# Informatik-Propädeutikum

Dozentin: Dr. Claudia Ermel

Betreuer: Sepp Hartung, André Nichterlein, Clemens Hoffmann

Sekretariat: Christlinde Thielcke (TEL 509b)

TU Berlin
Institut für Softwaretechnik und Theoretische Informatik
Prof. Niedermeier
Fachgruppe Algorithmik und Komplexitätstheorie
http://www.akt.tu-berlin.de

Wintersemester 2013/2014

# Gliederung

# Wiederholung: Funktion von RSA-Kryptosystemen

Verschlüsselung $E$ (*Encryption*) von Message $m$ mit dem öffentlichen Schlüssel $(N = p * q, e)$ des Empfängers:

$$E(m) = m^e \bmod N$$

Entschlüsselung $D$ (*Decryption*) von $E(m)$ mit dem aus $(N, e)$ berechneten privaten Schlüssel $d$ des Empfängers:

$$\begin{aligned} D(E(m)) &= (E(m))^d \bmod N \\ &= m \end{aligned}$$

*Beispiel:*

Bobs öffentlicher Schlüssel ist $(187, 7)$, also $N = 187$ und $e = 7$. Alice möchte Nachricht $m = 76$ an Bob schicken. Sie verschlüsselt $76$ mit Bobs öffentlichem Schlüssel:

$$E(76) = 76^7 \bmod 187 = 32$$

Bob berechnet den privaten Schlüssel $d = 23$ (nur Bob kann das!) und entschlüsselt die verschlüsselte Nachricht $32$ mit

$$D(32) = 32^{23} \bmod 187 = 76$$

# Digitale Signatur I

Zentrale Frage: Ist eine erhaltene Datei wirklich von einem bestimmten, angegebenen Absender? Wie beweise ich, dass eine Nachricht wirklich von mir kommt?

Antwort im analogen Schriftverkehr: Unterschriften!

$\rightsquigarrow$ Wie können digitale Unterschriften/Signaturen realisiert werden?

Anforderung an digitale Signatur:

1. Besitzer des Dokuments muss spezifische Unterschrift erzeugen können.
2. Unterschrift darf nicht vom Dokument getrennt werden können.
3. Außenstehende müssen Unterschrift verifizieren können.

# Digitale Signatur II

Realisierung mittels asymmetrischer Verfahren (z.B. RSA):

1. Mit dem geheimen Schlüssel wird das Dokument signiert: dazu wird die Entschlüsselungs-funktion „missbraucht“: $sig = D(m) = m^d \bmod N$
2. Mit dem öffentlichen Schlüssel wird das Originaldokument wiederhergestellt: dazu wird die Verschlüsselungsfunktion „miss-braucht“: $E(sig) = sig^e \bmod N$
3. Stimmt das resultierende Dokument mit dem im Klartext versandten Dokument überein (ist $E(sig) = m$), so ist die Signatur verifiziert.

