technische universität
dortmund

Vorlesung (WS 2014/15)

Sicherheit: Fragen und Lösungsansätze

Dr. Thomas P. Ruhroth

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Asymmetrische Verschlüsselung

[mit freundlicher Genehmigung basierend auf einem Foliensatz von Prof. Dr. Claudia Eckert (TU München)]

Literatur:

Claudia Eckert: IT-Sicherheit: Konzept - Verfahren - Protokolle, 7., überarb. und erw. Aufl., Oldenbourg, 2012.

E-Book: http://www.ub.tu-dortmund.de/katalog/titel/1362263

Agenda

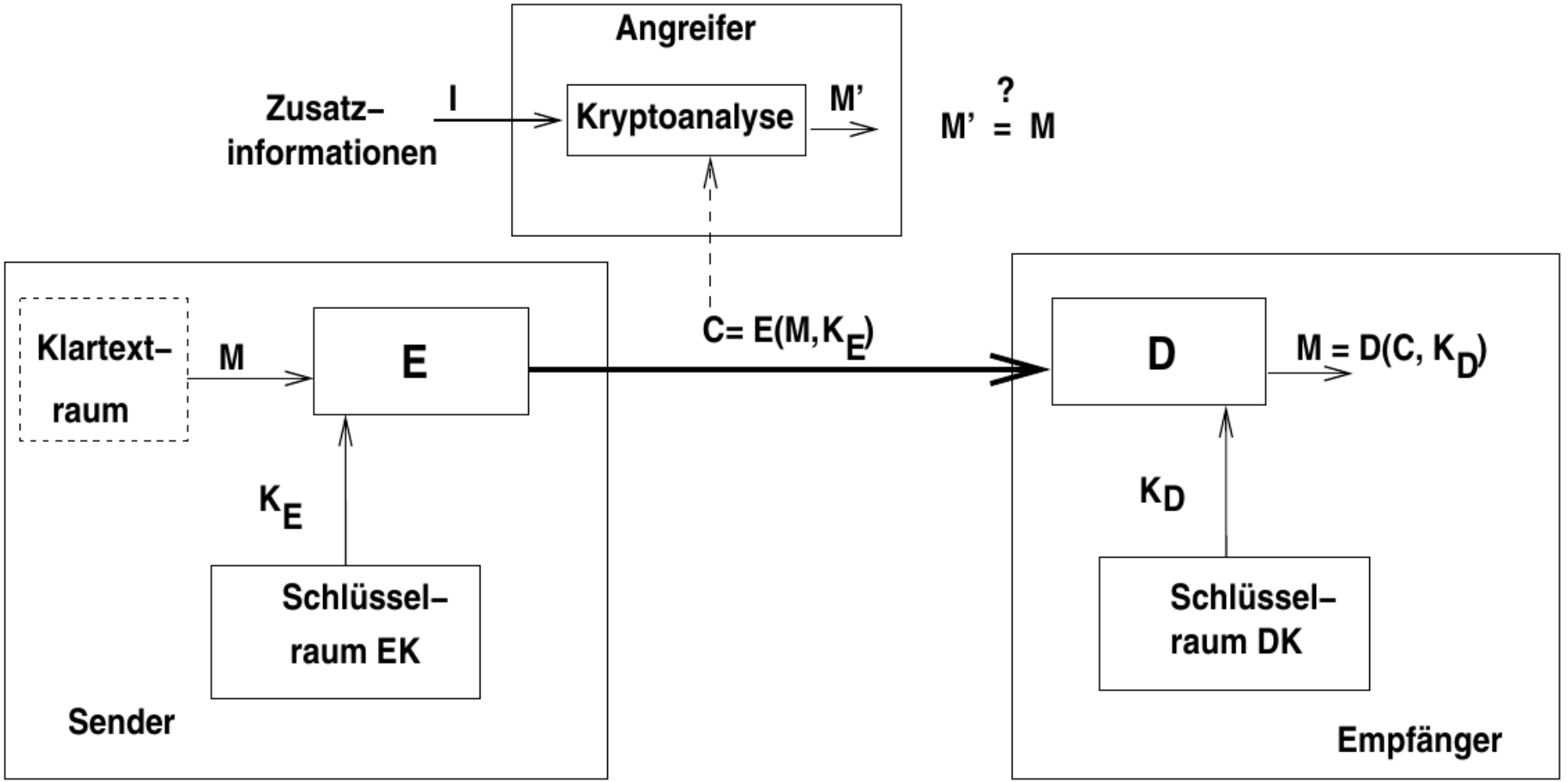
- Grundlagen
- Asymmetrische Verfahren
  - RSA
  - ECC
- Gedanken um weitere Verfahren und Eigenschaften

- Grundlagen und Algorithmen der asymmetrischen Verfahren kennen und anwenden können.

Wo waren wir

- Letzte Woche
  - Einführung Kryptographische Systeme
  - Symmetrische Verschlüsselungen
    - One-Time-Pad
    - DES
    - AES

