

Software Innovation Campus Paderborn

# Verschlüsselte Regelungen für vernetzte Systeme

Stand der Technik und offene Probleme

Moritz Schulze Darup

Lehrstuhl für Regelungs- und Automatisierungstechnik  
Fakultät für Elektrotechnik, Informatik und Mathematik



Software Innovation Campus Paderborn

# Verschlüsselte **Regelungen für vernetzte Systeme**

Stand der Technik und offene Probleme

Moritz Schulze Darup

Lehrstuhl für Regelungs- und Automatisierungstechnik  
Fakultät für Elektrotechnik, Informatik und Mathematik



## Crashkurs Regelungstechnik



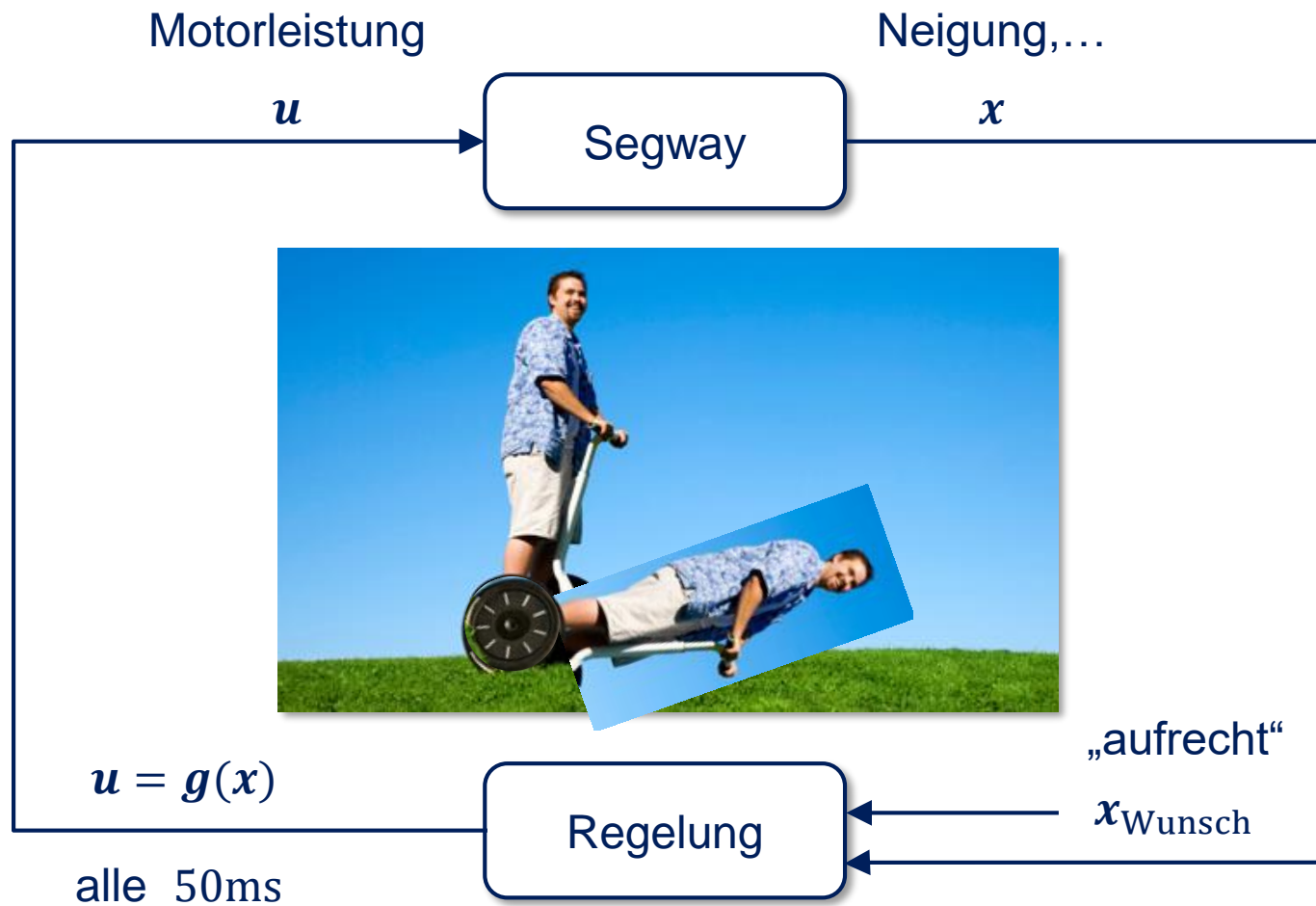
## Crashkurs Regelungstechnik



„aufrecht“

$x_{\text{Wunsch}}$

## Crashkurs Regelungstechnik



## Anwendungen: Gegenwart und Zukunft

- Klassische Regelungssysteme – „Isolierte Systeme“

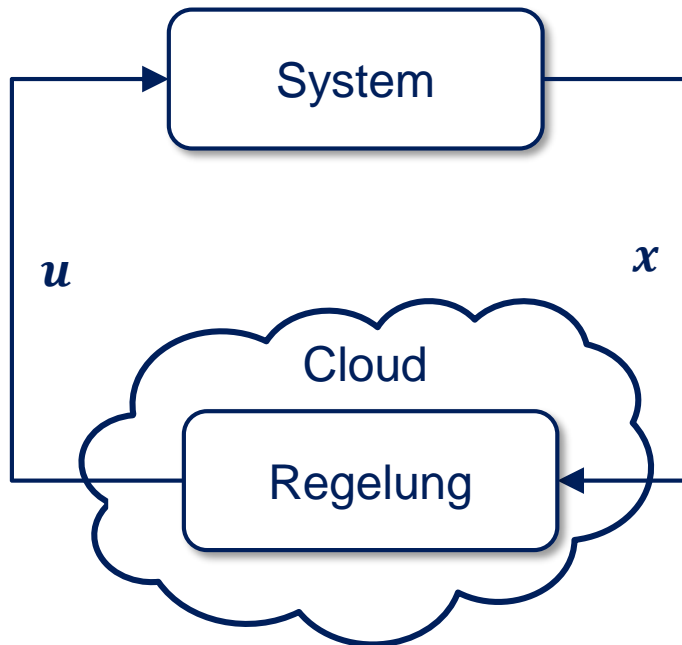


- Moderne Regelungssysteme – „Vernetzte Systeme“

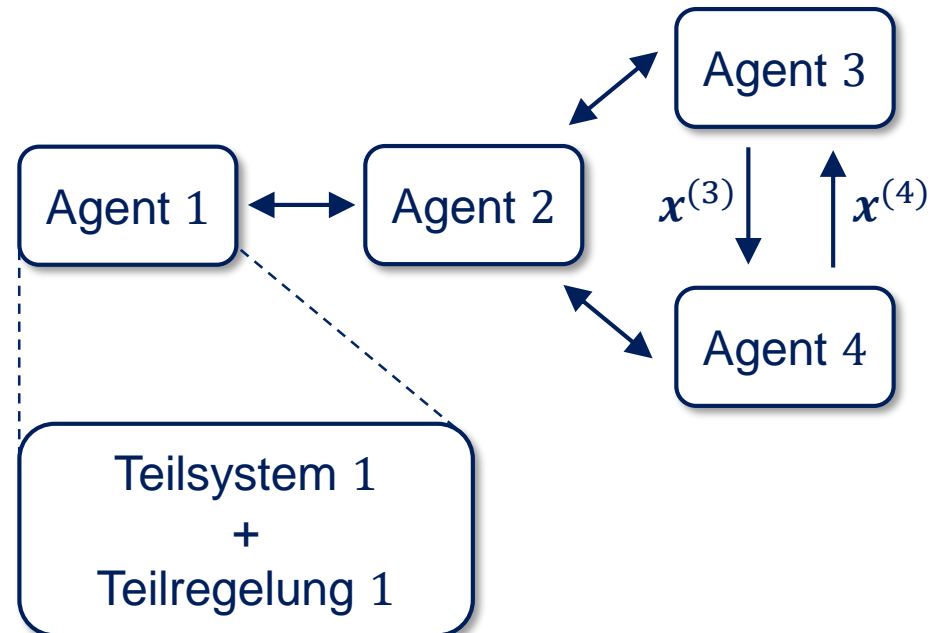


## Vernetzte Systeme erfordern vernetzte Regelungen

Cloud-basierte Regelung



Verteilte Regelung für verteilte Systeme



- Gemeinsamkeit: **Kommunikation sensibler Prozessdaten**

Software Innovation Campus Paderborn

# **Verschlüsselte** Regelungen für vernetzte Systeme

## Stand der Technik und offene Probleme

Moritz Schulze Darup

Lehrstuhl für Regelungs- und Automatisierungstechnik  
Fakultät für Elektrotechnik, Informatik und Mathematik







## Thema ist hochaktuell

ZEIT  ONLINE

10.01.2019

Politik Gesellschaft Wirtschaft Kultur ▾ Wissen **Digital** Campus ▾ Arbeit Entdecken Sport ZEITmagazin Podcasts mehr ▾



DATENSICHERHEIT

**Private Daten Hunderter vern. Sys. veröffentlicht**

## Thema ist hochaktuell

ZEIT  ONLINE

Hier nur deterministische  
und kausale Systeme ;-)

10.01.2019

Politik Gesellschaft Wirtschaft Kultur ▾ Wissen **Digital** Campus ▾ Arbeit Entdecken Sport ZEITmagazin Podcasts mehr ▾



DATENSICHERHEIT

# Private Daten Hunderter Politiker veröffentlicht

- Angriffe auf Regelungssysteme: Stuxnet, Duqu, Industroyer, Triton...

Software Innovation Campus Paderborn

# **Verschlüsselte Regelungen** für vernetzte Systeme

## Stand der Technik und offene Probleme

Moritz Schulze Darup

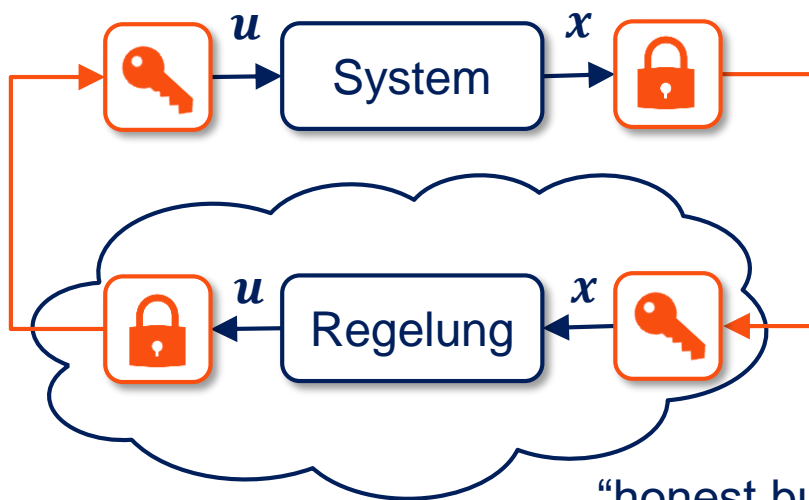
Lehrstuhl für Regelungs- und Automatisierungstechnik  
Fakultät für Elektrotechnik, Informatik und Mathematik



## Verschlüsselte Cloud-basierte Regelung

- Ziel: Datensicherheit während Übertragung und in Cloud

### Verschlüsselte Kommunikation



- + einfache Umsetzung
