# Integritätsprüfung der Ergebnisse homomorpher Operationen mit Secret Sharing Mechanismen

Einleitungsvortrag für eine Bachelorarbeit

Matthias Ulbrich

Betreuerin: Dipl.-Inf. Saffija Kasem-Madani

Leitung: Prof. Michael Meier

Informatik 4 – IT Security – Universität Bonn

universitätbonn

Rheinische
Friedrich-Wilhelms-
Universität Bonn

# Übersicht

1. Motivation des Themas
2. Grundlagen
    1. Das Pailler Kryptosystem
    2. Shamir‘s Secret Sharing
3. Anwendungsfall der Bachelorarbeit
4. Zeitplan

# 1. Motivation des Themas

**Hintergrund:**

- Datenergebung zur Optimierung von Prozessen, Forschungszwecken …

  Datenschutz ⟶ Pseudonymisierung

# 1. Motivation des Themas

**Hintergrund:**

- Datenergebung zur Optimierung von Prozessen, Forschungszwecken …

  Datenschutz ⟶ Pseudonymisierung

- Um pseudonymisierte Daten verknüpfen zu können wird der gleiche Klartext $cl$ auf verschiedene Arten vorverarbeitet.

  $n$ Verfügbarkeitsoptionen:

$$ps(cl) = (\underbrace{p_1(cl)}_{=}, \underbrace{p_2(cl)}_{+}, \dots, p_n(cl))$$

# 2.1 Pailler Verschlüsselung

- **Schutzziel:** Verarbeitung von personenbezogenen Daten ohne deren Kenntnisnahme
- **Maßnahme:**
  **Additiv-homomorphe asymmetrische Verschlüsselung mit Pailler**
  Addition von Chiffren ohne vorherige Entschlüsselung

# 2.1 Pailler Verschlüsselung

- **Schutzziel:** Verarbeitung von personenbezogenen Daten ohne deren Kenntnisnahme
- **Maßnahme:**
  **Additiv-homomorphe asymmetrische Verschlüsselung mit Pailler**
  Addition von Chiffren ohne vorherige Entschlüsselung

konkret:
Anstelle von Addition auf Klartexten führt man ein äquivalente Operation auf dem Bild unter einem **Isomorphismus** durch.

$$Pailler_{k_{ö}}(m_1) * Pailler_{k_{ö}}(m_2) = Pailler_{k_{ö}}\ (m_1 + m_2)$$

# 2.1 Pailler Kryptosystem: Ausnutzung von Mallebalilty

- **Mallebility**:
  Gegeben Chiffrat $c$ von $m$, kann der Angreifer $c'$ berechnen welches das Chiffrat zu $f(m)$ darstellt.
- Bei Pailler ist $f$ eine Funktion die Chiffrate addiert.
  Beispiel für eine bösartige Anfrage:

$$Pailler_{k_{\ddot{o}}}(m) * Pailler_{k_{\ddot{o}}}(0) = Pailler_{k_{\ddot{o}}}(m + 0)$$
$$= Pailler'_{k_{\ddot{o}}}(m)$$

- Homomorphe Verschlüsselungsverfahren wie Pailler sind anfällig für Malleability **per Design**.

# 2.1 Shamir‘s Secret Sharing

- **Schutzziel:** Verhindern des Missbrauchs eines Geheimnisses durch eine Partei (minimales Vertrauen)
- **Maßnahme:** Geheimnis abhängig machen von mehreren **Shares** welche auf mehrere Parteien verteilt werden.

# 2.1 Shamir‘s Secret Sharing

- **Schutzziel:** Verhindern des Missbrauchs eines Geheimnisses durch eine Partei (minimales Vertrauen)
- **Maßnahme:** Geheimnis abhängig machen von mehreren **Shares** welche auf mehrere Parteien verteilt werden.

hier: Shamir‘s Secret Sharing
geheimer Schlüssel als Nullstelle eines unbekannten Polynoms, Polynom mit Shares rekonsturierbar

(k,n) Schema

- n Parteien insgesamt
- ein k-Tupel kann entschlüsseln


# 2.1 Shamir‘s Secret Sharing: Teilen des Geheimnisses

1. Wähle zufälliges Polynom:

$$p(x) = \sum_{i=0}^{k} a_i x^i = \underbrace{a_0}_{secret} + \sum_{i=1}^{k} a_i x^i$$

