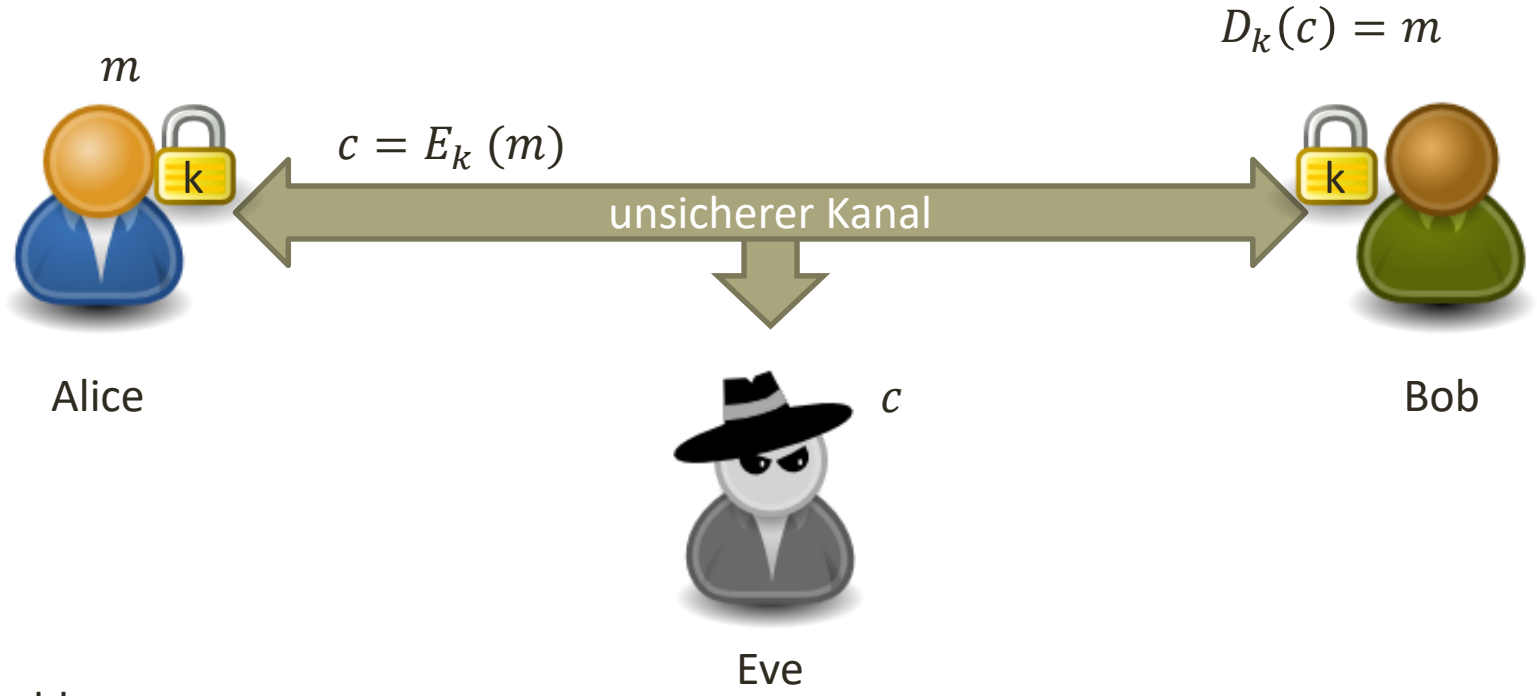
Diffie-Hellmann in modernen Kryptosystemen