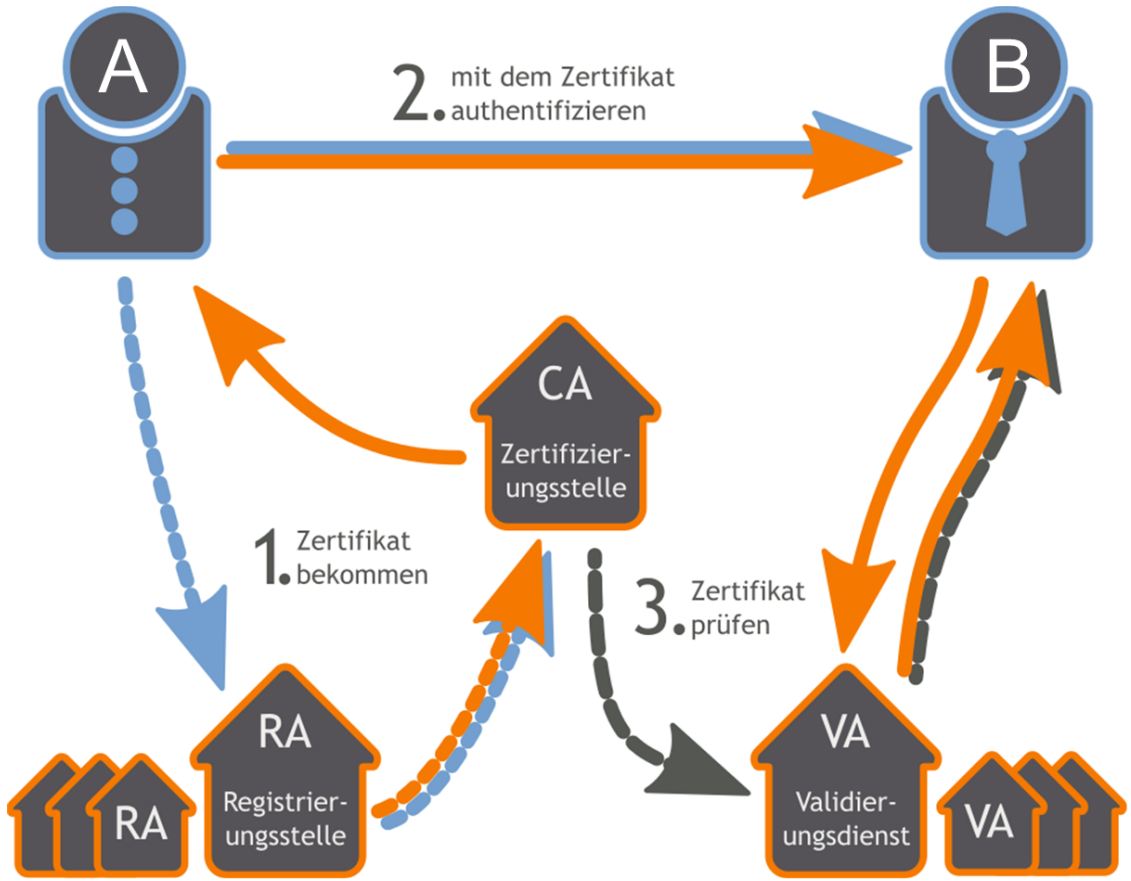
1. Bob will Alice ein signiertes Dokument mit Nachricht $m = 76$ schicken. Er berechnet $sig = D(76) = 76^{23} \bmod 187 = 32$. Er verschickt $(76, 32)$.
2. Alice stellt mit Bobs öffentlichem Schlüssel das Dokument aus der Signatur wieder her: $E(32) = 32^7 \bmod 187 = 76$.
3. Das Ergebnis stimmt mit dem im Klartext versandten Dokument überein, also ist Bobs Signatur verifiziert.

# Digitale Signatur III

Beachtet werden muss:

1. Aus Klartext *und* Geheimtext darf der private Schlüssel nur „schwer“ rekonstruierbar sein $\rightsquigarrow$ sichergestellt durch RSA-Verfahren.
2. Problem: Der öffentliche Schlüssel muss eindeutig dem Absender zugeordnet sein, damit man nicht einen falschen öffentlichen Schlüssel untergeschoben bekommt $\rightsquigarrow$ Öffentlicher Schlüssel wird mittels *Zertifizierungsstelle* überprüft.
3. Problem: Daten werden doppelt verschickt (als Klar- und Geheimtext) $\rightsquigarrow$ Code einsparen durch Verwendung von *Hashfunktionen*, um statt des ganzen Dokuments nur einen kurzen (eindeutigen) Hashwert zu ent- und verschlüsseln.