$(x_i, y_i)$ sind die zu teilenden Shares

# 2.1 Shamir‘s Secret Sharing: Teilen des Geheimnisses

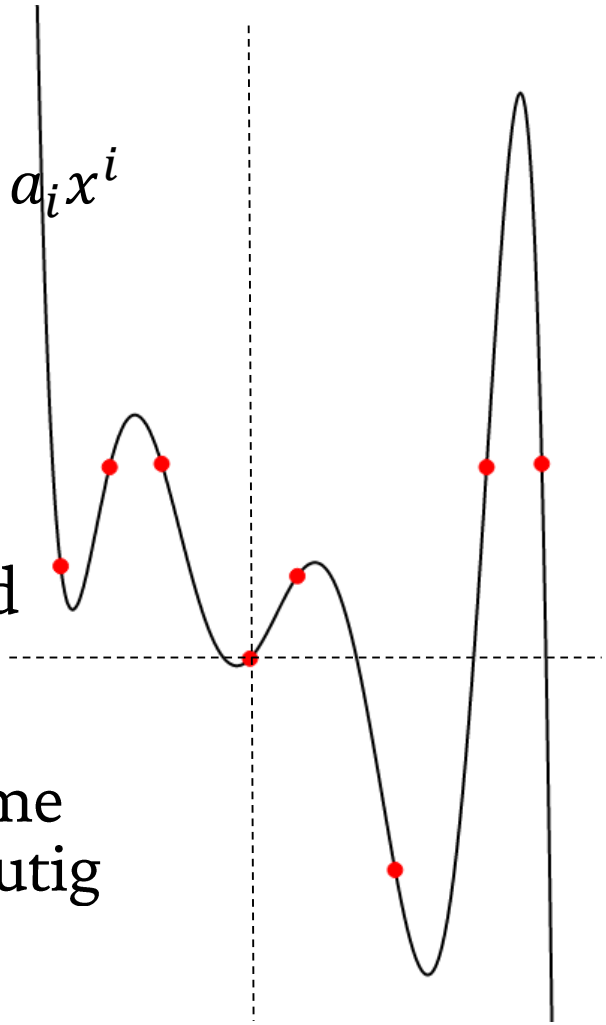
1. Wähle zufälliges Polynom:

$$p(x) = \sum_{i=0}^{k} a_i x^i = \underbrace{a_0}_{secret} + \sum_{i=1}^{k} a_i x^i$$

$(x_i, y_i)$ sind die zu teilenden Shares

2. Erzeuge n-k viele zusätzliche Stütz-punkte zur Rekonstruktion.
**Überbestimmtes** Polynom vom Grad n > k: bestimmt durch k+1 Parameter

bis k-1 viele Schlüssel: ∞-viele Polynome
k bis n viele Schlüssel: Polynom eindeutig



# 2.1 Shamir‘s Secret Sharing: Rekonstruktion des Geheimnisses

Lagrange-Interpolation, Rekonstruktion des Polynoms $p$ anhand der Shares $(x_i, y_i)$

Idee:

$$\mathrm{p}(x) = \sum_{i=0}^{k} \delta_{x_i}(x) y_i \qquad \delta_{x_i}(x) = \begin{cases} 0, falls\ x = x_j \\ 1, falls\ x = x_i \\ *, sonst \end{cases}$$

Wie muss $\delta_{x_i}$ aussehen?

# 2.1 Shamir‘s Secret Sharing: Rekonstruktion des Geheimnisses

Lagrange-Interpolation, Rekonstruktion des Polynoms $p$ anhand der Shares $(x_i, y_i)$

Idee:

$$\mathrm{p}(x) = \sum_{i=0}^{k} \delta_{x_i}(x) y_i \qquad \delta_{x_i}(x) = \begin{cases} 0, falls\ x = x_j \\ 1, falls\ x = x_i \\ *, sonst \end{cases}$$