- keine Vertraulichkeit in Cloud

### Verschlüsselte Regelung



- + durchgehende Verschlüsselung
- Realisierung?

## Schlüsseltechnologie

- **Homomorphe Verschlüsselungsverfahren** erlauben einfache mathematische Operationen auf verschlüsselten Daten
- Das Paillier-Kryptosystem [Paillier1999] stellt beispielsweise Operationen  $\alpha$  und  $\mu$  bereit, so dass  $z_1 + z_2 = \text{Dec}\left(\alpha(\text{Enc}(z_1), \text{Enc}(z_2))\right)$  und  $z_1 z_2 = \text{Dec}\left(\mu(z_1, \text{Enc}(z_2))\right)$
- Ermöglicht voll **verschlüsselte Additionen** und **Multiplikationen** mit einem verschlüsselten Faktor
- Erlaubt **verschlüsselte** Cloud-basierte Auswertung **linearer Regelungen**

$$\mathbf{u} = \mathbf{g}(\mathbf{x}) = \mathbf{K}\mathbf{x} = \begin{pmatrix} K_{11}x_1 + \cdots + K_{1n}x_n \\ \vdots \\ K_{m1}x_1 + \cdots + K_{mn}x_n \end{pmatrix}$$

- Erstmalig beschrieben in [Kogiso2015]

Software Innovation Campus Paderborn

# Verschlüsselte Regelungen für vernetzte Systeme

Stand der Technik und offene Probleme

Moritz Schulze Darup

Lehrstuhl für Regelungs- und Automatisierungstechnik  
Fakultät für Elektrotechnik, Informatik und Mathematik



## Verschlüsselung komplexer Regelungen - Agenda

- Verschlüsselte Cloud-basierte **modellprädiktive Regelungen (MPR)**
- Verschlüsselte verteilte **kooperative Regelungen**

1 Crashkurs MPR

2 Verschlüsselte MPR mittels **Offline-Optimierung**

3 Verschlüsselte MPR mittels Online-Optimierung

4 Crashkurs kooperative Regelung

5 Verschlüsselte kooperative Regelung mittels **Offline-Strukturierung**

## Verschlüsselung komplexer Regelungen - Agenda

- Verschlüsselte Cloud-basierte modellprädiktive Regelungen (MPR)
- Verschlüsselte verteilte kooperative Regelungen

**1** Crashkurs MPR

**2** Verschlüsselte MPR mittels Offline-Optimierung

**3** Verschlüsselte MPR mittels Online-Optimierung

**4** Crashkurs kooperative Regelung

**5** Verschlüsselte kooperative Regelung mittels Offline-Strukturierung



## MPR ist wie (vorausschauendes) Autofahren



- MPR basiert auf folgender Strategie:
  1. Berechnung der **optimalen Eingriffssequenz** basierend auf dem **aktuellen Zustand  $x(k)$**  für einen endlichen **Prädiktionshorizont  $N$**
  2. Aufschalten des „ersten“ Eingriffs und Wiederholung der Prozedur im nächsten Zeitschritt  $k + 1$
- Stärke liegt in der simultanen Berücksichtigung eines **Gütekriterien** und von **Zustands- und Eingriffsbeschränkungen**

