 \equiv 81 \equiv -10$$

$$82^4 \equiv 9$$

$$82^8 \equiv -10$$

$$82^{16} \equiv 9$$

$$82^{32} \equiv -10$$

$$82^{59} \equiv 82^{32+16+8+2+1} \equiv -10 \cdot 9 \cdot -10 \cdot -10 \cdot -9 \equiv 10$$

Beweis, dass RSA-Entschlüsselung funktioniert

Wir zeigen $n = N^d \bmod m$, indem wir zeigen: $(n^e)^d \equiv n \bmod m$. Wir zeigen zunächst:

$$(1) : (n^e)^d \equiv n \bmod p \quad \text{und} \quad (2) : (n^e)^d \equiv n \bmod q \quad (2)$$

Es ist $\bar{d} = \frac{1}{e}$ in $\mathbb{Z}_{\tilde{m}}$, also gilt: $de - 1 = k\tilde{m}$. Daraus folgt:

$de = 1 + k\tilde{m} = 1 + k(p-1)(q-1)$. Beweis von (1):

Fall 1: $p|n \Rightarrow n \equiv 0 \bmod p \Rightarrow (n^e)^d \equiv 0 \bmod p$.

Fall 2: $p \nmid n$

$\Rightarrow (n^e)^d = n^{ed} = n^{1+k(p-1)(q-1)} = n \cdot (n^{p-1})^{k(q-1)} \equiv n \bmod p$, da $n^{p-1} \equiv 1 \bmod p$ nach dem kleinen Satz von Fermat.

Beweis von (2) analog.

Aus (1) und (2) folgt: $(n^e)^d - n = k_1 \cdot p = k_2 \cdot q$. mit geeigneten $k_1, k_2 \in \mathbb{Z}$. Da p, q Primzahlen, steckt k_1 in q und k_2 in p . Also gilt: $(n^e)^d - n = k_3 \cdot p \cdot q$. Daraus folgt $(n^e)^d \equiv n \bmod m$. □

Angriff auf das RSA-Verfahren

Die Sicherheit des Verfahrens hängt davon ab, dass der Angreifer das öffentlich bekannte m nicht in die beiden Primfaktoren p und q zerlegen kann. Sonst könnte er \tilde{m} berechnen und dann auch das Inverse zu dem öffentlichen e in $\mathbb{Z}_{\tilde{m}}$.

Aufwand für die Faktorensuche

Wenn wir versuchen, die Faktoren einer Primzahl mit 300 Stellen zu finden, testen wir nur Primzahlen, die kleiner als $\sqrt{10^{300}} = 10^{150}$ sind. Nach der Abschätzung von Euler gilt für große n , dass es ca. $\frac{n}{\ln(n)}$ Primzahlen unterhalb von n gibt.

$\ln(10^{150}) = 150 \cdot \ln(10) \approx 150 \cdot 2.3 = 345.4$. Also müssen wir $\frac{10^{150}}{345.5} \approx 3 \cdot 10^{147}$ Kandidaten testen.

Annahme: 1 Computer schafft 10^{12} Prüfungen pro Sekunde (1 Million Millionen). Das sind im Dauerbetrieb pro Jahr: $10^{12} \cdot 60 \cdot 60 \cdot 24 \cdot 365 \approx 3 \cdot 10^{19}$ Prüfungen. Das bedeutet $\frac{3 \cdot 10^{147}}{3 \cdot 10^{19}} = 10^{128}$ Jahre für alle Prüfungen (Alter des Weltalls: ca. 10^{10} Jahre).

Wenn jeder Mensch einen Computer beisteuern würde und manche zwei kämen wir auf 10 Milliarden = 10^{10} Computer. Wenn jeder dieser Computer 1000 mal schneller wäre, würde es immer noch $10^{128-10-3} = 10^{115}$ Jahre dauern.

Kryptographische Hashfunktionen

Hashfunktionen bilden Eingabewerte (z.B. ein Text oder eine Datei) auf einen Wert fester Länge ab, den *Hash* der Eingabe. Beispiel: *SHA-256* bildet Eingaben auf eine Bitfolge der Länge 256 ab.

Kryptographische Hashfunktionen sind kollisionsresistente Einwegfunktionen. Kollisionsresistent bedeutet, es ist praktisch unmöglich, zwei Eingaben zu finden, die denselben Hash ergeben. Einwegfunktion bedeutet, es ist praktisch unmöglich, aus dem Hashwert den Eingabewert zu rekonstruieren.

Digitale Signatur

Mit dem RSA-Verfahren und einer kryptographischen Hashfunktion kann eine digitale Signatur erstellt werden.

Alice m_A, e_A, d_A

$$\left. \begin{array}{l} N \\ \downarrow \text{berechnet} \\ \text{Sig} = \text{hash}(N)^{d_A} \bmod m_A \end{array} \right\} \xrightarrow[N, \text{Sig}]{\text{Schickt an Bob}}$$

Bob

$$(\text{hash}(N))^{e_A} \bmod m_A \stackrel{?}{=} \text{Sig} \quad \leftarrow \text{Bob prüft}$$

Falls Prüfung ok, ist Bob sicher, dass N von Alice stammt.

Durch eine digitale Signatur wird der Diffie-Hellman Schlüsselaustausch vor einem Man-in-the-middle-Angriff geschützt.

Alice m_A, e_A, d_A
a

p, g

Bob m_B, e_B, d_B
b

berechnet

berechnet

$$A = g^a \bmod p$$

$$\text{Sig}(A) = \text{hash}(A)^{d_A} \bmod m_A$$

$$B = g^b \bmod p$$

$$\text{Sig}(B) = \text{hash}(B)^{d_B} \bmod m_B$$



Austausch von $A, B, \text{Sig}(A), \text{Sig}(B)$

$$\text{hash}(B) \stackrel{?}{=} \text{Sig}(B)^{e_B} \bmod m_B$$

prüfen

$$\text{hash}(A) \stackrel{?}{=} \text{Sig}(A)^{e_A} \bmod m_A$$

$$A^b \equiv B^a \equiv g^{ab} \bmod p$$

gemeinsames Geheimnis, wenn Prüfung ok.