# Technikseminar SS2012

ECC - Elliptic Curve Cryptography
Kryptosysteme basierend auf elliptischen Kurven

11.06.2012

# Gliederung

- Was ist ECC?
- ECC und andere Verfahren

- Diffie-Hellman-Schlüsselaustausch
  → Funktionsweise
  → Protokoll

- Elliptische Kurven
  - → Was ist eine elliptische Kurve
  - → Punkt Addition/Multiplikation
  - → Endliche Gruppen
  - → DL-Problem an ECC

- Sicherheit ECC (Vergleich mit RSA)

# Was ist ECC?

- Modernes, asymmetrisches Kryptographieverfahren
- Basiert auf dem Problem des diskreten Logarithmus
  → Einwegfunktionen
- Anfänge in den 1980er Jahren
- Vorgeschlagen von Victor Miller und Neal Koblitz

- Unterschied zu anderen Kryptosystemen:

  Zur Verschlüsselung werden Punkte genutzt anstatt einfache Zahlen!

# ECC und andere Verfahren

- DH Protokolle werden mit ECC „modifiziert“
- ECDH zum Schlüsselaustausch
- ECDSA als Variante der Digitalen Signatur
- Vorteil: Keine effektiven Algorithmen zur Überwindung der Einwegfunktion des ECC bekannt

# Diffie-Hellman-Schlüsselaustausch (mit natürlichen Zahlen)

- Bildung endlicher Gruppen über Primzahlen durch Restwertberechnung
- Wählen einer Primzahl p
- Gruppe G(p) ist endlich und sich wiederholend
- Wählen eines Elements α
- $\alpha^a \equiv r \bmod p$ für $a=1...\infty$
  → Beispiel Gruppe

# Vorgehensweise

- Wählen einer öffentlichen Primzahl p
- Wählen eines öffentlichen Generators α

Für beide Parteien gleich

- Wählen des privaten Schlüssels a und erzeugen des Restwertes A in $\alpha^a \equiv A \bmod p$
- A stellt den öffentlichen Schlüssel dar
- Beide Parteien tauschen ihre öffentlichen Schlüssel A (für Alice) und B (für Bob) aus
- Alice' Rechenoperation: $B^a \equiv K \bmod p$
- Bobs Rechenoperation: $A^b \equiv K \bmod p$

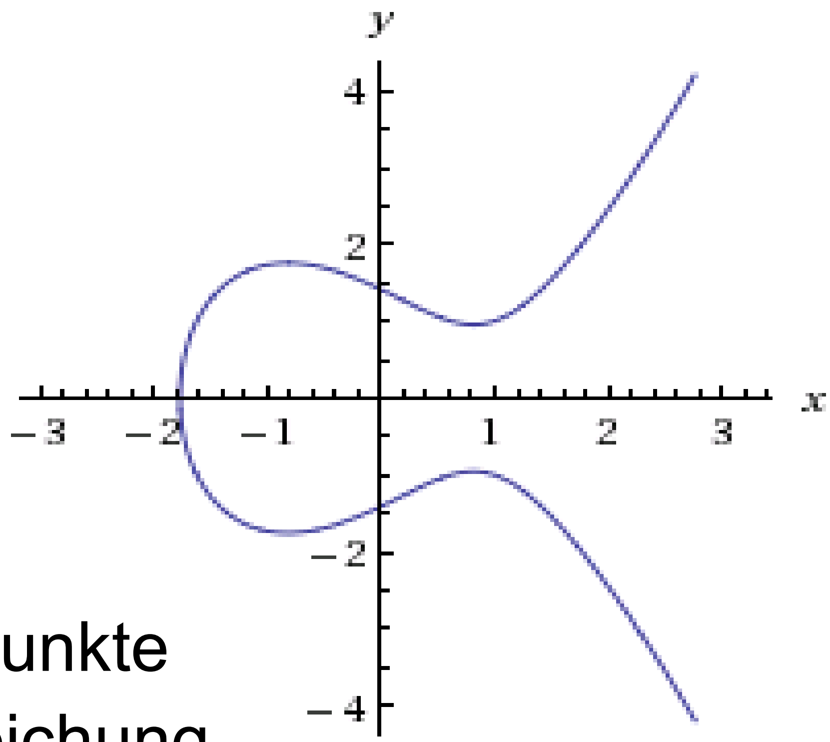
  → Gemeinsamer Schlüssel „K“ bestimmt

# Elliptische Kurven

- Grundform der elliptischen Kurve:
  E: $y^2=x^3+ax+b \in \mathbb{R}^2$

- Kurve über endlichem Körper:
  $G_f(F_p)= y^2=x^3+ax+b \bmod p$
  -Hinweis: $a,b \in F_p$

