Diffie-Hellmann Key Exchange

Heinrich Moser

http://www.heinzi.at

Schlüsselübermittlung (key exchange)

Problem:

- Symmetrischer Schlüssel muss ausgetauscht werden

Lösungen:

- verschiedene Kanäle
- asymmetrische Verschlüsselung
- Schlüsselaustauschprotokoll

Diffie-Hellmann

- Sitzungsschlüssel wird von A und B gemeinsam erzeugt
- erster Public-Key Algorithmus (1976)
- nur für Schlüsselaustausch (keine Verschlüsselung)
- ein paar Jahre zuvor vom GCHQ entwickelt

Diffie-Hellmann

gemeinsam: g, p

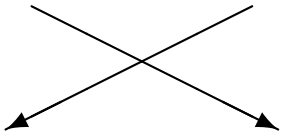
	A	B	
wählt	x	y	
berechnet	g^x	g^y	$\bmod p$
berechnet	$(g^y)^x$	$= (g^x)^y$	$= g^{xy} \bmod p$

symmetrischer Schlüssel: $g^{xy} \bmod p$

Diffie-Hellmann

gemeinsam: g, p

	A	B	
wählt	x	y	
berechnet	g^x	g^y	mod p
berechnet	$(g^y)^x$	$= (g^x)^y$	$= g^{xy}$ mod p