## Optimierung führt auf quadratisches Programm

- Für „einfache“ lineare Systeme folgt die **optimale Eingriffssequenz**

$$\mathbf{z}^*(\mathbf{x}) := \begin{pmatrix} \hat{\mathbf{u}}^*(0) \\ \vdots \\ \hat{\mathbf{u}}^*(N-1) \end{pmatrix} = \mathbf{u}(k)$$

aus der Lösung eines **quadratisches Programms (QP)** der Form

$$\min_{\mathbf{z}} \frac{1}{2} \mathbf{z}^\top \mathbf{H} \mathbf{z} + \mathbf{x}^\top \mathbf{F}^\top \mathbf{z} \quad \text{s.t.} \quad \mathbf{G} \mathbf{z} \leq \mathbf{E} \mathbf{x} + \mathbf{d}$$

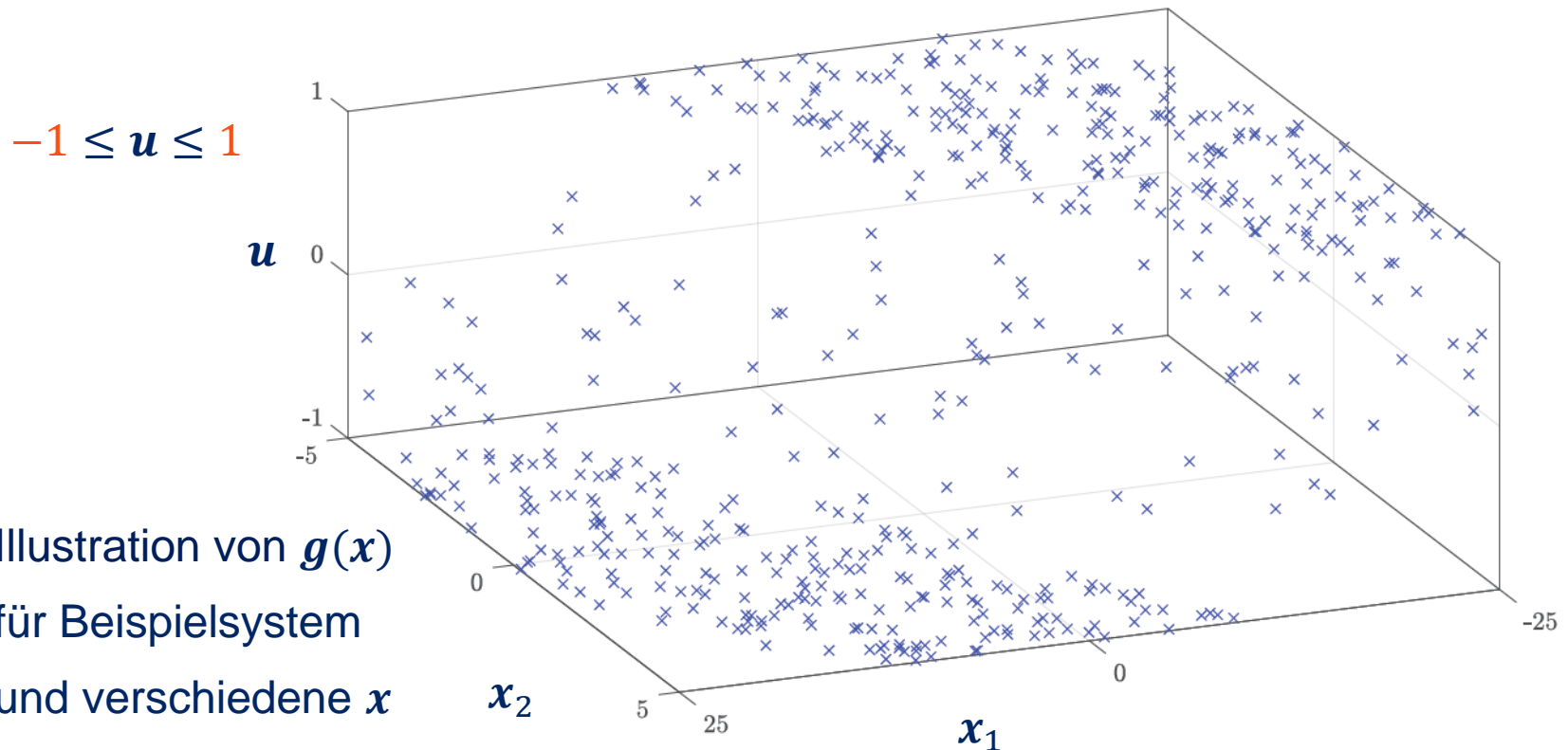
↑  
zustandsabhängiges  
quadratisches  
**Gütekriterium**

↑  
zustandsabhängige  
affine  
**Beschränkungen**

- Im Prinzip ist dieses QP in jedem Zeitschritt für  $\mathbf{x} = \mathbf{x}(k)$  zu lösen