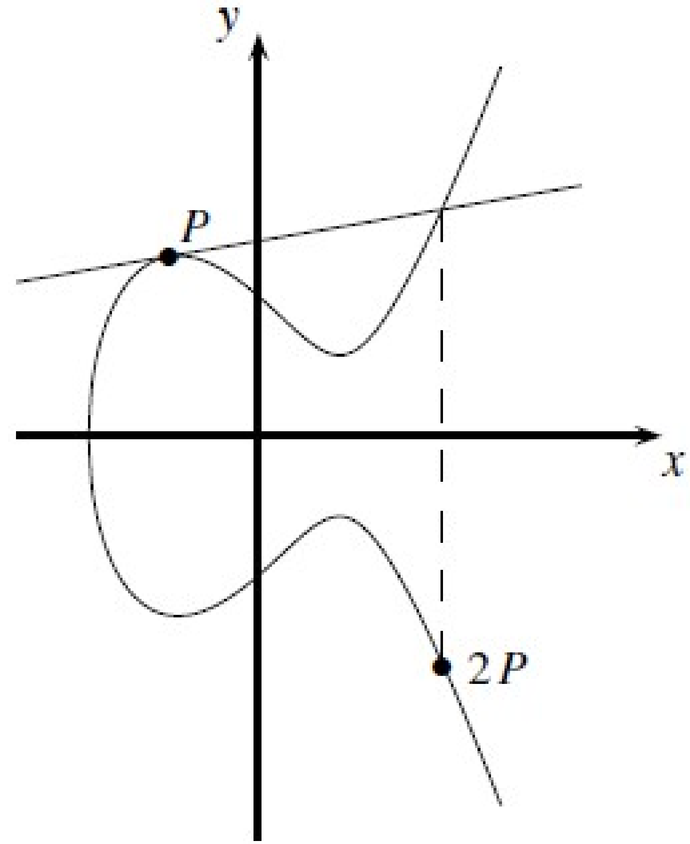
  → Nur eine endliche Menge Punkte auf der Kurve lösen die Gleichung




Wie finde ich die Punkte auf der Kurve?

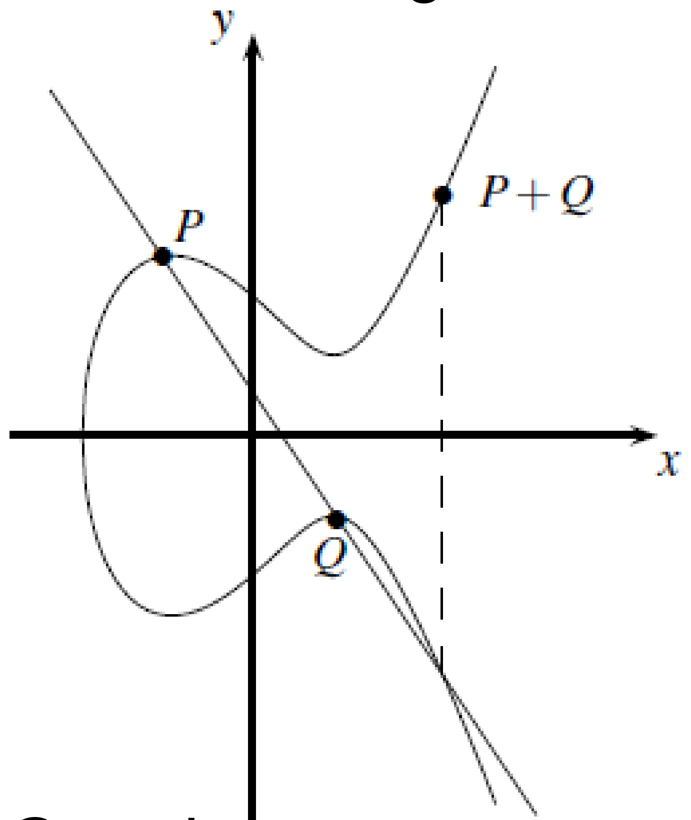
# Punktverdoppelung

- Mit nur einem bekannten Punkt einen weiteren Punkt auf der Kurve erzeugen
- <u>Graphisch:</u>
  Schnittpunkt der Tangente an P mit Kurve an x-Achse spiegeln
- <u>Rechnerisch:</u>
  $x_3 = s^2 - x_1 - x_2 \mod p$
  $y_3 = s(x_1 - x_3) - y_1 \mod p$
- Bei Punktverdoppelung:
  $S = (3x_1^2 + a) / 2y_1$