# Zertifizierung: Prinzip von Public-Key-Infrastrukturen

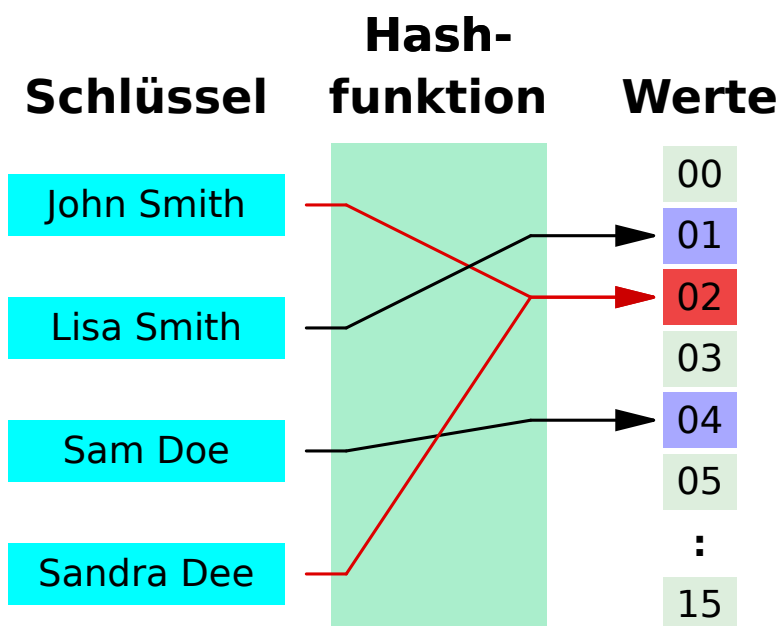



1. Teilnehmer A authentifiziert sich gegenüber Registrierungsstelle RA. RA bestätigt den öffentlichen Schlüssel von A gegenüber der Zertifizierungsstelle (CA). CA stellt A ein Zertifikat aus.
2. A authentifiziert sich gegenüber Teilnehmer B mit dem Zertifikat.
3. B lässt das Zertifikat von einem Validierungsdienst (VA) überprüfen.