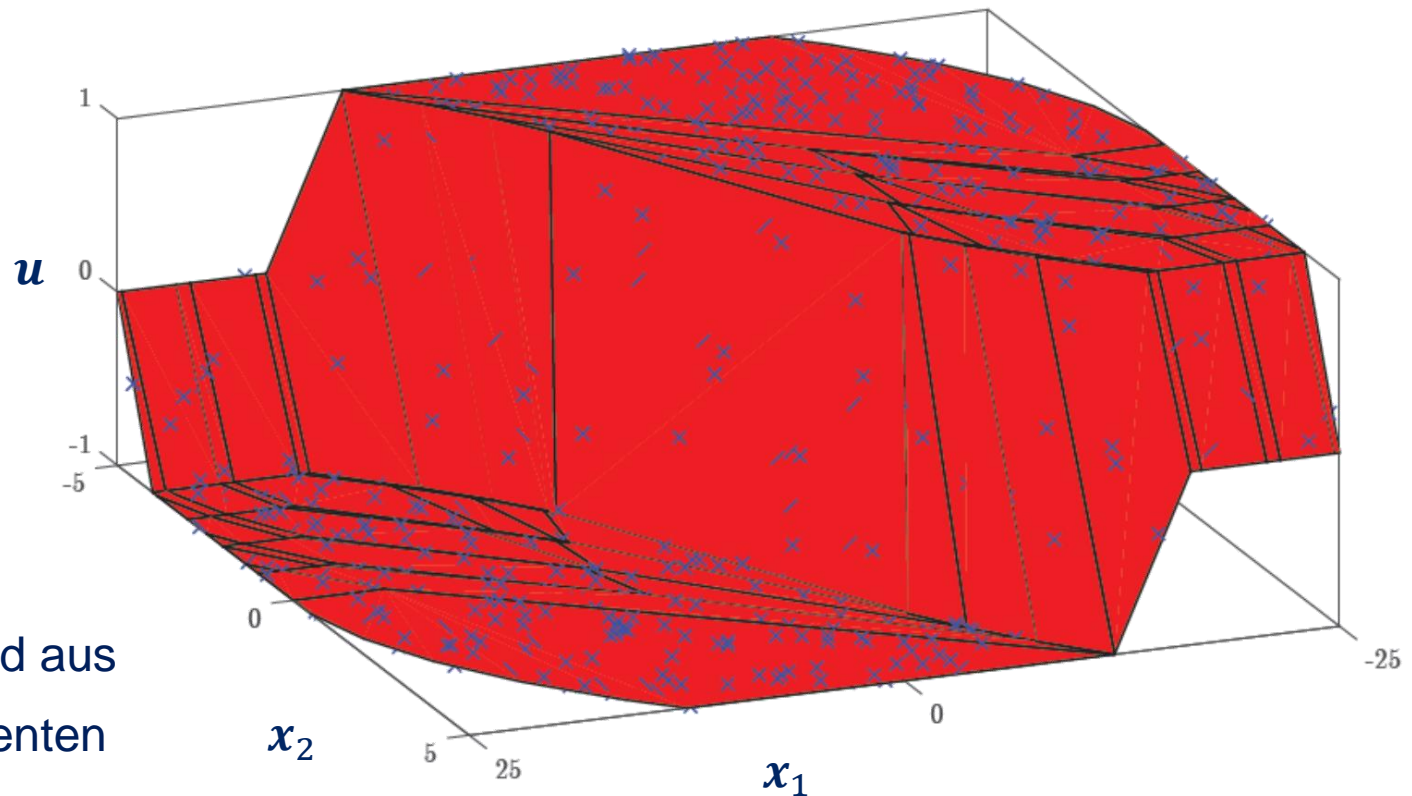
## Illustration einer MPR

- Da der **Regelungseingriff** jedoch nur vom **Zustand** abhängt (und nicht vom Zeitpunkt  $k$ , an dem er auftritt) gilt auch hier (indirekt)  $u = g(x)$



## Struktur einer MPR

- Es lässt sich zeigen, dass eine MPR für lineare Systeme in ein **stückweise affines Regelgesetz**  $u = g(x)$  mündet



hier bestehend aus  
 $s = 83$  Segmenten

## Offline-Optimierung

- Das stückweise affine Regelgesetz lässt sich **offline** mittels **parametrischer Optimierung** berechnen und man erhält

$$\mathbf{g}(\mathbf{x}) = \begin{cases} \mathbf{K}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} & \text{falls } \mathbf{x} \in \mathcal{R}_1 \\ \vdots & \vdots \\ \mathbf{K}^{(s)}\mathbf{x} + \mathbf{b}^{(s)} & \text{falls } \mathbf{x} \in \mathcal{R}_s \end{cases}$$

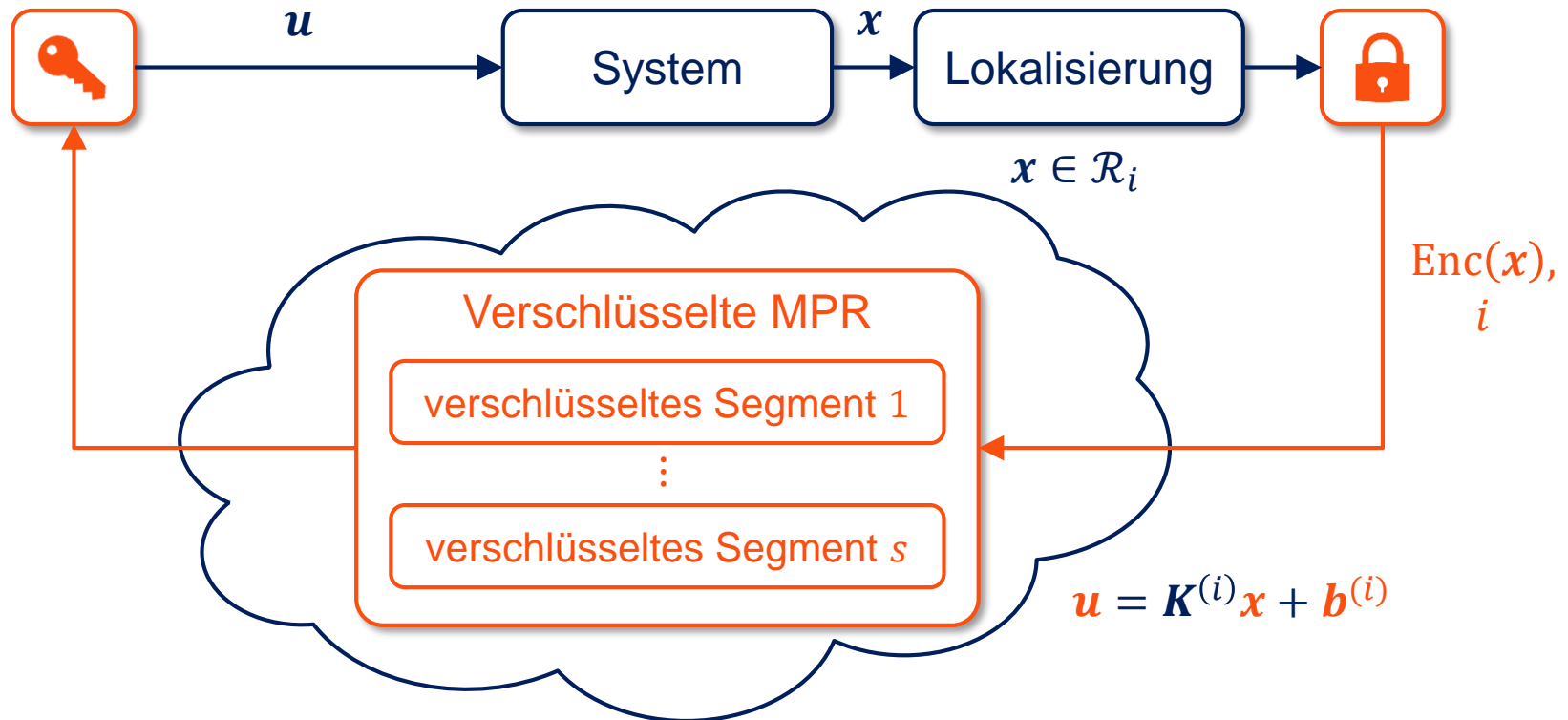
- Die **Regionen**  $\mathcal{R}_i$  sind Polytope (Vielecke) im Zustandsraum
- Die Struktur der einzelnen Segmente ähnelt  $\mathbf{u} = \mathbf{K}\mathbf{x}$  aus der Intro
- Tatsächlich gilt für  $\mathbf{x} \in \mathcal{R}_i$

$$\mathbf{u} = \mathbf{g}(\mathbf{x}) = \mathbf{K}^{(i)}\mathbf{x} + \mathbf{b}^{(i)} = \begin{pmatrix} K_{11}^{(i)}x_1 + \dots + K_{1n}^{(i)}x_n + b_1^{(i)} \\ \vdots \\ K_{m1}^{(i)}x_1 + \dots + K_{mn}^{(i)}x_n + b_n^{(i)} \end{pmatrix}$$

- Verschlüsselung** kann analog zur Intro über Paillier erfolgen

## Verschlüsselte MPR

- Die **Lokalisierung** des Zustands erfolgt am („smarten“) **Sensor**



- Architektur des Reglers ist **suboptimal**; liefert aber **Machbarkeitsnachweis**

## Verschlüsselung komplexer Regelungen - Agenda

- Verschlüsselte Cloud-basierte modellprädiktive Regelungen (MPR)
- Verschlüsselte verteilte kooperative Regelungen