symmetrischer Schlüssel: g^{xy} mod p

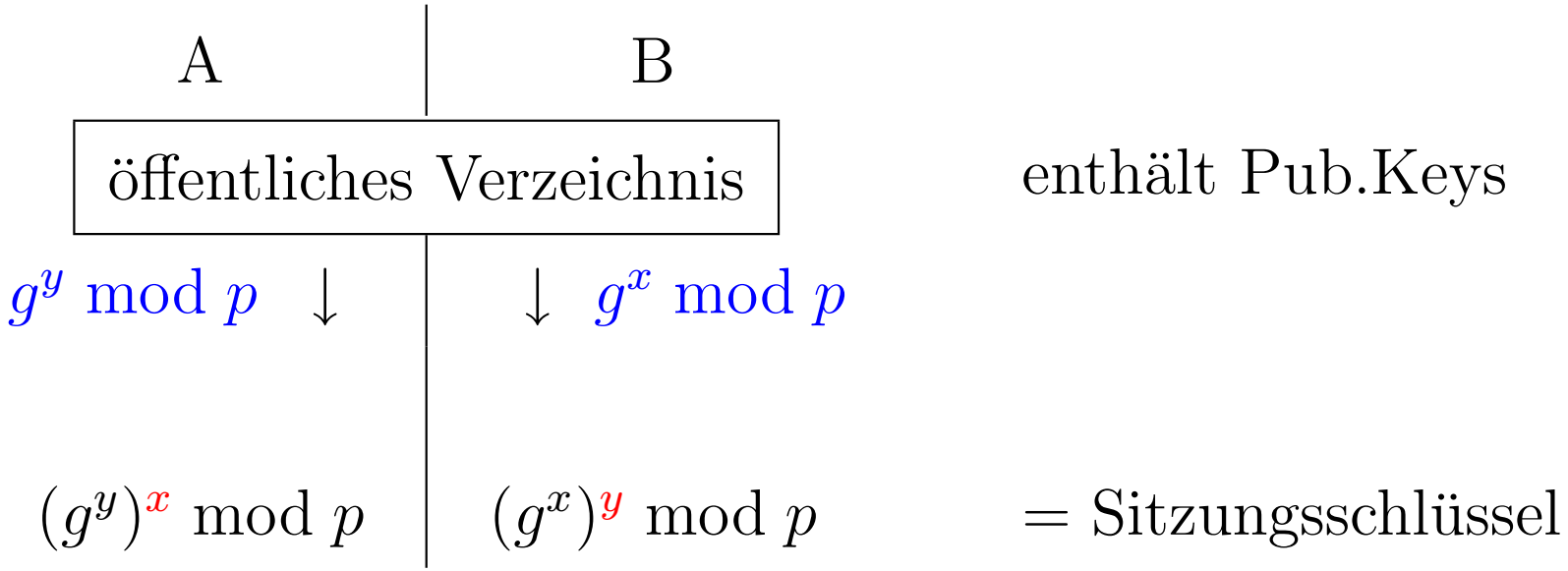
Algorithmus

1. A und B wählen gemeinsam öffentliche Zahlen
   - p (große Primzahl)
   - g (primitiv bzgl. p)
2. A wählt x ($< p$, geheim), schickt $X = g^x \bmod p$ an B
3. B wählt y ($< p$, geheim), schickt $Y = g^y \bmod p$ an A
4. A berechnet $Y^x \bmod p$ $(= g^{xy} \bmod p)$
5. B berechnet $X^y \bmod p$ $(= g^{xy} \bmod p)$

$g^{xy} \bmod p$ ist der gemeinsame symmetrische Schlüssel.

Public-Key Infrastruktur

- $g^x \bmod p$ ist der öffentliche Schlüssel.
- x ist der private Schlüssel.



Warum sicher?

- $X = g^x \qquad \rightarrow \qquad x = log_g X$

- Im Körper $\mathbb{Z}_p = (\{0, 1, \ldots, p-1\}, *, +)$, p Primzahl:
  $X = g^x \qquad \rightarrow$ Bestimmung von x „schwer“
  (Bedingung: auch $\frac{p-1}{2}$ ist eine Primzahl).

- mindestens so schwer wie Primzahlfaktorisierung (RSA)

Exponentialfunktion

- $\exp_g : \mathbb{N} \rightarrow \mathbb{Z}_p$
- g „primitiv bzgl. p“, „Primitivwurzel“, „Erzeuger“ $\Leftrightarrow$ $\{g^x | x \in \{1, \ldots, p-1\}\} = \{1, \ldots, p-1\}$
- square-and-multiply: g^{13} (13 = binär 1101)

  $g^{13} = g^{2^3+2^2+2^0} = g^{2^3} * g^{2^2} * g^{2^0} = ((g^2)^2)^2 * (g^2)^2 * g$

  - höchstens $2 * log_2(x)$ Multiplikationen

Primitivwurzel

$$\mathbb{Z}_5 \ (p = 5)$$

	x =	1	2	3	4
$g = 1$	$1^x =$	1	1	1	1
$g = 2$	$2^x =$	2	4	3	1
$g = 3$	$3^x =$	3	4	2	1
$g = 4$	$4^x =$	4	1	4	1

Diskreter Logarithmus

- $\log_g : \mathbb{Z}_p \to \mathbb{N}$
- Keine Annäherung $\quad (x < y \not\Rightarrow \log_g(x) < \log_g(y))$
- Lösungen: (sub-)exponentialer Aufwand bzgl. Bitlänge
  - durchprobieren
  - baby-step giant-step (hoher Speicherbedarf)
  - Pohlig-Hellmann (kleine Primteiler von $p-1$)
  - Pollard-Rho
  - Index-Calculus

Attacken

- **Man-in-the-middle Attack**
  Abhilfe: Station-to-Station Protokoll (Diffie, van Oorschot, Wiener: 1992) oder Interlock Protokoll (Rivest, Shamir: 1984)
- **Bekannte Verfahren**
  Abhilfe: p gut wählen
- **Quantencomputer**
- **Knacken des symmetrischen Verfahrens**

Attacken (Zukunft)

- **allgemeine Lösung diskreter Logarithmus**

  Gegeben: g^x

  Gesucht: x

- **allgemeine Lösung Diffie-Hellmann-Problem**

  Gegeben: g^x, g^y

  Gesucht: g^{xy}

Unterlagen

- W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22: 644-654, 1976.
- Patent 1997 abgelaufen
- ElGamal-Algorithmus (gleiches Prinzip)
- RFC 2631 (Juni 1999), basierend auf ANSI X9.42
- Vortragsfolien: `http://www.heinzi.at`

FRAGEN?