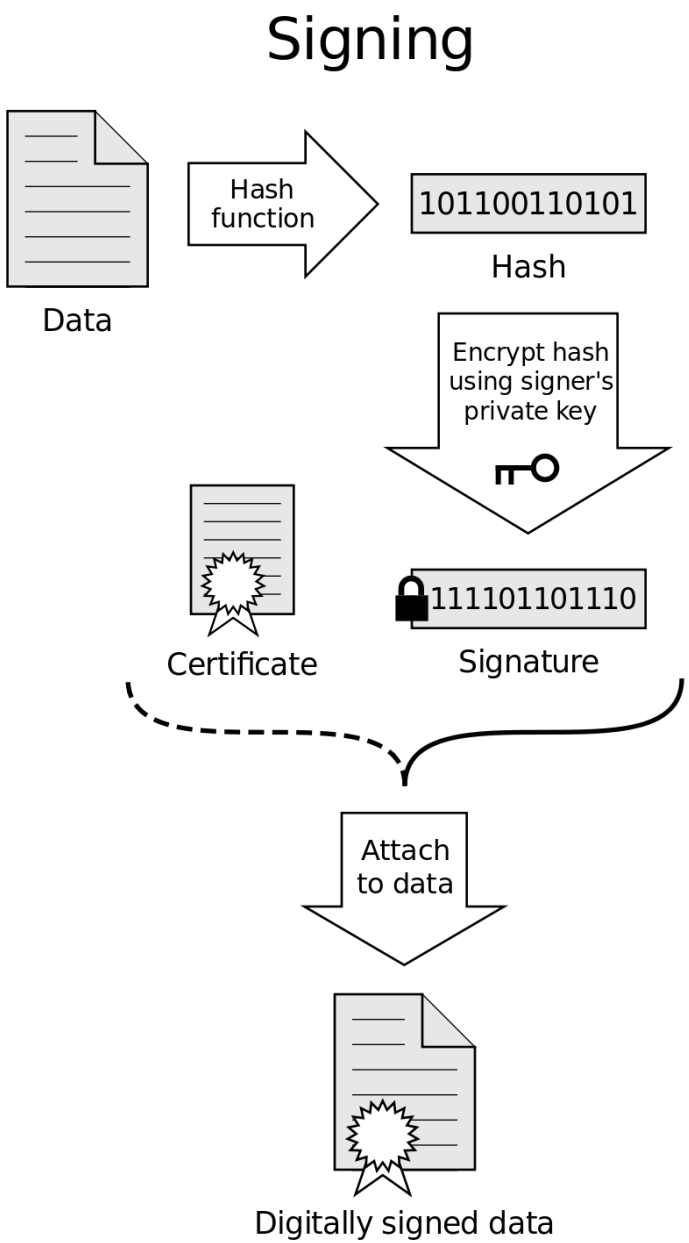
# Hashfunktionen

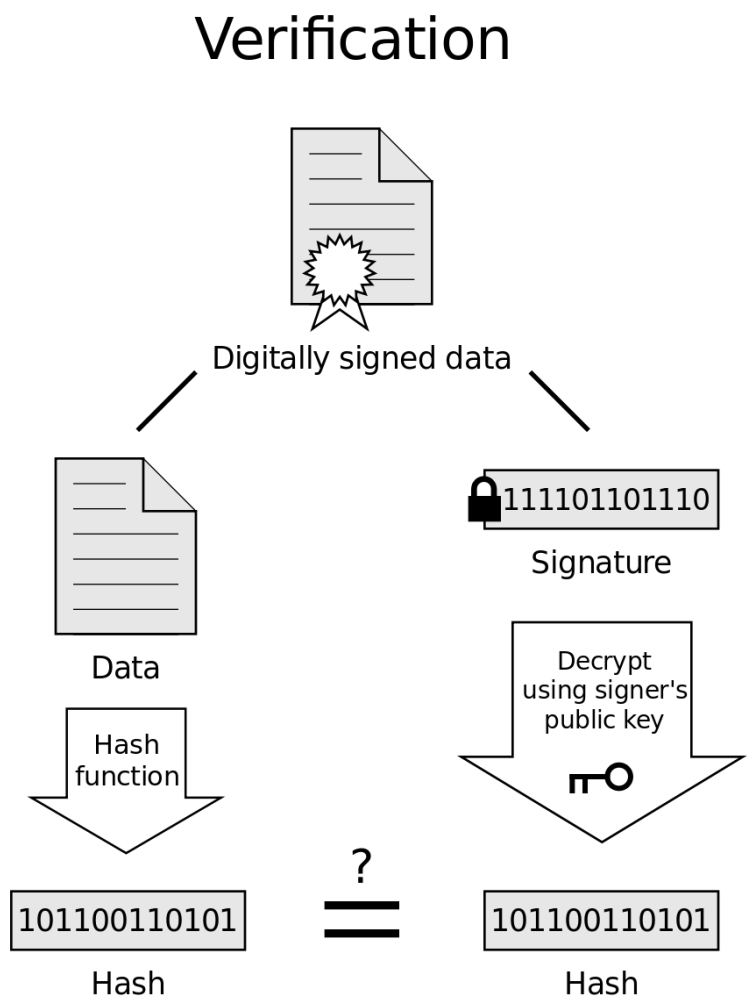
Funktion die von einer großen (potentiell unendlichen) Schlüsselmenge (z. B. alle Binärzeichenketten) in eine kleinere Wertemenge (z. B. Binärzeichenkette mit max. 512 Bit) abbildet.

**Folgerung:** Es gibt Schlüssel mit selbem Hashfunktionswert (nicht injektiv).
**Anforderung:** Für gegeben Hashfunktionswert $f(x)$ soll es schwer sein ein $x'$ zu finden mit $f(x') = f(x)$. Beachte dabei dass mgl. $x \neq x'$.



# Digitale Signatur IV




If the hashes are equal, the signature is valid.