Kryptographische Verfahren



Formal:
Kryptographische Verfahren

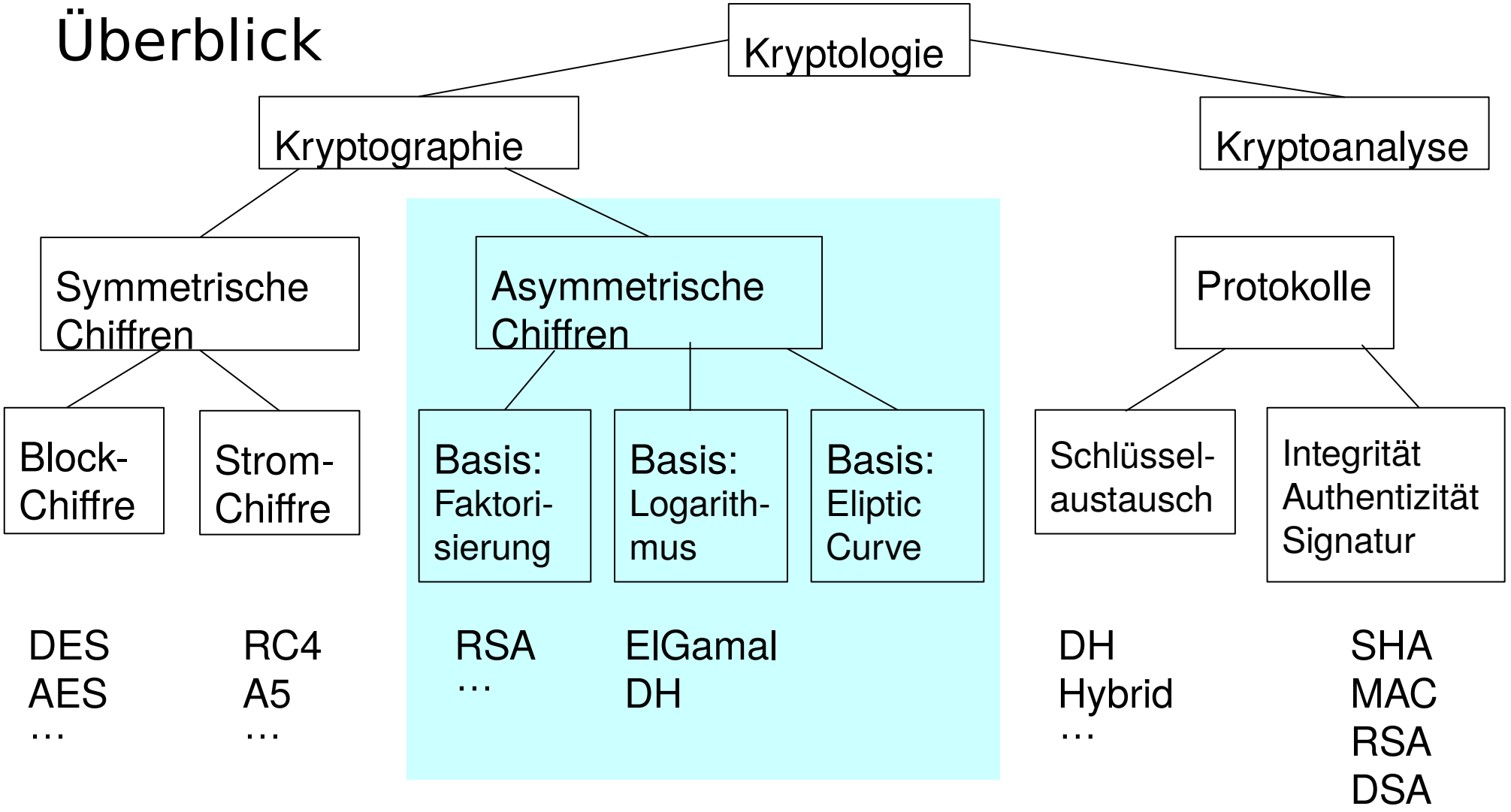
Definition: Ein kryptographisches Verfahren ist gegeben durch ein Tupel: (M,C,EK,DK,E,D)

1. M ist die Menge von Klartextnachrichten (Plaintext) *M*, über dem Alphabet A_1
2. C ist die Menge von Kryptonachrichten (Chiffretext) *C*, über dem Alphabet A_2
3. der Menge von Verschlüsselungs-Schlüsseln EK,
4. der Menge von Entschlüsselungs-Schlüsseln DK, und der Abbildung:

   $f : EK \rightarrow DK$, mit: $d = f(e)$, $e \in EK$, $d \in DK$
5. Der Familie E der Verschlüsselungsverfahren $E = \{E_k \mid E_k : M \rightarrow C\}$
6. Der Familie D der Entschlüsselungsverfahren $D = \{D_k \mid D_k : C \rightarrow M\}$, mit