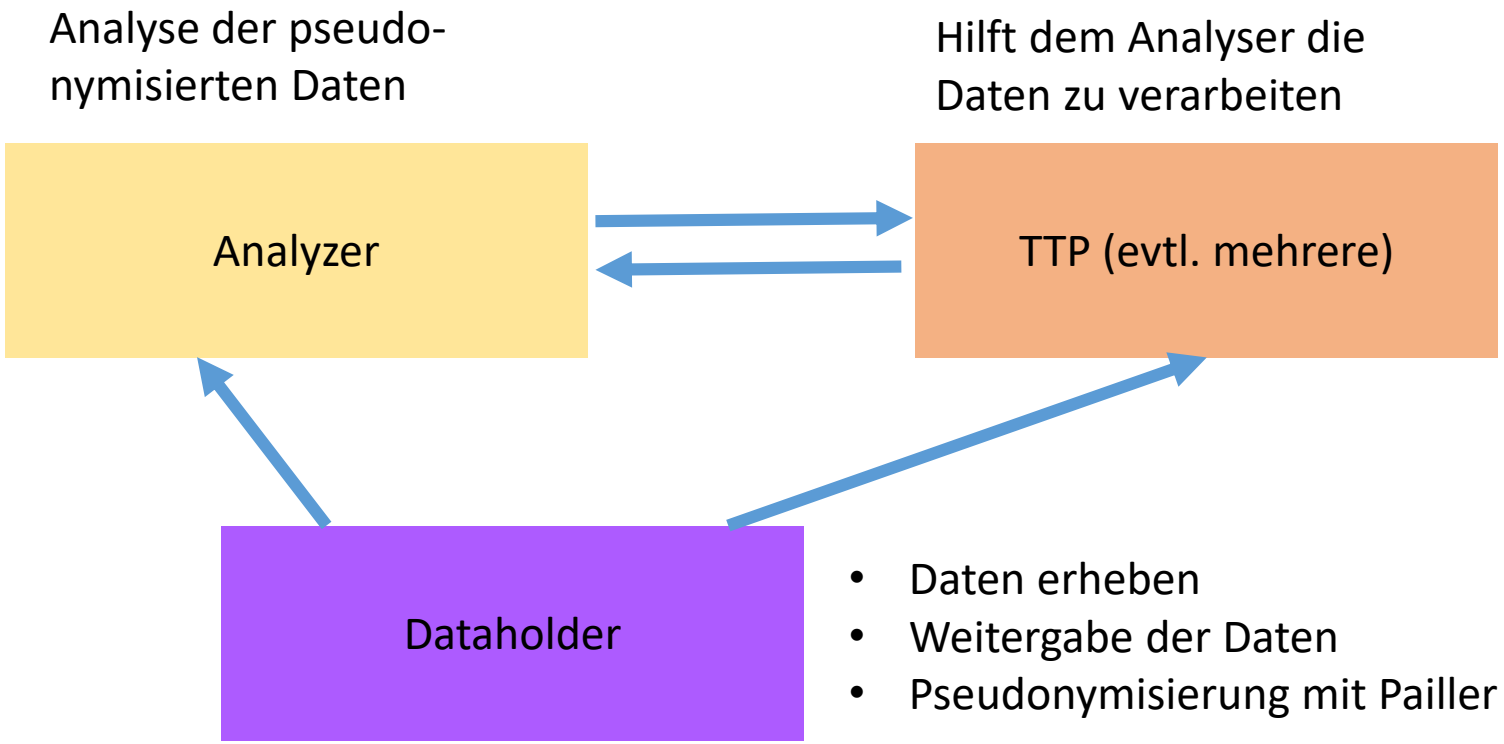
Wie muss $\delta_{x_i}$ aussehen?