# Anwendungen von RSA und Digitaler Signatur

- **PGP: Pretty Good Privacy** (Phil Zimmermann, 1991)
  - z.B. bei Email (Thunderbird): Enigmail Add-On (OpenPGP Standard)
  - Hybride Verschlüsselung: Symmetrische Verschl. der eigentlichen Nachricht, asymmetrische Verschl. (RSA) des verwendeten Schlüssels

| asymmetrisch für Empfänger 1 verschlüsselter Schlüssel der Nachricht |
|---|
| ⋮ |
| asymmetrisch für Empfänger *n* verschlüsselter Schlüssel der Nachricht |
| symmetrisch verschlüsselte Nachricht |

  - basiert auf *Web of Trust* (keine zentrale Zertifizierung)
  - Auch Signieren von Emails möglich (Hashfunktion SHA-256)
- **S/MIME** (Secure Multipurpose Internet Mail Extensions)
  - ebenfalls hybride Verschlüsselung von Emails
  - erfordert X.509-basierte Zertifikate
- **HTTPS** ( SSL/TLS zur Verschl. zwischen Server und Client)
  - rechenaufwändiges Verschlüsseln auf Serverseite, daher nur vereinzelt
  - Grundlage: Asymmetrische Verschlüsselung mit Zertifikat

# Aufteilen und Verteilen von Geheimnissen I

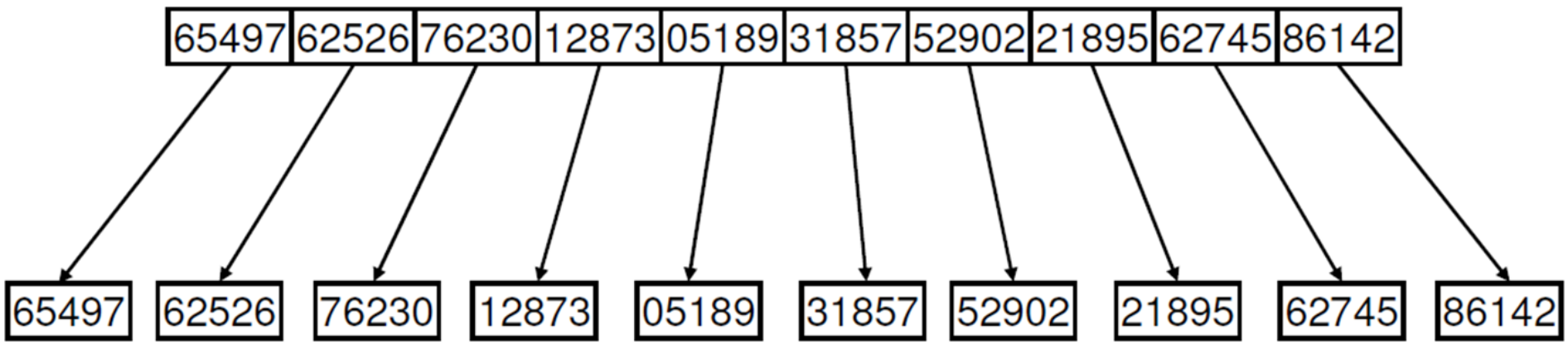
**Ziel**: Verteile Geheimnis so auf $n$ Parteien, dass man beliebige $k$ von ihnen zusammenbringen muss, um das Geheimnis zu lüften (z.B. Code zum Öffnen eines Tresors).

Beispiele:

- Geheimzahl eines Safes soll auf Kommission aus $n$ Personen verteilt werden, so dass je $k$ der $n$ Personen gemeinsam den Safe öffnen können.
- Alice möchte geheime Daten auf $n$ Servern speichern, so dass die einzelnen Server die Daten nicht rekonstruieren können, Alice jedoch aus den Teildaten von je $k$ Servern die Daten rekonstruieren kann.