**1** Crashkurs MPR

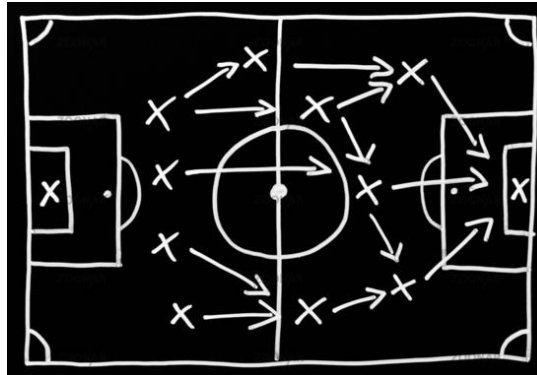
**2** Verschlüsselte MPR mittels Offline-Optimierung

**3** Verschlüsselte MPR mittels Online-Optimierung

**4** Crashkurs kooperative Regelung

**5** Verschlüsselte kooperative Regelung mittels Offline-Strukturierung

## Kooperative Regelung ist wie Fußballspielen (mit Taktik)



- Kooperative Regelung basiert auf folgender Strategie:
  1. In jedem Zeitschritt werden **lokale Zustände  $x^{(i)}$**  (oder weitere Infos) mit benachbarten „Agenten“ **ausgetauscht**
  2. Basierend auf den vorliegenden Zuständen berechnet jeder Agent einen **lokalen Regelungseingriff  $u^{(i)}$**  mit Blick aufs **gemeinsame Ziel**
- Stärke liegt in der **dezentralen Reglerimplementierung**, die häufig deutlich „schlanker“ ist als eine zentralisierte Lösung



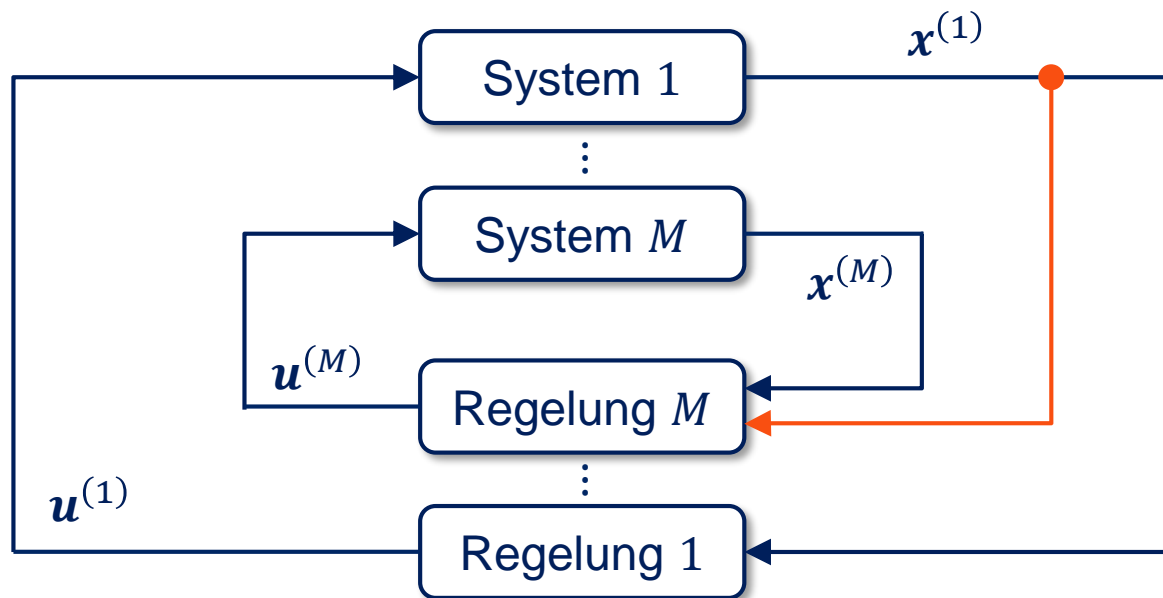
## Kooperative Regelung

- Die lokalen Regelungsgesetze weisen also die Form

$$g^{(i)}(\mathbf{x}^{(i)}, \{\mathbf{x}^{(j)} \mid j \in \mathcal{N}^{(i)}\})$$

auf, wobei  $\mathcal{N}^{(i)}$  die Menge der Nachbarn des Agenten  $i$  bezeichnet

- Verteilte kooperative Regelung **zentralisiert** interpretiert:



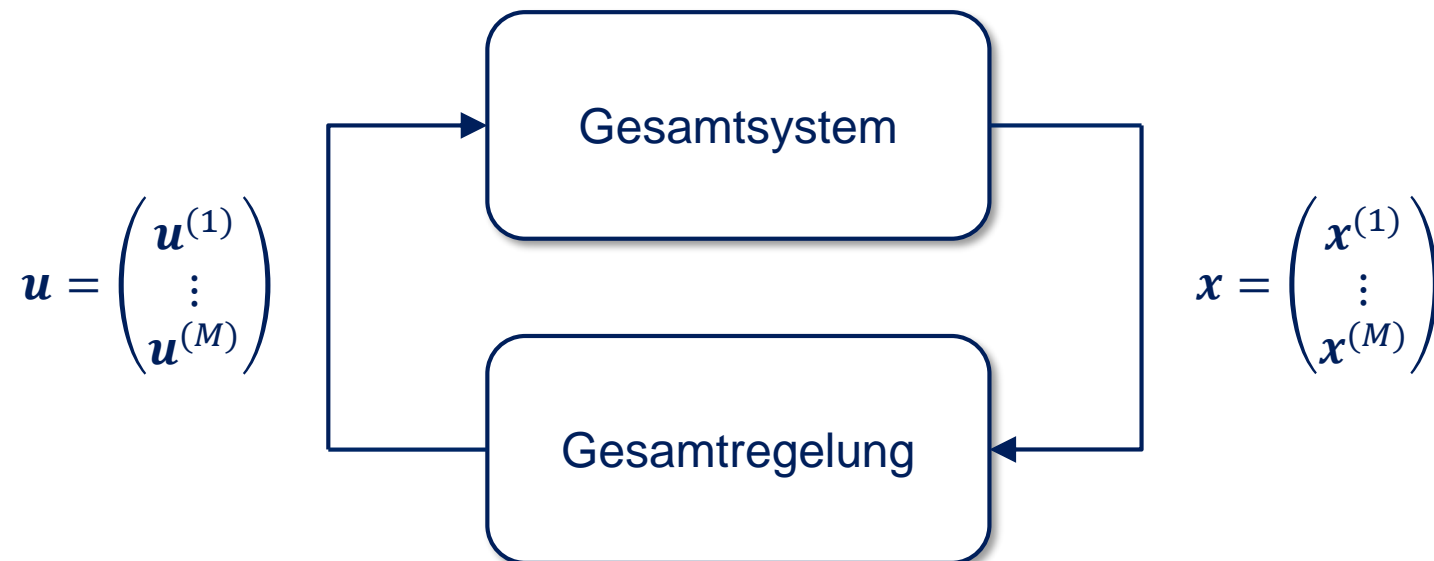
## Kooperative Regelung

- Die lokalen Regelungsgesetze weisen also die Form

$$g^{(i)}(x^{(i)}, \{x^{(j)} \mid j \in \mathcal{N}^{(i)}\})$$

auf, wobei  $\mathcal{N}^{(i)}$  die Menge der Nachbarn des Agenten  $i$  bezeichnet

- Verteilte kooperative Regelung **zentralisiert** interpretiert:



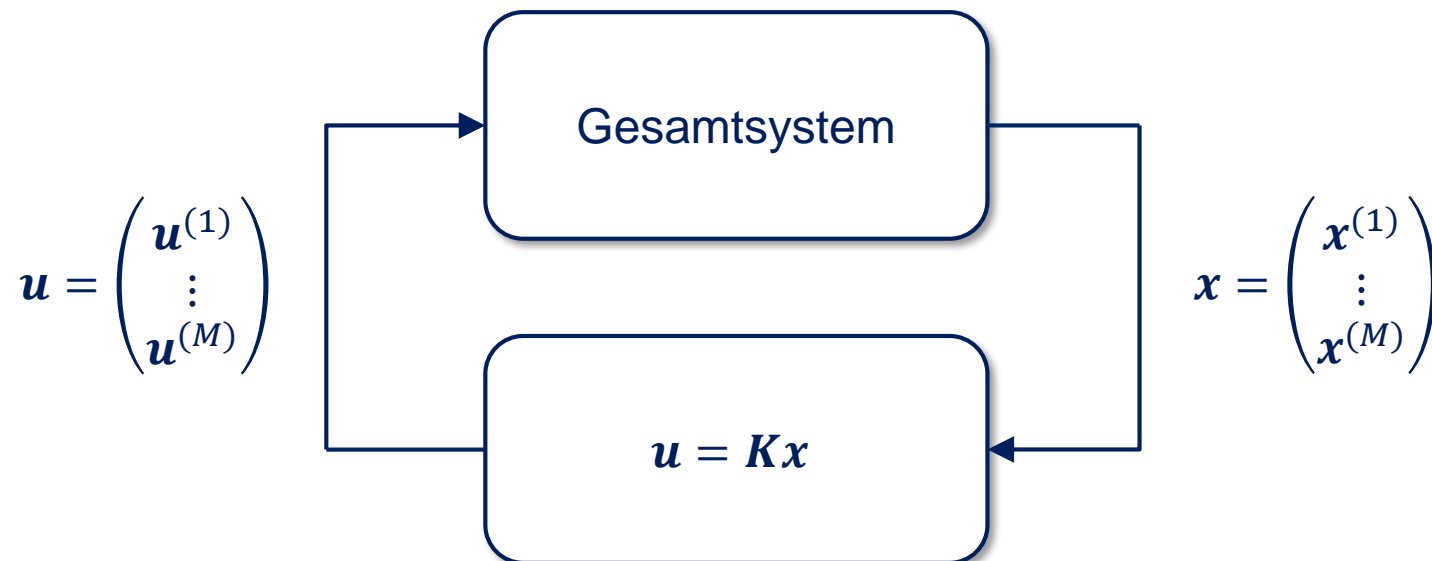
## Lineare kooperative Regelung?

- Die lokalen Regelungsgesetze weisen also die Form

$$g^{(i)}(x^{(i)}, \{x^{(j)} \mid j \in \mathcal{N}^{(i)}\})$$

auf, wobei  $\mathcal{N}^{(i)}$  die Menge der Nachbarn des Agenten  $i$  bezeichnet

- Verteilte kooperative Regelung **zentralisiert** interpretiert:



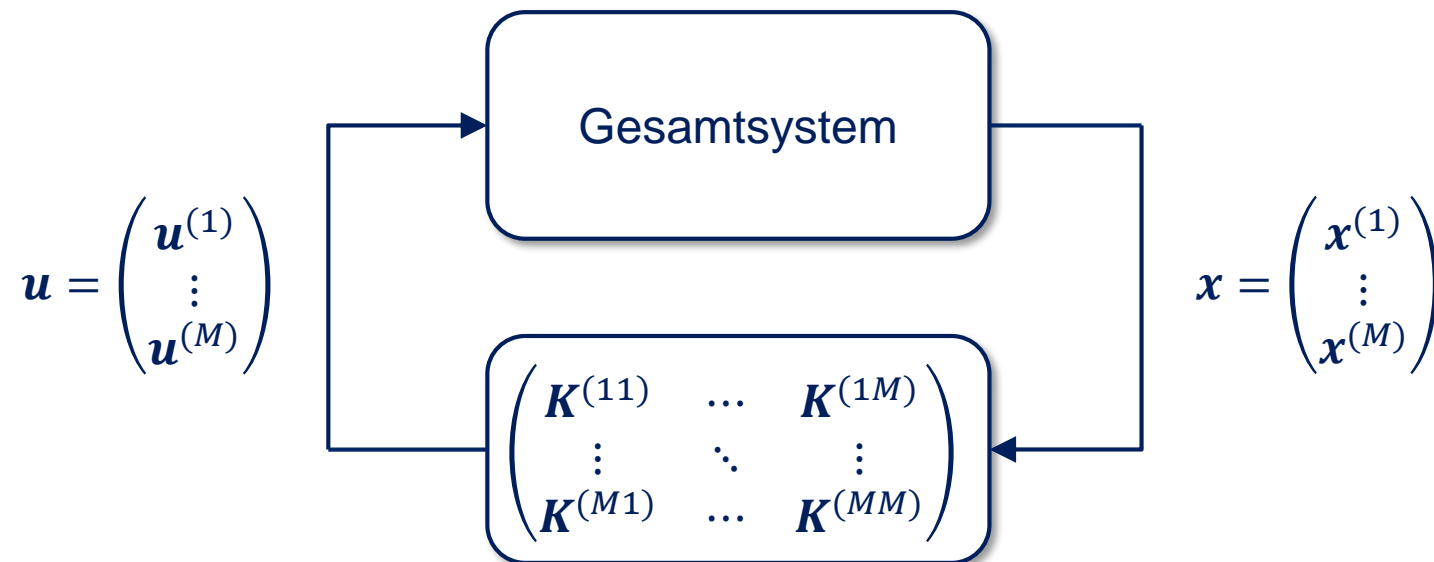
## Lineare kooperative Regelung?

- Die lokalen Regelungsgesetze weisen also die Form

$$g^{(i)}(x^{(i)}, \{x^{(j)} \mid j \in \mathcal{N}^{(i)}\})$$

auf, wobei  $\mathcal{N}^{(i)}$  die Menge der Nachbarn des Agenten  $i$  bezeichnet

- Verteilte kooperative Regelung **zentralisiert** interpretiert:

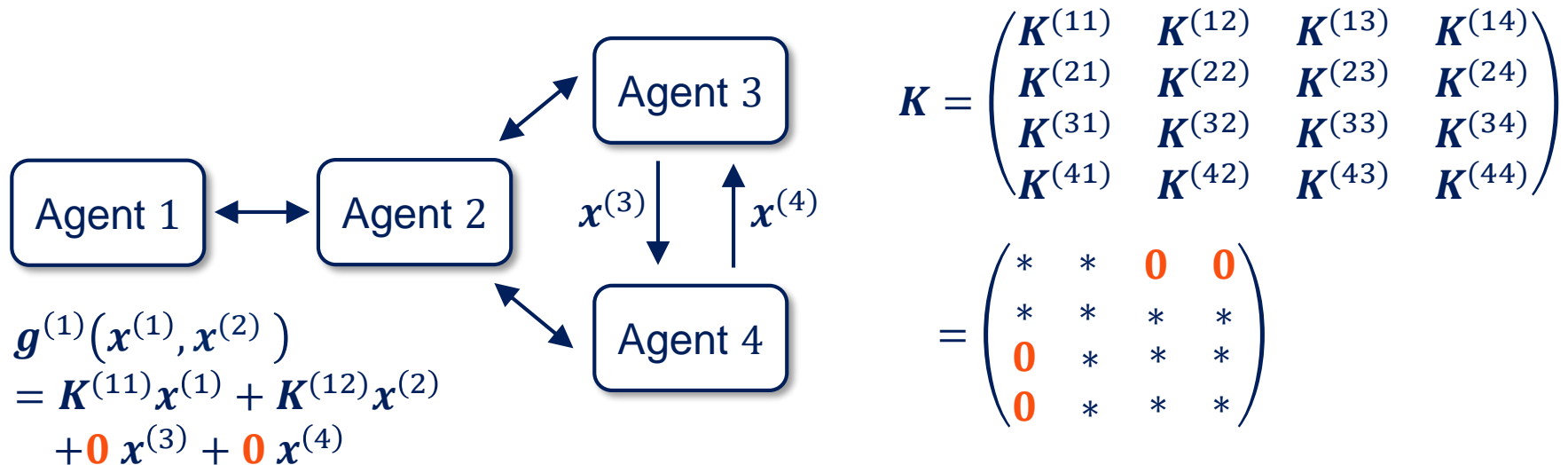


## Lineare kooperative Regelung?

- Problem: Vollbesetzte Matrix  $K$ , die nur Blöcke  $K^{(ij)} \neq \mathbf{0}$  aufweist, verletzt – außer für vollständigen Kommunikationsgraph – die Struktur

$$g^{(i)}(\mathbf{x}^{(i)}, \{\mathbf{x}^{(j)} \mid j \in \mathcal{N}^{(i)}\})$$

- Tatsächlich würde dann  $g^{(i)}(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)})$  gelten
- Abhilfe schafft **strukturierte Reglermatrix  $K$**



## Lineare kooperative Regelung!

- Strukturierte Reglermatrix  $K$  lässt sich **offline** berechnen (sofern existent)
- Berechnung ist (für unstrukturierte Initialisierung) **nicht trivial**
- Systematische Lösung glückte erst in [Lin2011]
- Es stellt sich die Frage, wie die resultierenden Regelungen

$$g^{(i)}(x^{(i)}, \{x^{(j)} \mid j \in \mathcal{N}^{(i)}\}) = K^{(ii)}x^{(i)} + \sum_{j \in \mathcal{N}^{(i)}} K^{(ij)}x^{(j)}$$

**verschlüsselt** ausgewertet werden können

- Insbesondere gilt es die **Zustände**  $x^{(j)}$  benachbarter Agenten vor Einsichtnahme des Agenten  $i$  zu schützen
- Die Agenten sollen darüber hinaus keine Informationen über die **Regelungsstrategie** benachbarter Agenten erhalten (wird gleich wichtig)

## Verschlüsselte kooperative Regelung

- Der direkte Zugriff auf  $x^{(j)}$  durch Agent  $i$  lässt sich verhindern, indem Agent  $j$  anstelle von  $x^{(j)}$  den Regelungsanteil  $v^{(ij)} := K^{(ij)}x^{(j)}$  übermittelt
- Agent  $i$  berechnet dann

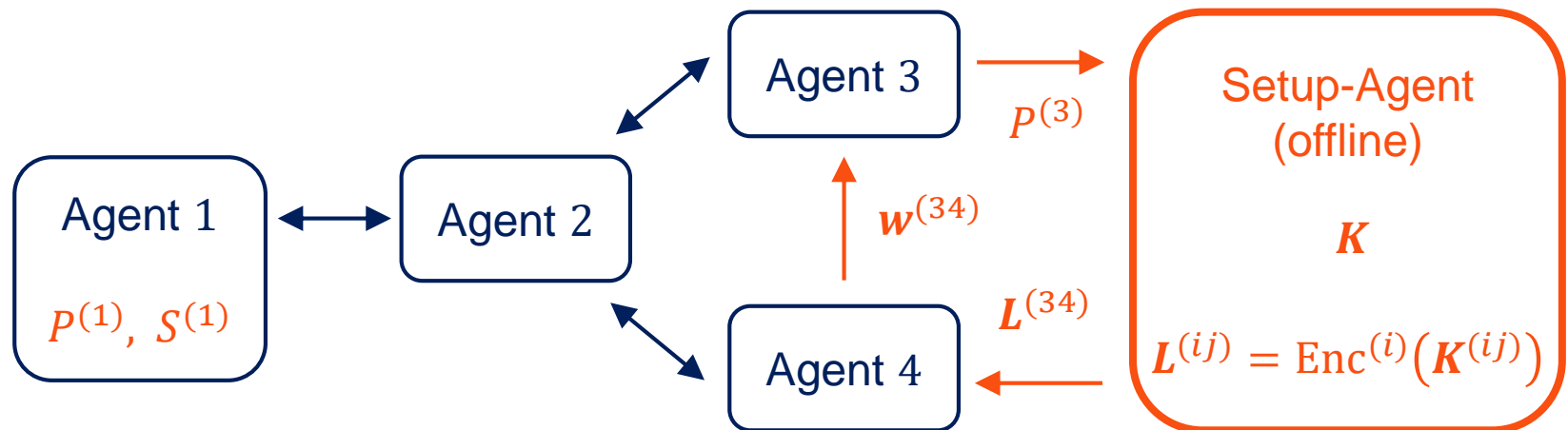
$$g^{(i)}(x^{(i)}, \{x^{(j)} \mid j \in \mathcal{N}^{(i)}\}) = K^{(ii)}x^{(i)} + \sum_{j \in \mathcal{N}^{(i)}} v^{(ij)}$$

- Kennt Agent  $i$  die (Einträge und die Dimension der) Matrix  $K^{(ij)}$  nicht, so ist die **Rekonstruktion von  $x^{(j)}$  aus  $v^{(ij)}$  unmöglich**
- Agent  $j$  hat mit  $K^{(ij)}$  und  $v^{(ij)}$  jedoch Infos über die **Strategie** von  $i$
- Diese Infos lassen sich jedoch wiederum über Paillier **verschlüsseln**:

$$v^{(ij)} = K^{(ij)}x^{(j)} = \begin{pmatrix} K_{11}^{(ij)}x_1^{(j)} + \dots + K_{1n}^{(ij)}x_n^{(j)} \\ \vdots \\ K_{m1}^{(ij)}x_1^{(j)} + \dots + K_{mn}^{(ij)}x_n^{(j)} \end{pmatrix}$$