# Punktaddition

- Mit 2 bekannten Punkten einen weiteren erzeugen
- <u>Graphisch:</u>
  Schnittpunkt der Geraden durch
  P u. Q mit Kurve an der
  x-Achse spiegeln
- <u>Rechnerisch:</u>
  $x_3=s^2-x_1-x_2 \bmod p$
  $y_3=s(x_1-x_3)-y_1 \bmod p$
- Bei Punktaddition:
  $S=(y_2-y_1) / (x_2-x_1)$ ← Steigung der Geraden



# $G_f(F_p)$

- Ermitteln aller Punkte auf der Kurve
  → wiederholtes Anwenden der Operationen
- Zusätzliche neutrales Element:
  → Abstrakter Punkt bei y=+/-∞
- So dass P+(-P)=б
- Zusammen mit den Punkten der Kurve ergibt sich eine endliche, wiederkehrende Gruppe

→ Beispiel

# DL-Problem bei ECC

- Wahl des geheimen Schlüssels d
- Punkt P(x,y) mit d „multiplizieren“

$$dP = P+P+P+P+...+P = T$$

- Erzeugtes Element T kann als öffentlich Schlüssel genutzt werden
- Unmöglich von T auf d zu schließen ohne alle Elemente der Kurve auszurechnen

# Verschlüsselung

- Zu verschlüsselnde Nachricht M ist eine ganze Zahl
- Erzeugen eines Kurvenpunktes für M
- $x=Mk+i$
- K so wählen, dass sie p nicht teilt und nicht zu klein ist
- i variieren bis Kurvenpunkt gefunden ist ($i < k$)
- Wahrscheinlichkeit keinen Punkt zu finden ist $2^{-k}$
- Anschließend kann der Punkt verschlüsselt werden ($dP=T$)

# Beispiel ECDH

- Parameter E, P und p festgelegt und öffentlich
- E: $y^2=x^3+2x+2 \bmod 17$
- $P=(5,1)$