   $\forall M \in M$ gilt: $D_d(E_e(M)) = M$ mit $e \in EK$, $d \in DK$, $f(e) = d$

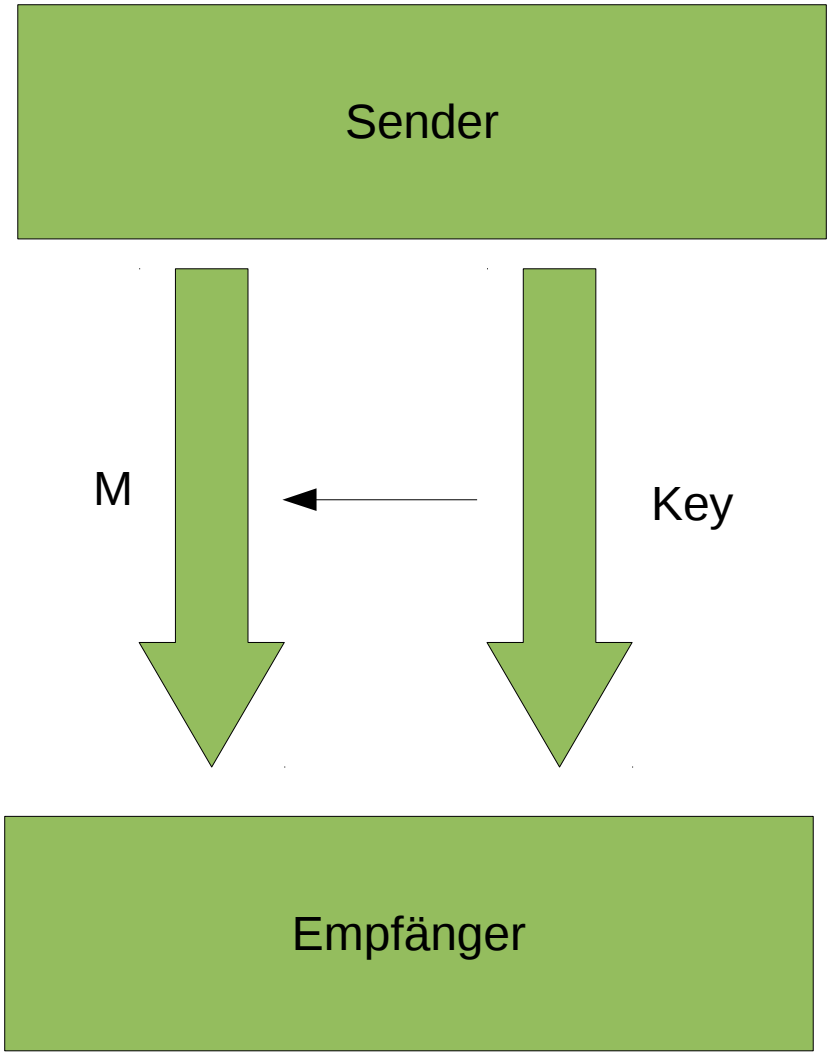
Überblick


Kryptologie
Kryptographie
Kryptoanalyse
Symmetrische Chiffren
Asymmetrische Chiffren
Protokolle
Block-Chiffre
Strom-Chiffre
Basis: Faktori-sierung
Basis: Logarith-mus
Basis: Eliptic Curve
Schlüssel-austausch
Integrität Authentizität Signatur
DES
AES
…
RC4
A5
…
RSA
…
ElGamal
DH
DH
Hybrid
…
SHA
MAC
RSA
DSA

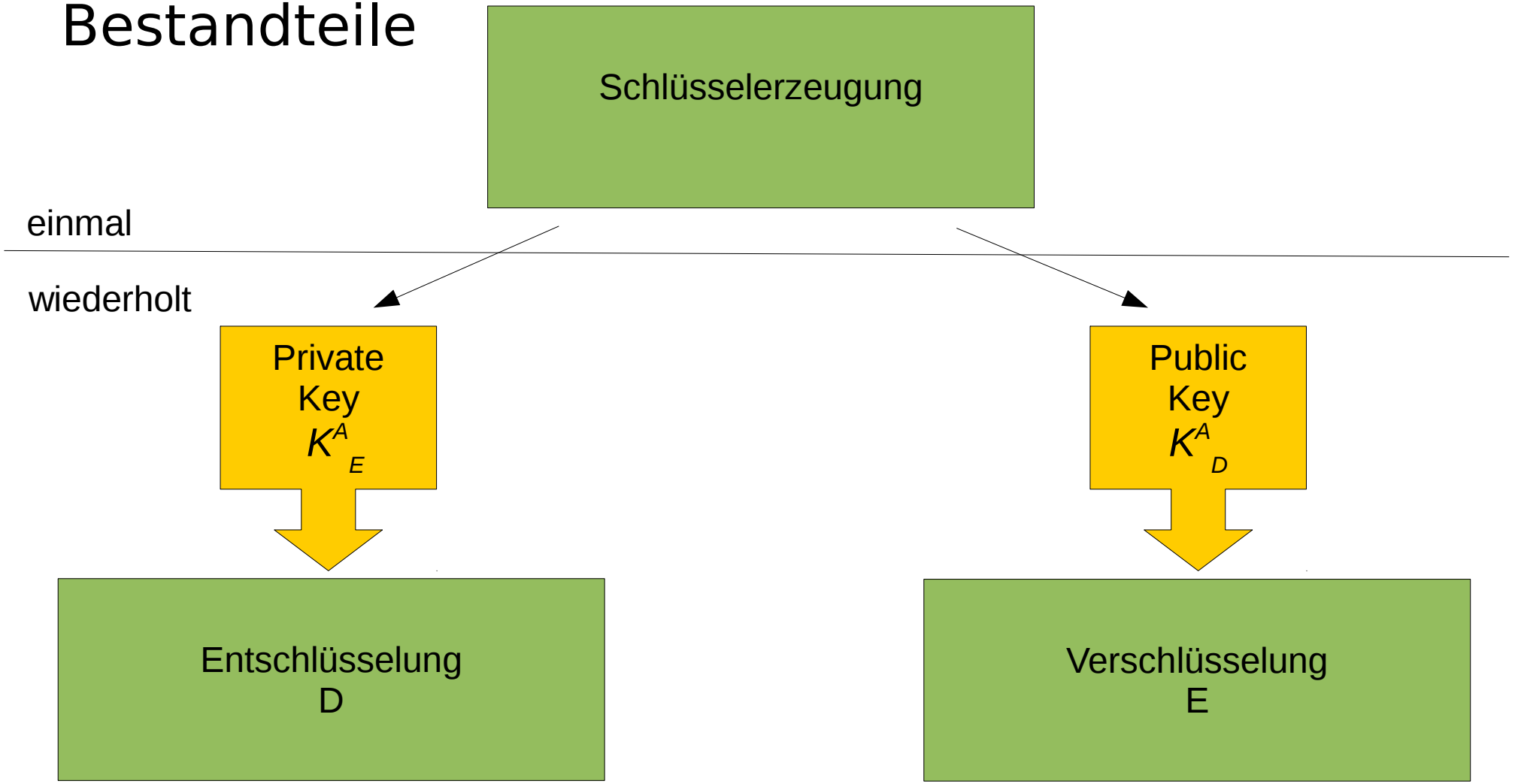
Asymmetrische Verfahren

Das Schlüsselproblem

- Schlüssel müssen sicher Übermittelt werden
  - Geheim
  - Authentizität
- Keine Wiederverwendung