# Aufteilen und Verteilen von Geheimnissen II

**Erste Idee**: 10-aus-10-Geheimnisteilung.




Unsicherheit über Geheimnis jeder einzelnen Person: $10^{45}$

Unsicherheit über Geheimnis von $k$ Personen: $10^{50-5k}$

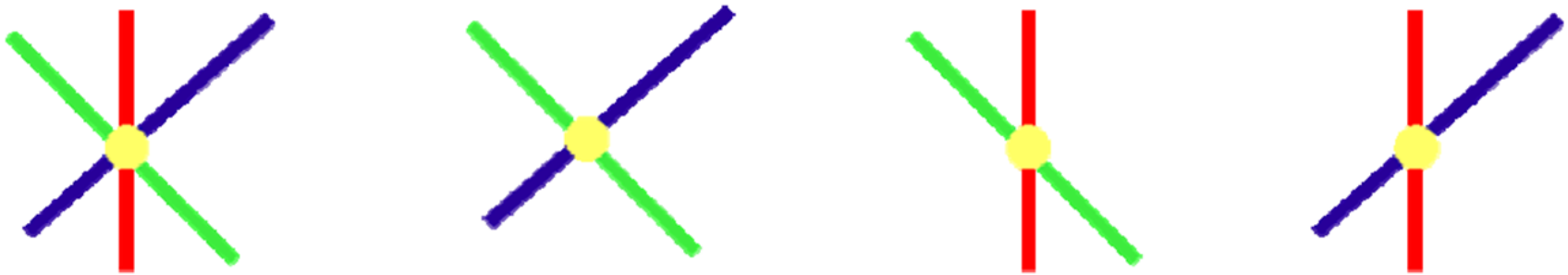
$\rightsquigarrow$ 9 aus 10 Personen lernen viel über Geheimzahl!

# Aufteilen und Verteilen von Geheimnissen III

**Zweite Idee**: Geometrische $2$-aus-$n$-Geheimnisteilung

Geheimnis: Punkt auf einer Ebene $P \in \mathbb{R}^2$

Teilgeheimnisse: $n$ Geraden $g_1, \ldots, g_n$ mit gemeinsamem Schnittpunkt $P$



Zwei oder mehr Teilnehmer gemeinsam kennen das Geheimnis (den Schnittpunkt).

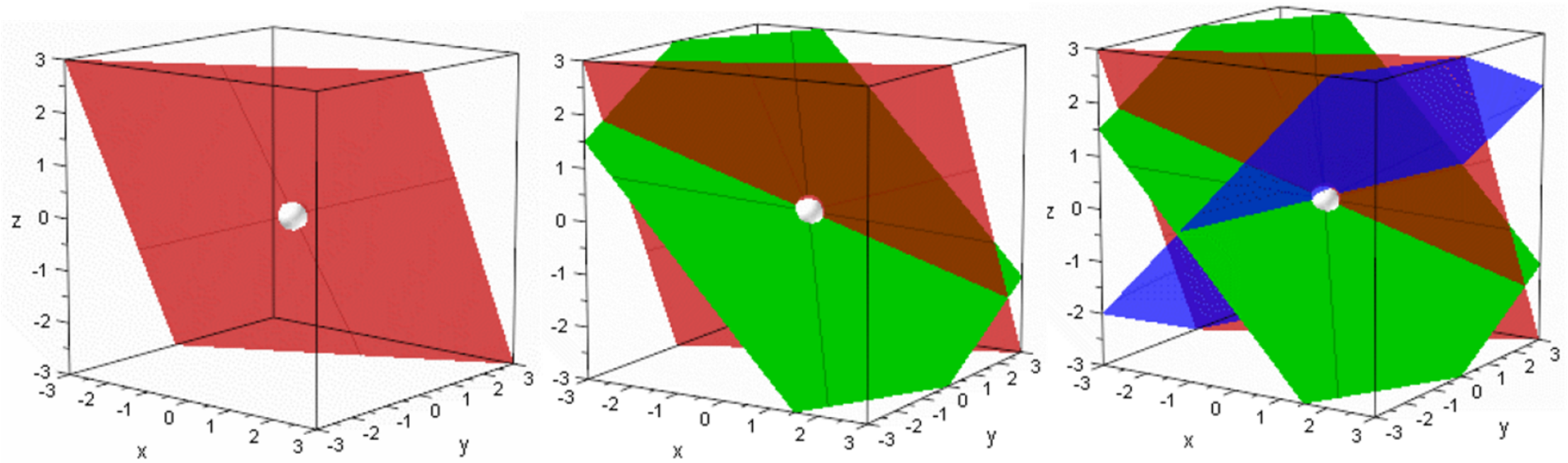
Ein Teilnehmer alleine kennt zwar seine Gerade, das Geheimnis kennt er nicht!