Prof. Dr.-Ing. Johann Uhrmann

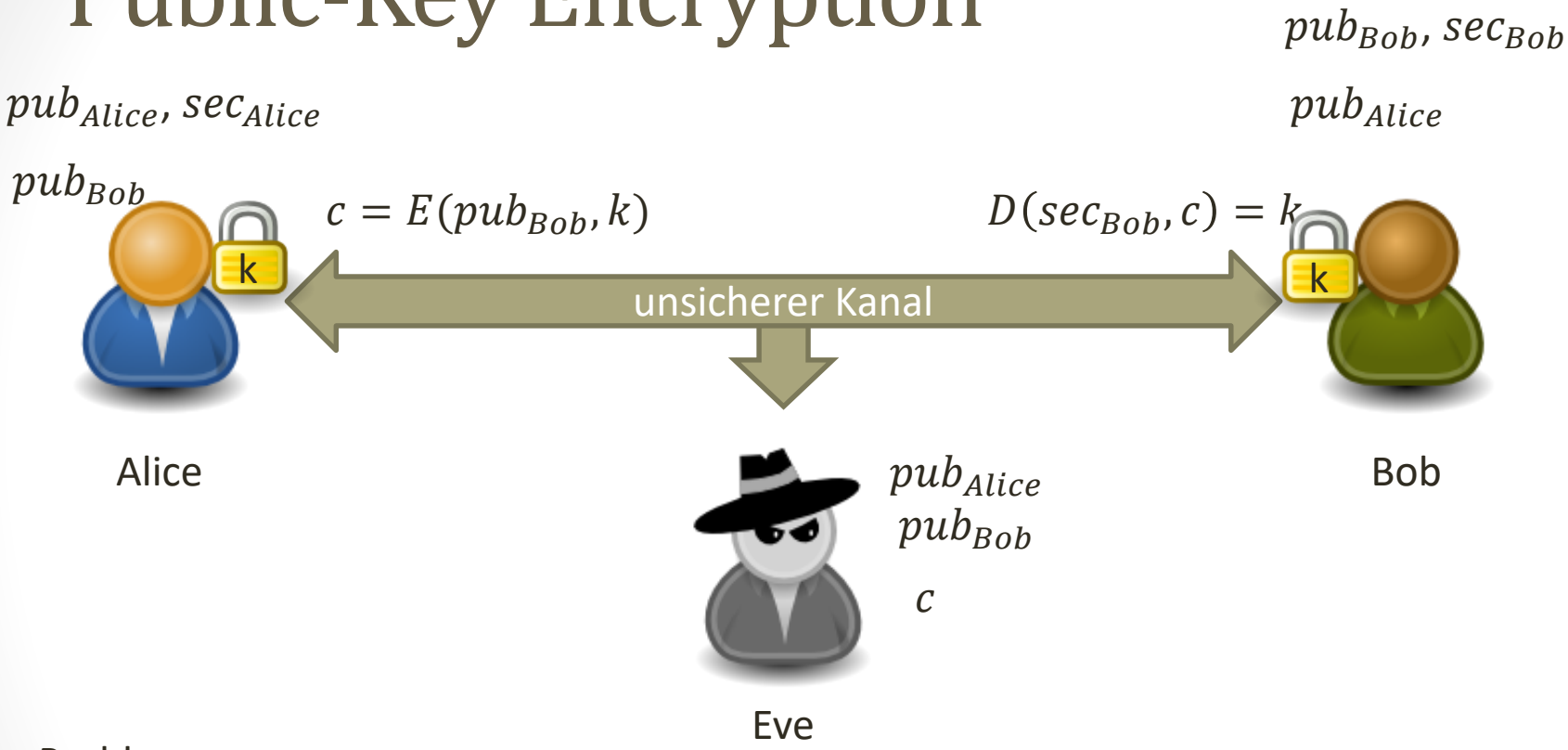
Kommunikation über unsicheren Kanal



Probleme:

- Wie können Alice und Bob sicherstellen, dass nur sie den geheimen Schlüssel k kennen?
- Wenn k über einen sicheren Kanal ausgetauscht werden muss, warum nicht dann gleich die ganze Nachricht m?
- Wie einen sicherer Kanal für jeden Kommunikationspartner finden?

Eine Möglichkeit: Public-Key Encryption



Probleme:

- Wie kann sichergestellt werden, dass der Public Key zum angeblichen Eigentümer gehört? ➔ gelöst durch Digitale Signaturen
- Der Secret Key hat eine lange Lebensdauer und wird oft wiederverwendet. Wenn er bekannt wird, dann kann aus c im Nachhinein k berechnet werden.