$$\delta_{x_i}(x) = \prod_{i \neq j} (x - x_j)(x_i - x_j)^{-1}$$

Anmerkung:
Man rechnet in Primzahlrestklassengruppen $\mathbb{Z}_p$

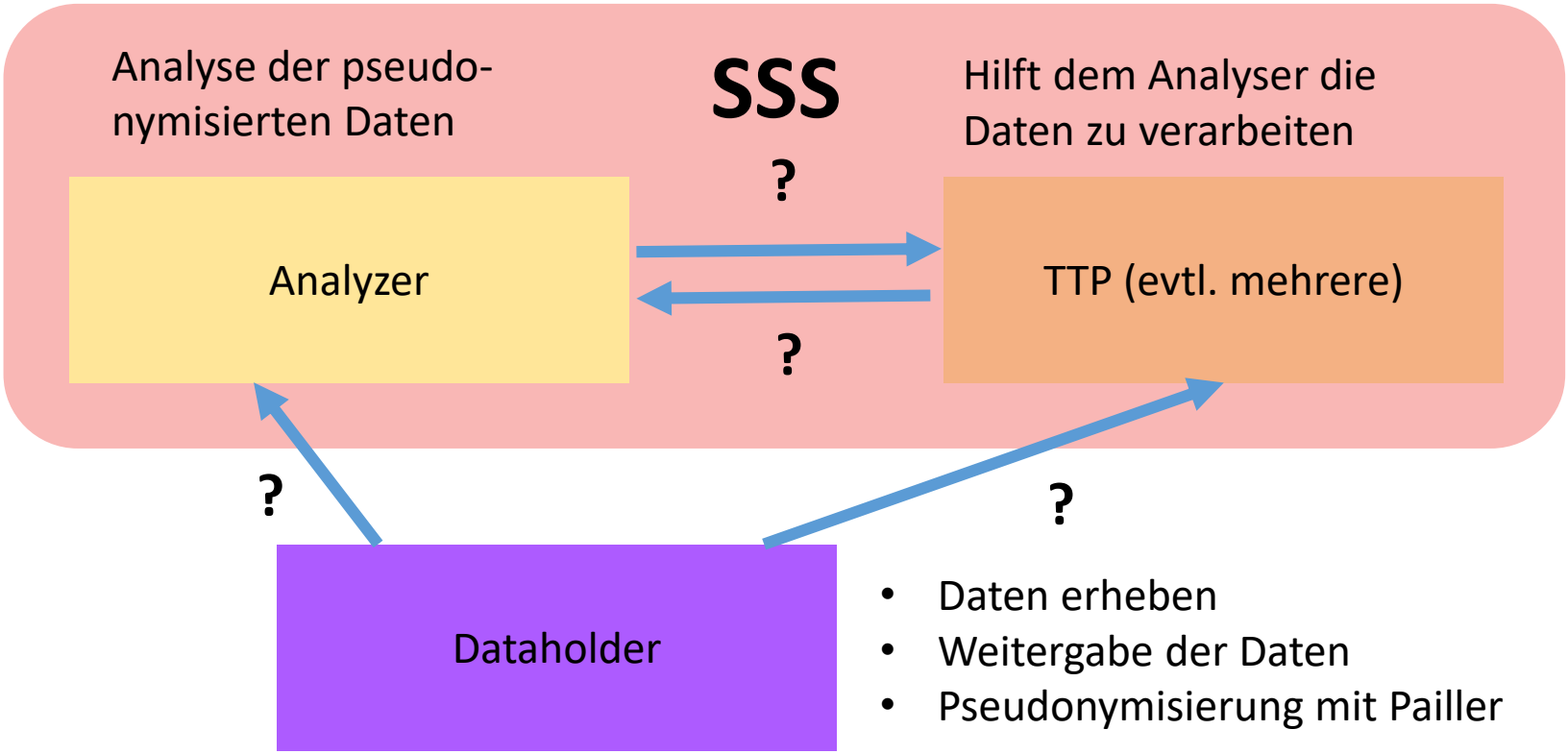
# 3. Anwendungsfall der Bachelorarbeit

Verarbeitung Pseudonymisierter Daten mit Secure Multiparty Computation bei minimalen Vertrauen



# 3. Anwendungsfall der Bachelorarbeit

1. **Design eines Verarbeitungsshemas** mit Secret Sharing (minimales Vertrauen)