Bestandteile



Public-Key Kryptographie: zentrale Idee:

- *Ein Schlüsselpaar (K^A_E , K^A_D) pro Kommunikationspartner A*
- *privater Schlüssel K^A_D (nur A bekannt) und*
- *öffentlicher Schlüssel K^A_E*
- K^A_E wird von Kommunikationspartnern B verwendet, um Nachrichten an A damit zu verschlüsseln.
- A verwendet K^A_D zum Entschlüsseln der mit K^A_E verschlüsselten Nachrichten

Asymmetrische Verfahren, Public-Key-Verfahren

Basis:

- Mathematische Funktionen zur Beschreibung der Verfahren,
- basieren auf zahlentheoretisch schwierig zu lösenden Problemen,
- ist notwendig, da man Teile des Schlüssel (public Key) preis gibt
- Bei asymmetrischen Verfahren:
  - keine kompakte mathematische Beschreibung der Chiffre,
    - für Teile notwendig (S-Box), aber nicht für gesamte Chiffre

Anforderungen an asymmetrische Verfahren

- Die Schlüsselpaare (K_E, K_D) müssen folgende Eigenschaft erfüllen:

  $\forall\, M \in A_1^* : D\,(E\,(M, K_E\,), K_D\,) = M$,

  K_E *sei* der öffentliche, K_D *der* geheime Schlüssel.

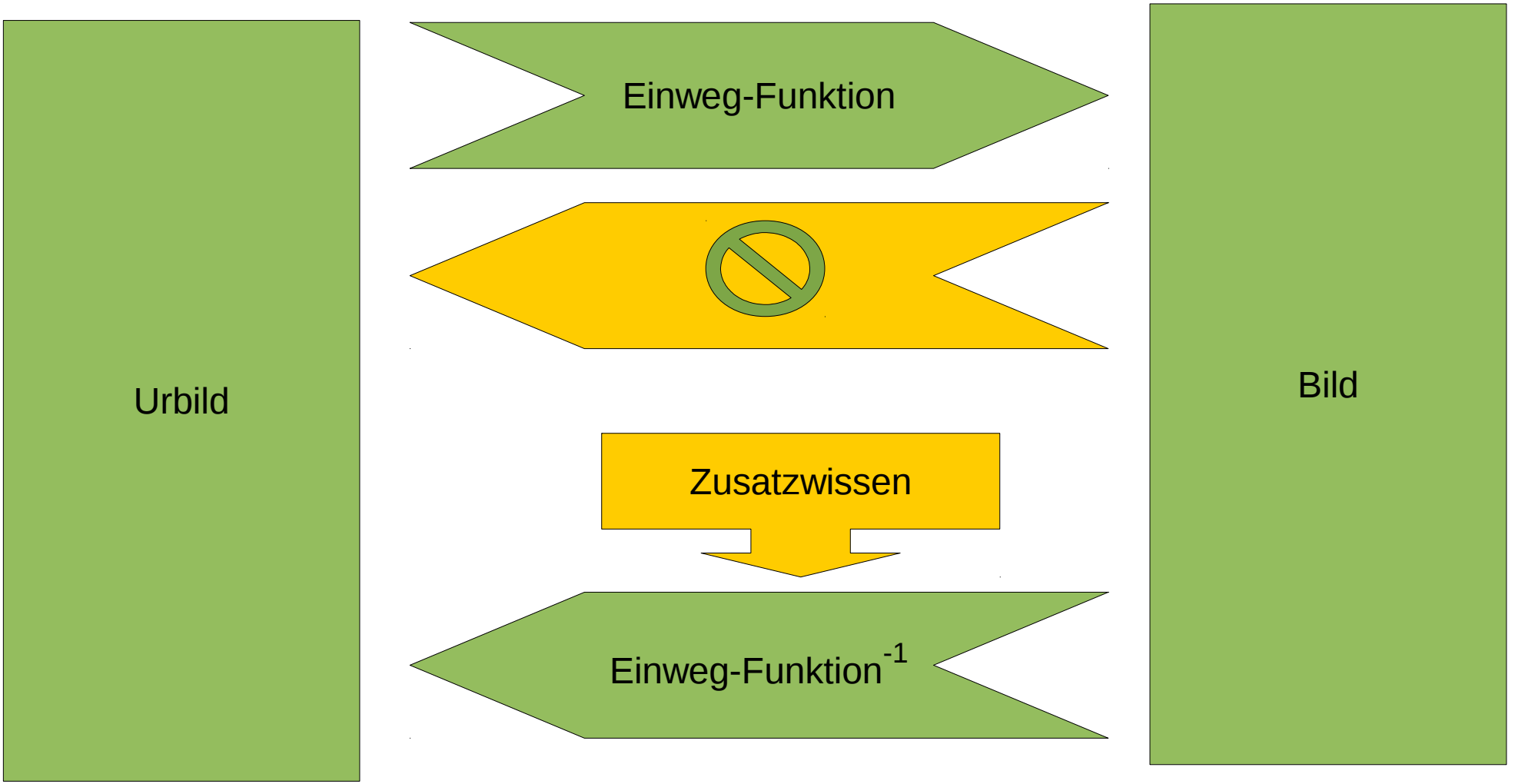
  Schlüsselpaare müssen effizient erzeugbar sein.
- Ver- und Entschlüsselungen (*E* und *D*) sind effizient durchführbar
- K_D ist aus K_E nicht mit vertretbarem Aufwand berechenbar

Optionale Eigenschaft:

$\forall\, M \in A_1^* : E\,(D\,(M, K_D\,), K_E\,) = D\,(E\,(M, K_E\,), K_D\,) = M$

Wozu ist diese Eigenschaft nützlich?