Das Problem: Kompromittierte Schlüssel

- Wird verschlüsselte Kommunikation aufgezeichnet und später der Schlüssel bekannt, dann können die aufgezeichneten Daten durch Unbefugte gelesen werden
- Beispiele für derartige Angriffe:

Angriff	ermöglicht	veröffentlicht
Heartbleed	Auslesen von Speicher via OpenSSL-Lücke, u.U. Private Key des Serverzertifikats	2014-04-07
Bleichenbacher	Ausnutzen von unterschiedlichen Fehlermeldungen zur Berechnung des Session-Keys („chosen plaintext“ mit SSLv2)	1998
DROWN	Verwendung des Bleichenbacher-Angriffs auf ca. 25% - 30% aller SSL-gesicherten Seiten im Internet	2016-03-01
nation state	Beschlagnahmung des Systems mitsamt Schlüsselmaterial	

Lösung: Diffie-Hellman

unsicherer Kanal

Whitfield Diffie

Martin Hellmann

Idee: Finde und benutze Rechenoperationen $\odot$, die

- kommutativ sind: $A \odot B \odot C = A \odot C \odot B$
- einfach durchzuführen, aber sehr schwierig umzukehren sind
  - $A \odot B \rightarrow$X einfach
  - Berechnung von B aus A und X schwierig

Diffie-Hellman

Whitfield Diffie

p

a

$A = p \odot a$

B

$k = B \odot a$

p

A, B

Eve

Martin Hellmann

p

b

$B = p \odot b$

A

$k = A \odot b$

Beide Kommunikationspartner

- einigen sich auf einen Operator p der Rechenoperation (public)
- wählen den anderen Operator als Secret zufällig (private)
- führen die Operation durch und übertragen das Ergebnis
- berechnen aus dem Ergebnis und dem eigenen Secret den gemeinsamen, geheimen Schlüssel $k = p \odot a \odot b = p \odot b \odot a = A \odot b = B \odot a$

Beispiel: „klassischer DH“

- Mathematisches Konstrukt: Primzahlenkörper

➔Potenzieren ist einfach

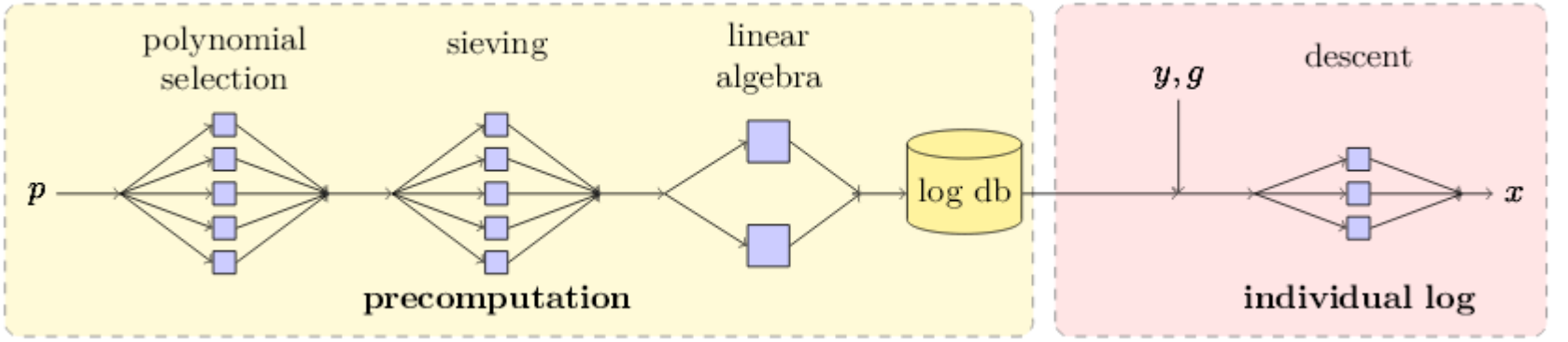
➔Logarithmieren ist schwierig

- Public-Informationen:
  - Primzahl p
  - „Generator“ $g < p$

- Alice wählt a, berechnet $A = g^a \bmod p$, sendet A
- Bob wählt b, berechnet $B = g^b \bmod p$, sendet B
- beide berechnen $k = g^{ab} \bmod p = A^b \bmod p = B^a \bmod p$