# Aufteilen und Verteilen von Geheimnissen IV

**Erweiterung der zweiten Idee**: Geometrische $3$-aus-$n$-Geheimnisteilung

Geheimnis: Punkt im $3$-dimensionalen Raum $P \in \mathbb{R}^3$

Teilgeheimnisse: $n$ Ebenen im $3$-dimensionalen Raum




Drei oder mehr Teilnehmer gemeinsam kennen das Geheimnis.
Zwei Teilnehmer alleine kennen das Geheimnis nicht, allerdings kennen sie ihre Schnittgerade.
Verfahren prinzipiell erweiterbar auf $k$-aus-$n$-Geheimnisteilung

# Aufteilen und Verteilen von Geheimnissen V

**Verfahren nach Adi Shamir**:
Betrachte allgemeine Parabelfunktion:

$$f(x) = ax^2 + bx + c$$

Kennt man drei Stützstellen (Punkte $(x, f(x))$), dann ist $f$ dadurch eindeutig bestimmt (*Lagrange-Interpolation*)

Vorgehen: Weise jedem Teilnehmer als Teilgeheimnis genau einen Punkt $(x, f(x))$ mit $x \neq 0$ auf der Parabel zu. Das Geheimnis $c$ sei dann als Funktionswert an der Stelle $x = 0$ kodiert.

Adi Shamir, 1952–

Verfahren erweiterbar auf $k$-aus-$n$-Geheimnisteilung, so dass weniger als $k$ Teilnehmer absolut nichts wissen!

# Aufteilen und Verteilen von Geheimnissen VI

**Beispiel für das Verfahren nach Adi Shamir**:
Sei $c = 3$ das Geheimnis, aufzuteilen auf 5 Teilnehmer, so dass mindestens 3 zusammenkommen müssen, um das Geheimnis zu lüften.

1. Wähle Polynom $p(x) = ax^2 + bx + c$ mit $a = 1, b = 2$ und $c = 3$.
2. Berechne 5 Punkte ($p(u_i)$: Teilgeheimnisse):
   $u_1 = 1, \quad u_2 = -1, \quad u_3 = 2, \quad u_4 = -2, \quad u_5 = 3$
   $p(u_1) = 6, \quad p(u_2) = 2, \quad p(u_3) = 11, \quad p(u_4) = 3, \quad p(u_5) = 18$

Kennt man drei Punkte $(u_i, p(u_i))$, dann ist $p$ dadurch eindeutig bestimmt:

Lagrange-Interpolation für Polynom $p(x)$ vom Grad $d$:

$$p(x) = \sum_{i=1}^{d+1} \left( p(u_i) \prod_{j \neq i} \frac{x - u_j}{u_i - u_j} \right)$$

# Commitment Scheme

**Ziel**: Auf eine Ausschreibung hin sollen mehrere Konkurrenten ihr Angebot bis zu einem bestimmten Ausschlusstermin einreichen. Vermeide, dass ein Konkurrent das Angebot eines anderen vorzeitig erfährt.

**Lösung**: Jeder Bieter schickt an den Ausschreiber sein Angebot verschlüsselt. Erst nach dem Ausschlusstermin wird dann der jeweilige Schlüssel zum Dekodieren des Angebots geschickt!

Dazu muss sichergestellt sein, dass die Verschlüsselungsfunktion injektiv ist (damit es zu einem Chiffre-Text höchstens einen passenden Klartext gibt).

# Quantenkryptologie

Ziel: Sichere Schlüsselübertragung für symmetrische Verfahren.

**Idee**: Nutze quantenphysikalische Effekte von Photonen (Heisenberg'sche Unschärferelation) zur sicheren Signalübertragung.

Entscheidend: Jedwedes Messen bzw. Mithören eines solchen Signals verändert es auch. $\rightsquigarrow$ Ein Abhörer wird bemerkt!

Anwendung zum Beispiel zur sicheren Schlüsselübertragung für One-Time-Pad.