Einweg-Funktionen mit Falltür



Einweg-Funktionen

Basis von Publik-Key Verfahren:

Einweg-Funktion (engl. one-way) $f : X \rightarrow Y$

Eigenschaften von Einweg-Funktionen:

(1) $\forall\, x \in X$ gilt: $f(x)$ ist effizient berechenbar (in P) und

(2) für fast alle $y \in Y$ gilt, dass es nicht effizient (min. NP) möglich ist, das Urbild zu

$y = f(x)$ zu berechnen, also $f^{-1}(y) = x$ ist schwer

- Existenz von Einwegfunktionen bis heute aber unbewiesen
- Problem ist verwandt mit **P ≠ NP**

Einweg-Probleme

3 Familien von PK-Verfahren basierend auf zugrundeliegenden, Mathematischen Problemen:

1. PK-Verfahren basierend auf dem Faktorisierungsproblem
2. PK-Verfahren basierend auf dem diskreten Logarithmusproblem
3. PK-Verfahren basierend auf elliptischen Kurven, Verallgemeinerung von (2)

Einweg-Probleme

Faktorisierung:

gegeben $n = pq$, *wobei* p,q Primzahlen

Einwegfunktion: f(p,q) = pq = n ist einfach,

Primfaktorzerlegungsproblem: $f^{-1}(n)$= pq ist schwer

Diskreter Logarithmus:

gegeben p Primzahl, $g \leq p$ und y

Einwegfunktion: $f(x) = g^x \bmod p = y$ einfach

Diskreter Logarithmus: $f^{-1}(y) = \log_g y \bmod p$ ist schwer

Diskreter Logarithmus auf elliptischen Kurven

Gegeben EC, P Punkt innerhalb EC

Einwegfunktion: f(P,x) = xP

Diskreter Logarithmus auf EC: $f^{-1}(P,Q) = a$ mit $Q = aP$ ist schwer

Faktorisierung mit Falltüre

Einweg-Funktionen mit Falltür (engl. trapdoor one-way):

- mit Zusatzinformation sind Urbilder effizient berechenbar
- **Bsp**.: gegeben $n = pq$, und Funktion f mit $f(x) = x^2 \bmod n$,
  - Invertierung von f schwierig ohne Kenntnis von p,q
  - Kenntnis von p,q ist ‚Falltür', mit p,q ist Invertierung effizient berechenbar

Beispiel für asymmetrische Verfahren mit Trapdoor-One-Way:

- RSA („Quasi-Standard"), Idee: Falltür zur Entschlüsselung ist der private Schlüssel

RSA-Verfahren (1978)

Rivest, Shamir, Adleman

Einweg-Funktion: Faktorisierung

RSA-Verfahren: Grundlagen

- Seien p,q Primzahlen, n = pq, dann gilt für jede natürliche Zahl m ≤ n und für jede natürliche Zahl k:

$$m^{k(p-1)(q-1)+1} \bmod n = m$$

- Euler'sche Zahl:

  $\varphi(m) = |\{ a | ggT(a,m) = 1 \wedge a < m\}|$ d.h. a ist relativ prim zu m

  Für Primzahlen *p* gilt: $\varphi(p) = p - 1$

- Satz von Fermat: sei p Prim, a Integer:

  $a^p = a \bmod p$ oder auch: $a^{p-1} = 1 \bmod p$

- Euler's Theorem: Falls *ggT(M,n)* = 1 dann $M^{\varphi(n)} = 1 \bmod n$

RSA

- Verschlüsselung und Entschlüsselung sind Funktionen auf dem Ganzzahl-Ring Z_n, wobei n=pq das RSA-Modul ist
- RSA verschlüsselt Klartext x, der ein Element aus Z_n sein muss, also ein Element aus {0, 1, … , n-1}
- Der Binärwert von x muss kleiner n sein

RSA-Verschlüsselung:

- Gegeben sei Klartext x und (n, e)= K_E der öffentlicher Schlüssel;
- Verschlüsselung $E(x, K_E) = x^e \bmod n = y$, *mit* $x,y \in Z_n$

RSA-Entschlüsselung:

- Gegeben sei der Kryptotext y und der geheime Schlüssel $d = K_D$
- Entschlüsselung: $x = D(y, K_D) = y^d \bmod n$

RSA Schlüsselpaare: Berechnung

1. wähle 2 große Primzahlen p , q
2. Berechne RSA-Modul $n = pq$
3. Berechne $\varphi(n) = (p-1)(q-1)$
4. Wähle öffentlichen Exponent e $\in \{1, 2, \ldots, \varphi(n)-1\}$ so, dass $ggT(\varphi(n), e) = 1$
5. Berechne privaten Exponenten d, so dass

   $ed \bmod \varphi(n) = 1 \bmod \varphi(n)$, da e so gewählt ist, dass $ggT(\varphi(n), e) = 1$, ist gewährleistet, dass es immer ein Inverses zu e modulo φ(n) gibt, d.h. dass es einen zugehörigen privaten Schlüssel d zu e gibt

Einsatzbereiche: Verschlüsseln, Signieren, Schlüsselaustausch

Beispiel RSA

Sei $p = 47$ und $q = 59$; $n = p * q = 2773$, $\varphi(n) = (p-1)(q-1) = 2668$

1. Wähle e = 17 und berechne d = 157 (da $ed = 1 \bmod \varphi(n)$)
2. **Codierung**: 4 Bit/Buchstabe; Blockgröße 2^8,

   zwei Buchstaben pro Block: 00 = , 01 = *a*, 02 = *b*, …
3. Klartextnachricht *M* = *„its all greek to me“* codiert als : 09 20 19 00 …
4. Verschlüsseln von 0920: $E(920, 17) = 920^{17} \bmod 2773 = 948$

   Kryptotext: 0948 2342 …
5. Entschlüsselung: $D(948, 157) = 948^{157} \bmod 2773$

Algorithmen für RSA

Benötigte Algorithmen für RSA u.a.:

- Primazahltest: z.B. Rabin Miller
- Berechnen der multiplikativen Inversen:

  z.B. Erweiterter Euklid (Euclid: ggT. Berechnung)
- Ver-/Entschlüsselung: schnelles Potenzieren großer Zahlen:
- z.B. Repeated Squaring and Multiplication Algorithmus

RSA Sicherheit: Angriffsklassen:

Protokoll (Implementierung), mathematische, Seitenkanal-Angriffe

Protokollangriffe: Nutzen der Multiplikativität: durch Padding lösbar

(1) Kenntnis von Klartexten M_1,M_2 und deren Signaturen:

$sig_1 = M_1^d \bmod n$ und $sig_2 = M_2^d \bmod n$

(2) Erzeugen von sig_3 ohne Kenntnis von d:

$(M_1M_2)^d \bmod n = sig_1\, sig_2 \bmod n = sig_3$ sig_3 ist gültige Signatur

Mathematische Angriffe

- Faktorisierung des Moduls n:
  - Angreifer kennt: Modul n, Public Key e und Kryptotext y
  - Angreifer kennt aber nicht $\varphi(n)$, *also die Primfaktoren von n*
  - Angreifer versucht n zu Faktorisieren,
  - wenn dies gelingt, dann kann er berechnen:
    - $\varphi(n) = (p-1)(q-1)$
    - $d^{-1} = e \bmod \varphi(n)$
    - $x = y^d \bmod n$ *Bingo*

  n mit 330 Bit in 1991 faktorisiert, n mit 426 Bit in 1994,

  n mit 512 Bit in 1999, n mit 664 Bit in 2005
- **Fazit:** Wettlauf mit Rechengeschwindigkeit: Modul muss groß sein: 2048 Bit bis 4096 Bit für starke Sicherheit

Elliptic Curve Cryptography

Einwegfunktion:

Diskreter Logarithmus auf elliptischen Kurven

Elliptische-Kurven-Kryptographie (ECC)

Motivation:

Asym. Verfahren wie RSA erfordern Exponantionsoperationen in Integer-Ringen und Feldern mit Parametern mit > 1000 bits.

- Hoher Berechnungsaufwand bei 32- oder 64-bit Arithmetik
- Große Parameter: Speicheraufwändig für Embedded Systems

Gesucht:

- Kleinere Feldgrößen mit äquivalentem Sicherheitsniveau

Lösung

- ECC: Gruppe von Punkten (Gruppe im Sinn der algebraischen Struktur) (anstatt Integer) mit Koeffizienten von 160-256 bits,
- Signifikante Reduktion des Berechnungsaufwandes

Elliptische Kurven

ECC basiert auf dem mathematischen Konzept der elliptischen Kurven: Punktmengen, die eine Polynomial-Gleichung erfüllen

Definition: Gegeben sei der Ring $\mathbf{Z}_p$, p > 3. Eine Elliptische Kurve ist die Menge der Punkte (x,y) aus $\mathbf{Z}_p$, für die gilt: $y^2 = x^3 + ax + b \bmod p$

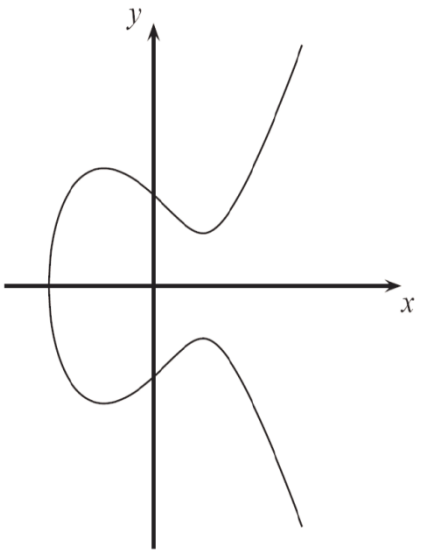
und einem imaginären Punkt θ, so dass gilt $4a^3+27b^2 \neq 0 \bmod p$.

Parameter a,b definieren die Form der jeweiligen Kurve

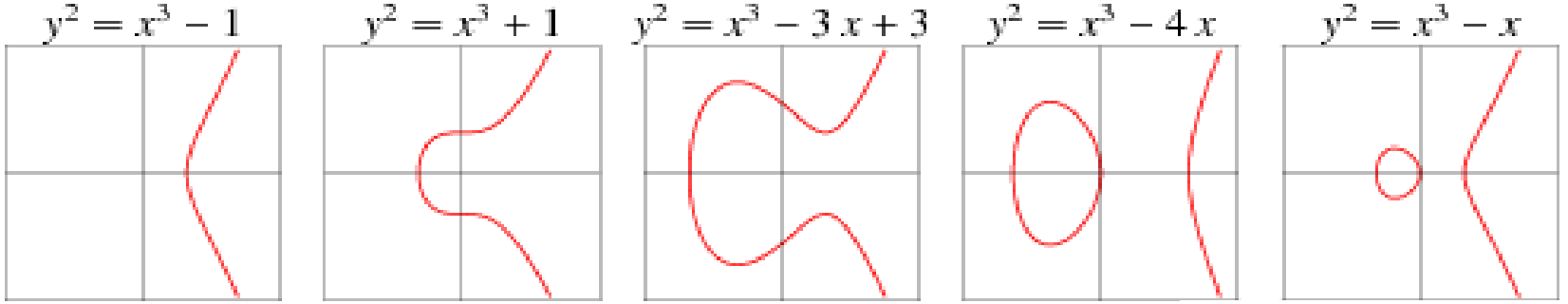
$\mathbf{Z}_p$ ist der Integer-Ring {0, .., p-1} mod p

- Beispiel in $\mathbf{R}^2$:

  $y^2 = x^3 - 3x + 3$



Beispiele in R²:

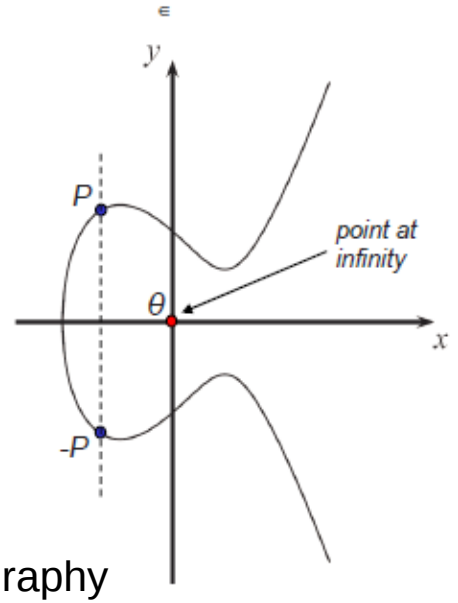


ECC: IdR Kurven über GF(p), wobei p eine Primzahl ist

Problem:

Aus EC muss Gruppe von Punkten werden:

- Erforderlich neutrales Element θ, so dass
  P + θ = θ + P
- EC sind symmetrisch entlang der X-Achse



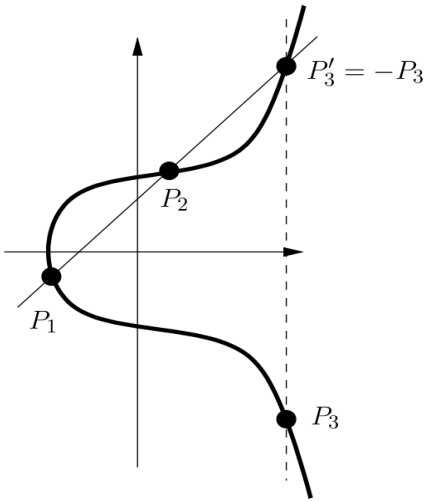
Quelle: C. Paar, Understanding Cryptography

Operationen

Addition in einer abel'schen Gruppe für Punkte P, Q, R:

- $P + Q = Q + P$ (kommutativ)
- $P + (Q + R) = (P + Q) + R$ (assoziativ)
- $P + 0 = 0 + P = P$ (neutrales Element: 0)
- $P + (-P) = 0$ (inverses Element: $-P$)
- $n \cdot P = \underbrace{P + P + \ldots + P}_{\text{n-mal}}$ (Skalarmultiplikation, effizient berechenbar)

Geometrische Interpretation der Punkt-Addition $P_1 + P_2 = P_3$



$P_1 + P_2 = P_3.$

Gruppenstruktur

Gruppenstruktur einer elliptischen Kurve über Z_p

Für eine elliptische Kurve $E : y^2 = x^3 - ax - b$ über Z_p gilt:

- Inverses Element: Für $P = (x, y)$ ist $-P = (x, -y)$
- Addition von Punkten:

  Für $P_i = (x_i, y_i)$ gilt: $P_1 + P_2 = (x_3, y_3)$ mit $\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & falls\ P_1 \neq P_2 \\ \frac{3x_1^2 + a}{2y_1}, & falls\ P_1 = P_2 \end{cases}$

  und $x_3 = \lambda^2 - x_1 - x_2$

  $y_3 = \lambda(x_1 - x_3) - y_1$
- Definition erfüllt Gruppenaxiome

Einweg-Eigenschaft:

Aus P, nP ist es nicht effizient möglich, n zu bestimmen.

ECC Diskreter Logarithmus Problem (ECDLP)

Gegeben sei ein primitives Element P und ein Element T auf der elliptischen Kurve E.

Das ECDL-Problem besteht darin, einen Integer-Wert d zu finden,

wobei $1 \leq d \leq |E|$ so dass

$$T = d \cdot P = \underbrace{P + P + \ldots + P}_{\text{d-mal}}$$

EC-Kryptosysteme basieren darauf, dass d groß ist und geheim gehalten wird.

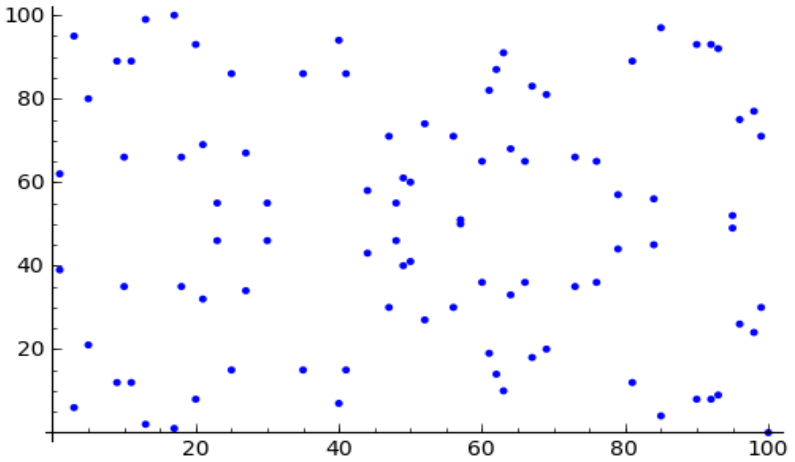
Falls d bekannt wäre, können bekannte, effiziente Verfahren, wie Square and Multiply auf ECs angewandt werden, um dP zu berechnen.