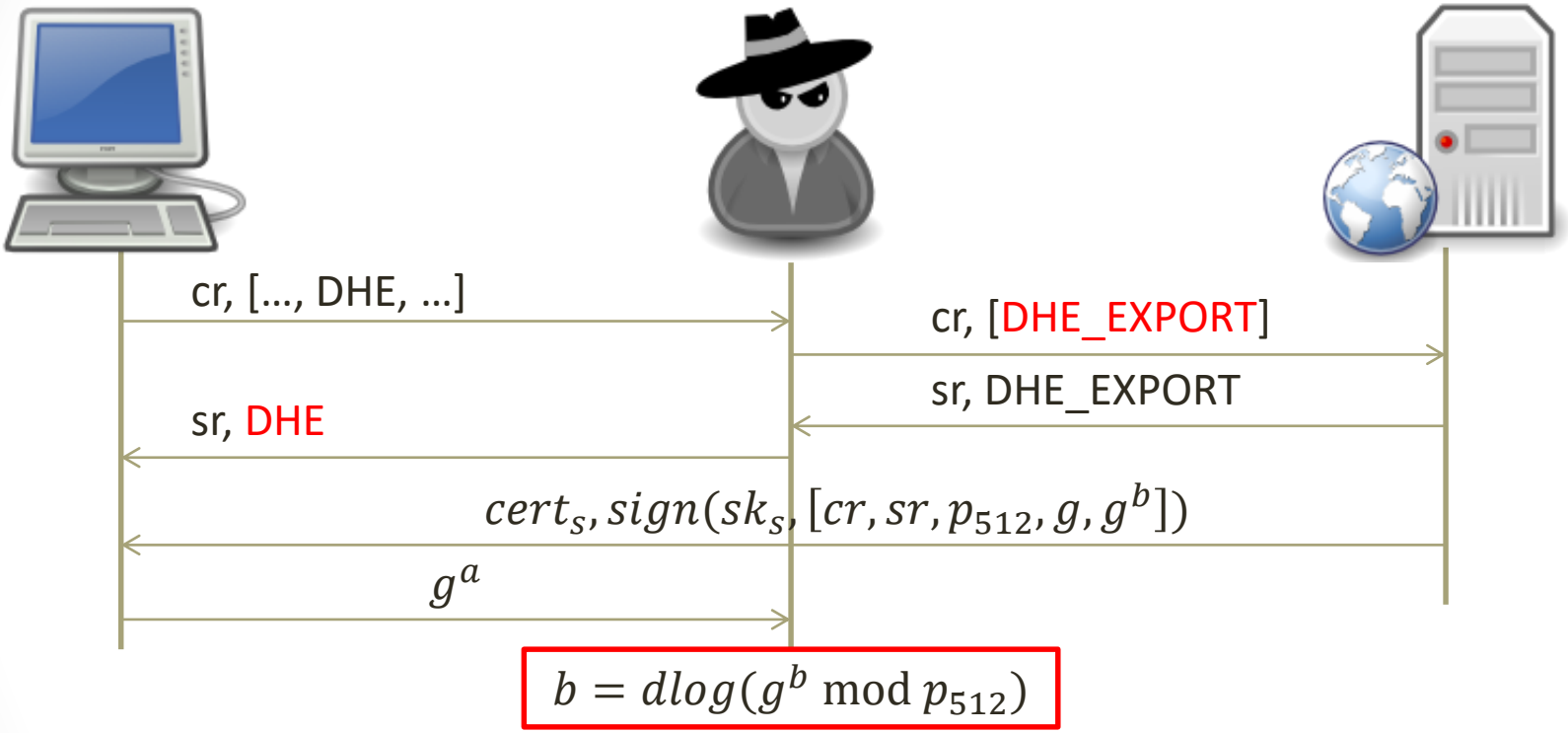
Sicherheit von klassischem DH

- Sicherheit von Diffie-Hellman auf Primzahlenkörpern hängt stark von der Länge der Primzahl ab
- bis 512 bit ➔ bereits gebrochen
- 768 bit ➔ mit moderatem Aufwand zu brechen
- 1024 bit ➔ vermutlich mit staatlicher Unterstützung zu brechen
- Number Field Sieve:



[weakdh.org]

LogJam-Angriff



Client, Server und Angreifer berechnen jetzt die Session-Keys aus g^{ab}, cr und sr.