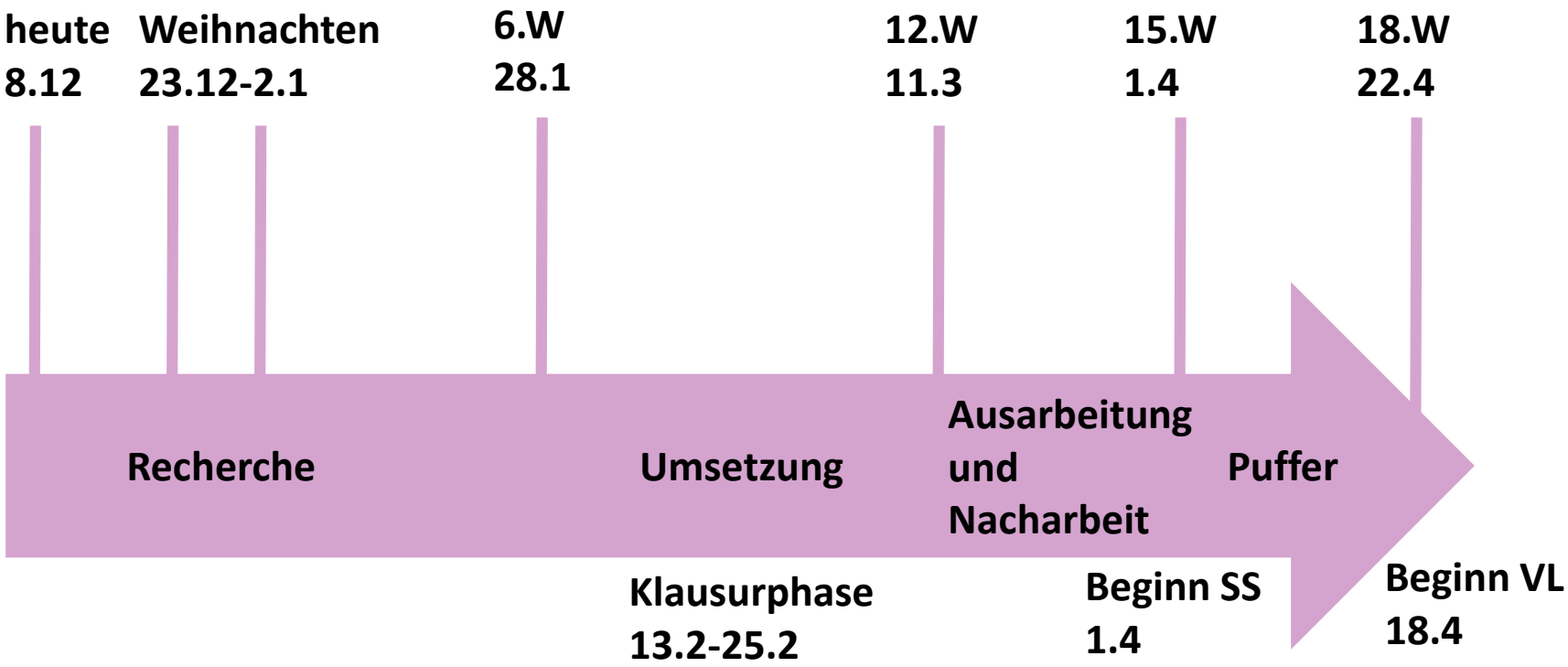


# 3. Anwendungsfall der Bachelorarbeit

1. **Design eines Verarbeitungsshemas** mit Secret Sharing (minimales Vertrauen)
   - Welche minimalen Informationen muss der Dataholder zur Verfügung stellen?
2. **Entwurf eines Protokolls** für die Kommunikation zwischen den Shareholdern
3. **Analyse und Bewertung**:
   - Malleability?
   - Performance
   - Evtl. Schwachstellen

# 4. Zeitplan für die Bachelorarbeit

- Insgesamt 4.5 Monate/18 Wochen;
  6W Recherche, 6W Umsetzung, 3W Ausarbeitung und Nacharbeit, 3W Puffer



# Anhang

Literaturnachweise


- 1979 Communications of the ACM Vol 22 Number 11, *How to Share a Secret*, Adi Shamir,
- 2008, *Introduction into Modern Cryptography*, Jonathan Katz and Yehuda Lindell
- 2012, *Krypto 2 VL 10, RUB*, Eike Kiltz http://www.crypto.ruhr-uni-bochum.de/imperia/md/content/Kiltz/12/ss12/krypto2/10.pdf
- 2006, *Paillier Cryptosystem: A Mathematical Introduction*, Tobias Volkhausen http://www2.cs.uni-paderborn.de/cs/ag-bloemer/lehre/proseminar_WS2005/material/Volkhausen_Ausarbeitung.pdf?PHPSESSID=edd79a27cb022cfa749b4a2baa7ea6f0
- 2012, *Homomophe Verschlüsselung*, Sophie Friedrich et al. https://www.cosy.sbg.ac.at/~uhl/PScrypt12/HomomorphicEncryption.pdf
- https://en.wikipedia.org/wiki/Paillier_cryptosystem
- https://de.wikipedia.org/wiki/Paillier-Kryptosystem
- https://en.wikipedia.org/wiki/Malleability_(cryptography)


Bildnachweise


- Folie 7: https://images-na.ssl-images-amazon.com/images/G/01/th/aplus/sentrysafe/B005P12C5A-1.jpg http://www.kaba.com/media-resized/321576/v13/resized752x-1/schluessel.jpg
- Folie 9: https://upload.wikimedia.org/wikipedia/commons/4/4c/Polynomial-interpolation.svg