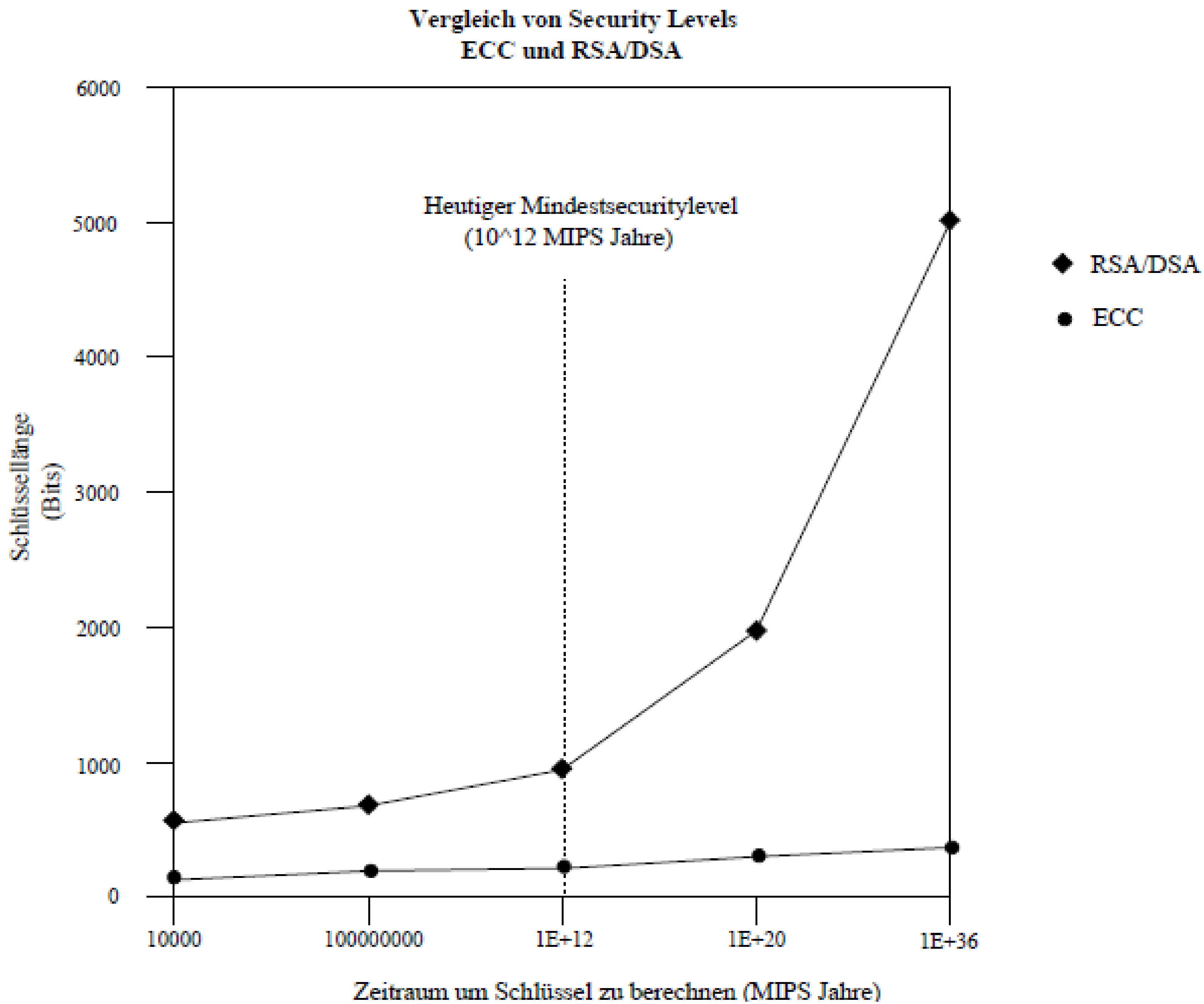
| **Alice** | | **Bob** |
|---|---|---|
| choose $a = k_{pr,A} = 3$ | | choose $b = k_{pr,B} = 10$ |
| $A = k_{pub,A} = 3P = (10,6)$ | | $B = k_{pub,B} = 10P = (7,11)$ |
| | $\xrightarrow{\quad A \quad}$ | |
| | $\xleftarrow{\quad B \quad}$ | |
| $T_{AB} = aB = 3(7,11) = (13,10)$ | | $T_{AB} = bA = 10(10,6) = (13,10)$ |

# Beispiel ECDH

- Gemeinsamer Schlüssel T kann als Sitzungsschlüssel genutzt werden
- I.d.R. Wird der Hash-Wert von x genutzt

# Sicherheit ECC (Vergleich RSA)

- Kompliziertere Rechnungen als RSA aber kürzere Schlüssel erforderlich
- Keine effektiven Algorithmen zur Lösung des DL-Problems in ECC
- Der beste Algorithmus hat exponentielle Laufzeit



Vergleich von Security Levels
ECC und RSA/DSA
6000
5000
4000
3000
2000
1000
0
Schlüssellänge
(Bits)
Heutiger Mindestsecuritylevel
(10^12 MIPS Jahre)
RSA/DSA
ECC
10000
100000000
1E+12
1E+20
1E+36
Zeitraum um Schlüssel zu berechnen (MIPS Jahre)

# Zusammenfassung

- Effizientes Verfahren
- Basiert auf DL-Problem: Punkte auf Kurve
- Asymmetrische Kryptographie
- Hält mit der Entwicklung der Computerindustrie mit
- Kombinierbar mit anderen Verfahren

# Quellenverzeichnis


- http://www.ecc-brainpool.org/
- http://www.frankdopatka.de/studium/koeln/mathe.pdf
- Cristof Paar, Jan Pelzl, 2010 „Understanding Cryptography“, Springer Verlag
- Anette Werner, 2002 „Elliptische Kurven in der Kryptographie“,  Springer Verlag
- Albrecht Beutelspacher, Jörg Schwenk, Klaus-Dieter Wolfenstetter, 2006 „Moderne Verfahren der Kryptographie“, vieweg Verlag