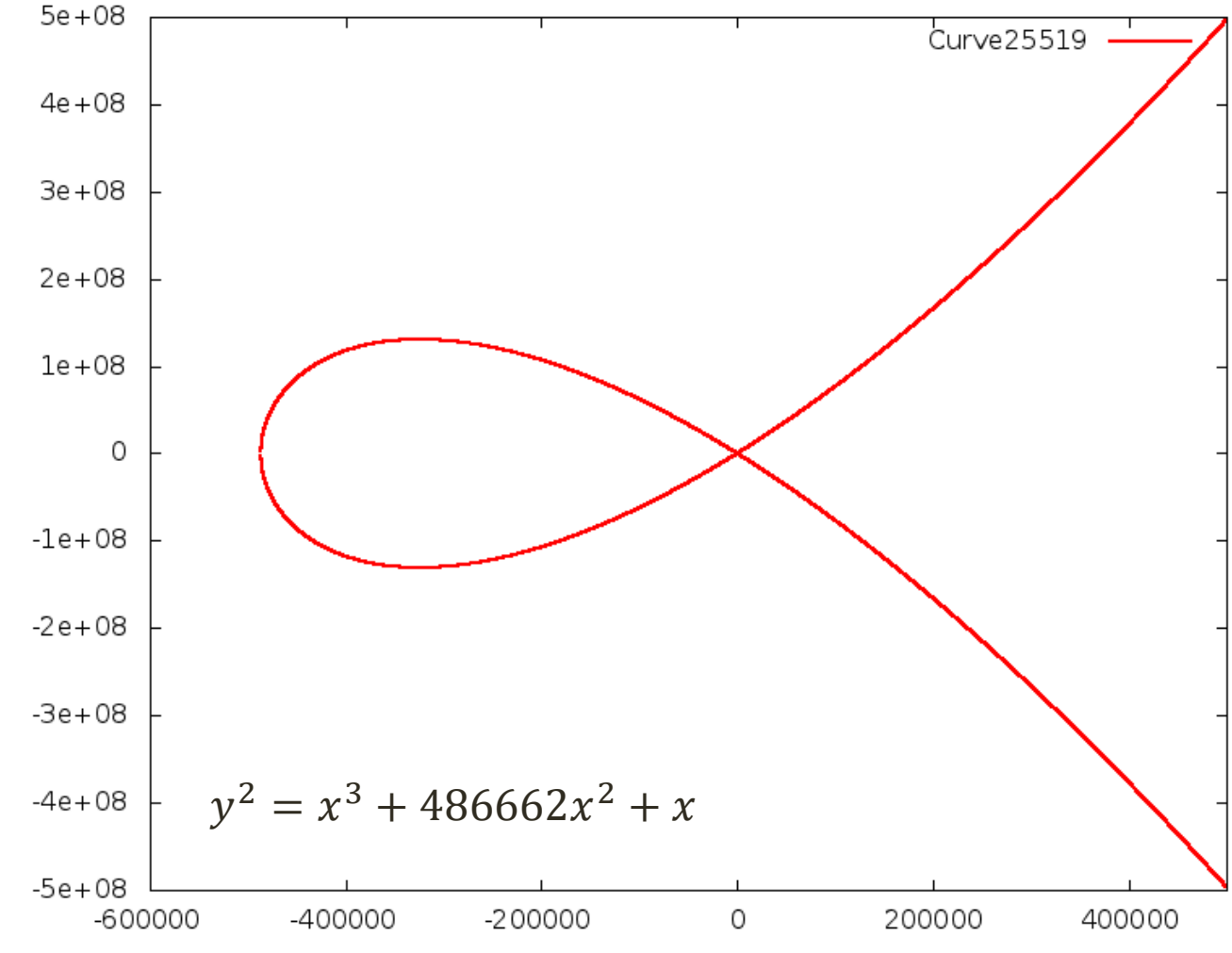
Elliptic Curves

- neue, alternative Menge, auf der die Rechenoperation durchgeführt wird

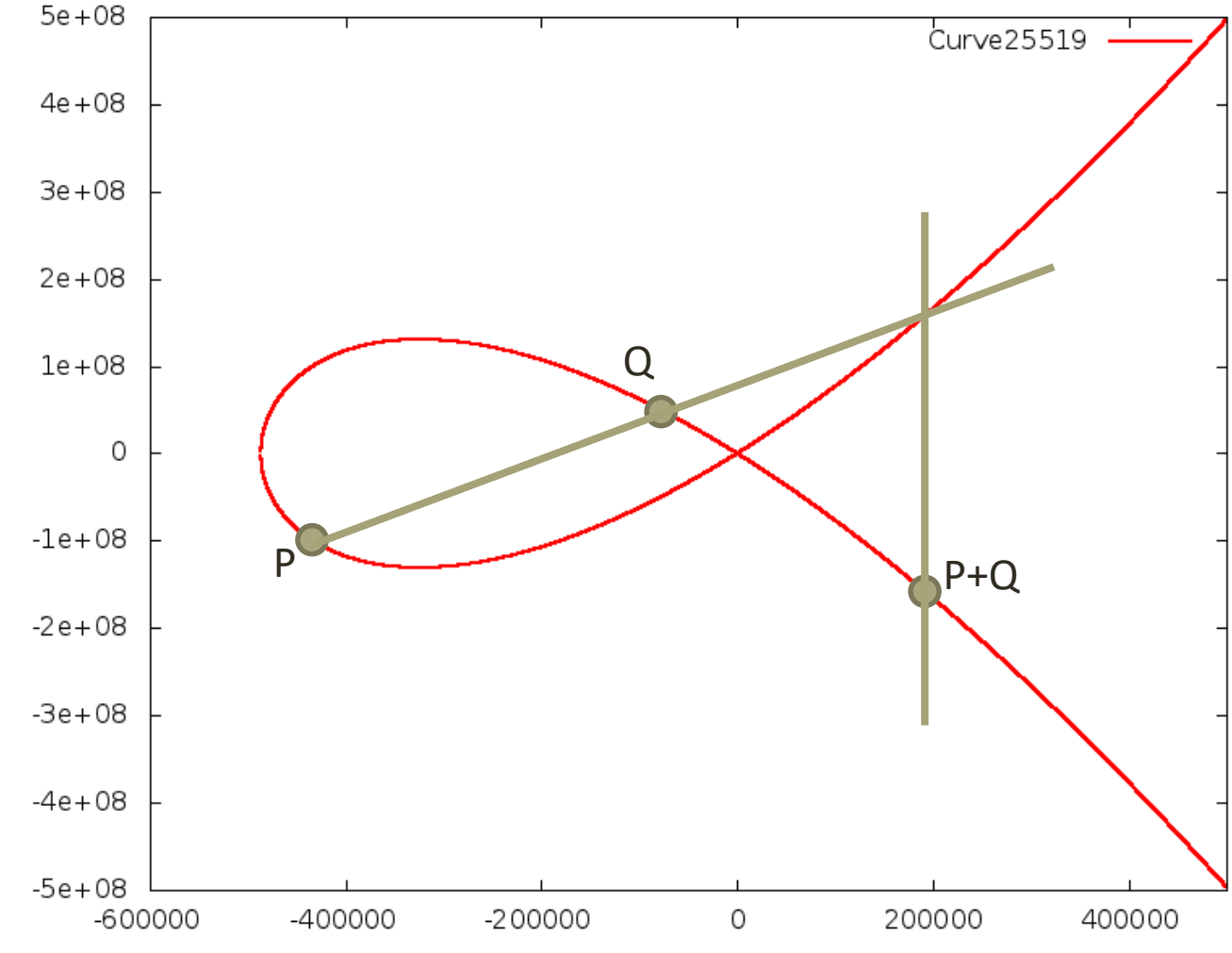
- Operation für Diffie-Hellman-Algorithmus:
  Multiplikation von Punkten auf einer Elliptic Curve mit einem Integer

- Public Informationen:
  - Definition der Kurve
  - ein Startpunkt G auf dieser Kurve
  - ein errechneter Punkt $Q = n\ G$
- Private Information: die Zahl n