Schlüsselerzeugung

Wähle $\mathbf{Z}_p$, p Primzahl (typisch: $2^{160} \leq p \leq 2^{320}$)

- Wähle E über dem Körper $\mathbf{Z}_p$
- Wähle Basispunkt $G \in E(\mathbf{Z}_p)$ der Ordnung n ($\approx p$)
- Wähle zufällig privaten Schlüssel $d \in \{1, \ldots, n-1\}$
- Berechne öffentlichen Schlüssel $Q = dG$

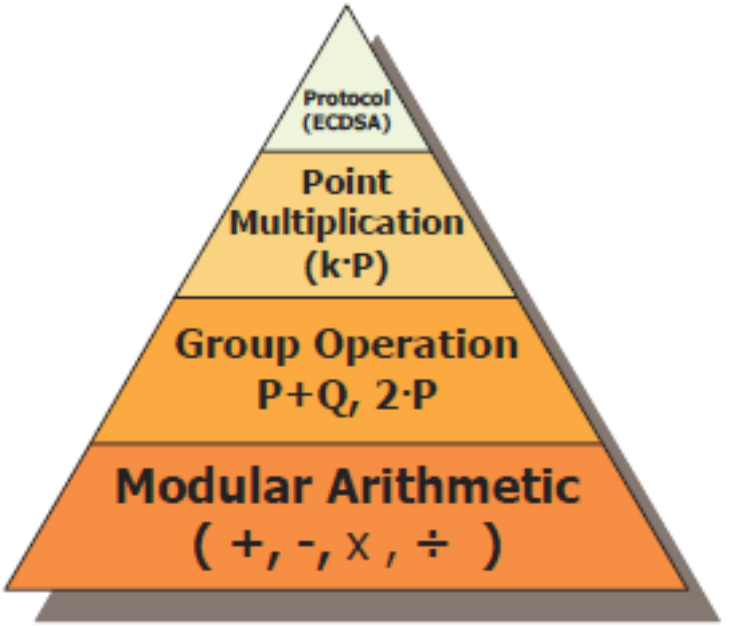
Bsp.: Elliptische Kurve $y^2 = x^3 + 2x + 3$ über $\mathbf{Z}_{101}$



Hard- und Software-Implementierung von ECC

EC Implementierungen werden idR. als Schichtenmodell betrachtet:

- Modulo-Arithmetik: am Aufwändigsten
- Operationen auf Gruppen:
  - Point Doubling & Point Addition
  - Point multiplication kann implementiert werden über Double-and-Add Methode
- Protokolle auf höheren Schichten, z.B. ECDSA, ECDH



Quelle: C. Paar, Understanding Cryptography

ECC Sicherheit

- ECC-Parameter können signifikant kleiner sein als bei RSA,
  - weil bekannte Angriffe auf ECs deutlich schwächer sind als verfügbare Faktorisierungsangriffe auf RSA.
  - Bekannteste Angriffsmethoden erfordern im Durchschnitt $\sqrt{p}$ Schritte
- Konsequenz:
  - Eine EC mit Primzahl p mit 160Bits und ca. 2^{160} Punkten, erfordert im Schnitt 2^{80} Berechnungsschritte für einen Angriff
  - Eine EC mit Primzahl p mit 256 Bits und ca. 2^{256} Punkten, erfordert im Schnitt 2^{128} *Berechnungsschritte*

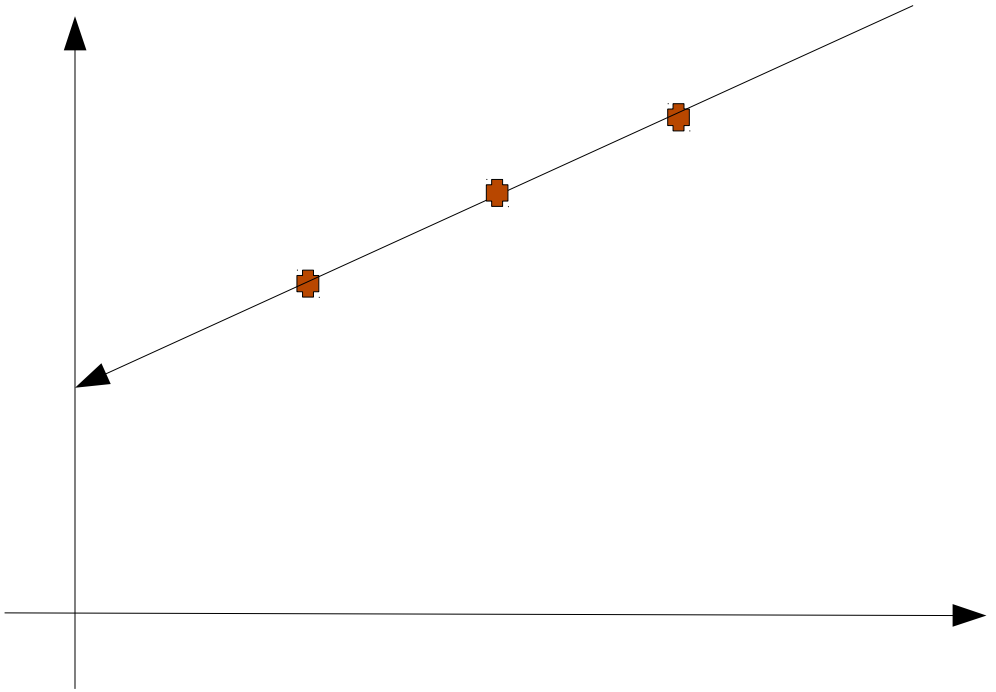
ECC kann zum Verschlüsseln, Signieren und zum Schlüsselaustausch eingesetzt werden.

Wenn dies ein Krytopgraphiekurs wäre
würden wir uns noch ansehen....

- Visuelle Kryptographie
- Verteile-Schlüssel-Verfahren
- Homomorphe Verschlüsselung

Verteiltes Schlüssel Wissen

- Absicherung von Aktionen die nur durch mehrere zusammen durchgeführt werden dürfen
  - Tresore
  - Transaktionen
  - Atombomben
- Idee für je zwei Personen:



Nächste Woche

Hashfunktion, Signatur
und
Schlüsselaustausch