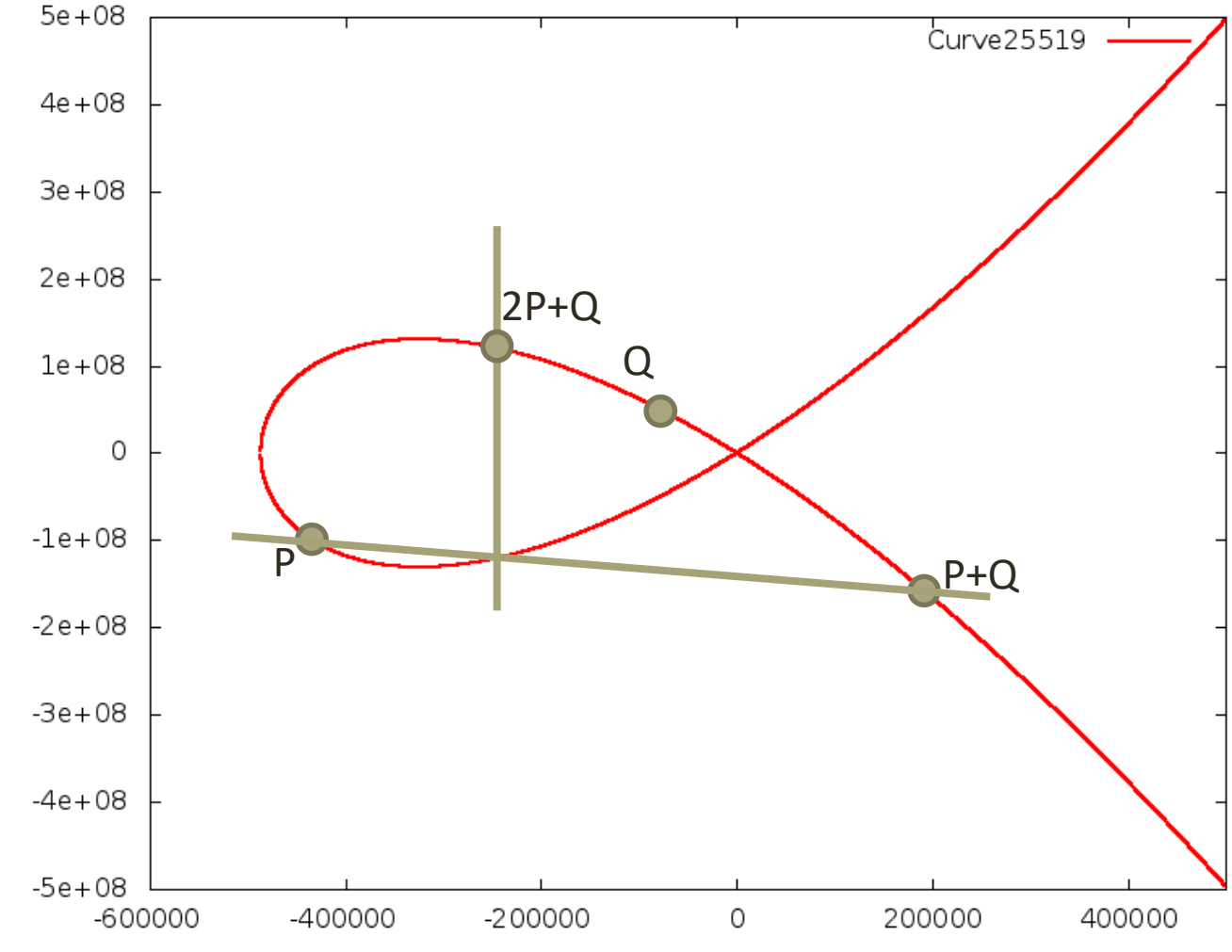
Elliptic Curves



Addition auf Elliptic Curves



Addition auf Elliptic Curves



EC-Diffie-Hellman

Q_A

Q_B

unsicherer Kanal

Curve25519
Startpunkt G (x=9)

d_A zufällig

$Q_A = d_A G$

Q_B

$X = d_A Q_B = d_A d_B G$

Eve
Curve25519
Startpunkt G (x=9)

$Q_A \quad Q_B$

Curve25519
Startpunkt G (x=9)

d_B zufällig

$Q_B = d_B G$

Q_A

$X = d_B Q_A = d_B d_A G$

Zusammenfassung

- Diffie-Hellman erlaubt zwei Kommunikationspartnern das Generieren von einem geheimen Schlüssel über eine unsichere Verbindung
- Für Man-in-the-Middle-Angriffe sind zusätzlich Zertifikate und Signaturen notwendig
- Elliptic Curves lösen Primzahlenkörper als Konstrukt ab, um geeignete Einwegfunktionen zu berechnen