## Implementierung der Regelung

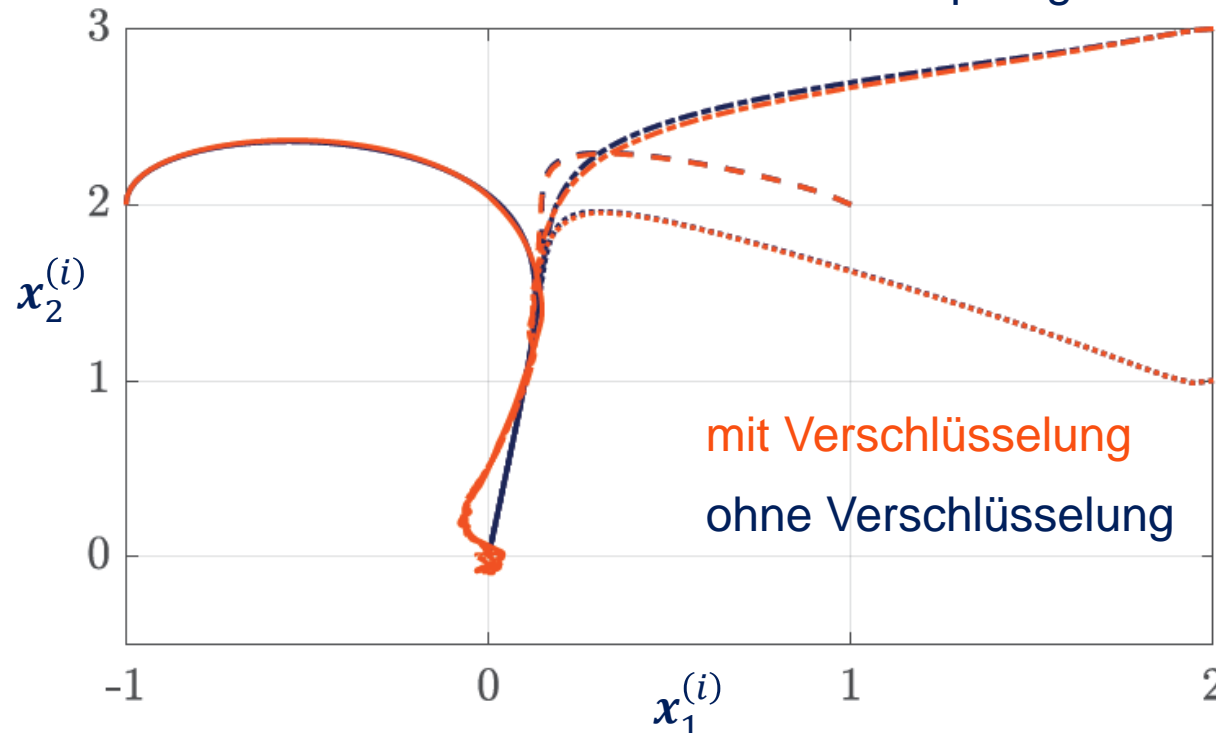
1. Jeder Agent generiert öffentlichen und geheimen Schlüssel  $P^{(i)}$  und  $S^{(i)}$
2. Ein **Setup-Agent** berechnet die strukturierte Matrix  $K$  offline
3. Er verschlüsselt  $K^{(ij)}$  mittels  $P^{(i)}$  und sendet das Resultat  $L^{(ij)}$  zu Agent  $j$
4. **Online** berechnet Agent  $j$  die **verschlüsselten**  $v^{(ij)}$  basierend auf  $L^{(ij)}$  und  $x^{(j)}$  und sendet das Resultat  $w^{(ij)}$  zu Agent  $i$
5. Agent  $i$  berechnet  $u^{(i)} = K^{(ii)}x^{(i)} + \sum_{j \in \mathcal{N}^{(i)}} \text{Dec}^{(i)}(w^{(ij)})$  mittels  $S^{(i)}$





## Beispiel für eine kooperative Regelung

- Vier Roboter sollen Schwarm bilden und Ursprung anfahren



- Gemeinsames Ziel wird trotz Quantisierungseffekten erreicht
- Verschlüsselung erfordert  $\approx 0.1s$  Rechenzeit in jedem Zeitschritt

Software Innovation Campus Paderborn

# Verschlüsselte Regelungen für vernetzte Systeme

Stand der Technik und **offene Probleme**

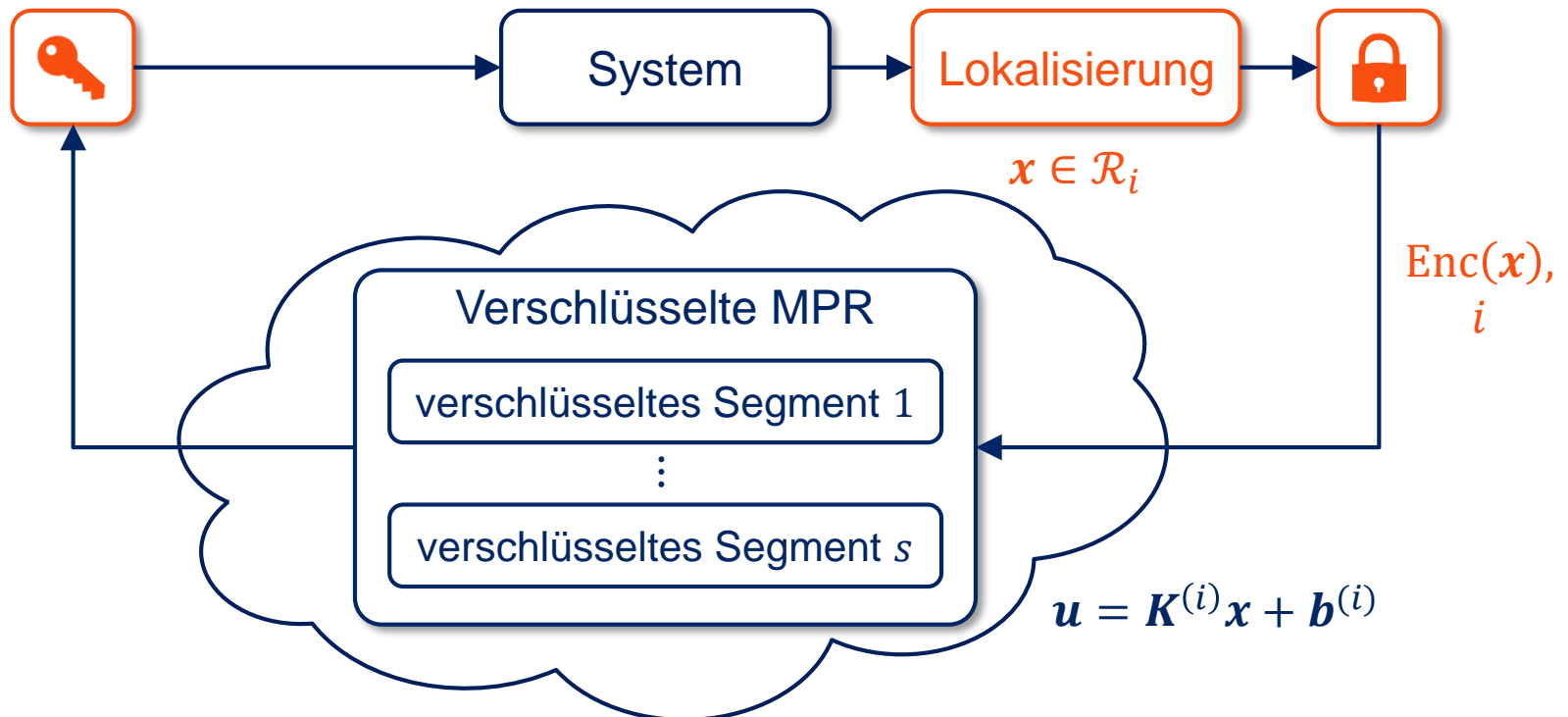
Moritz Schulze Darup

Lehrstuhl für Regelungs- und Automatisierungstechnik  
Fakultät für Elektrotechnik, Informatik und Mathematik



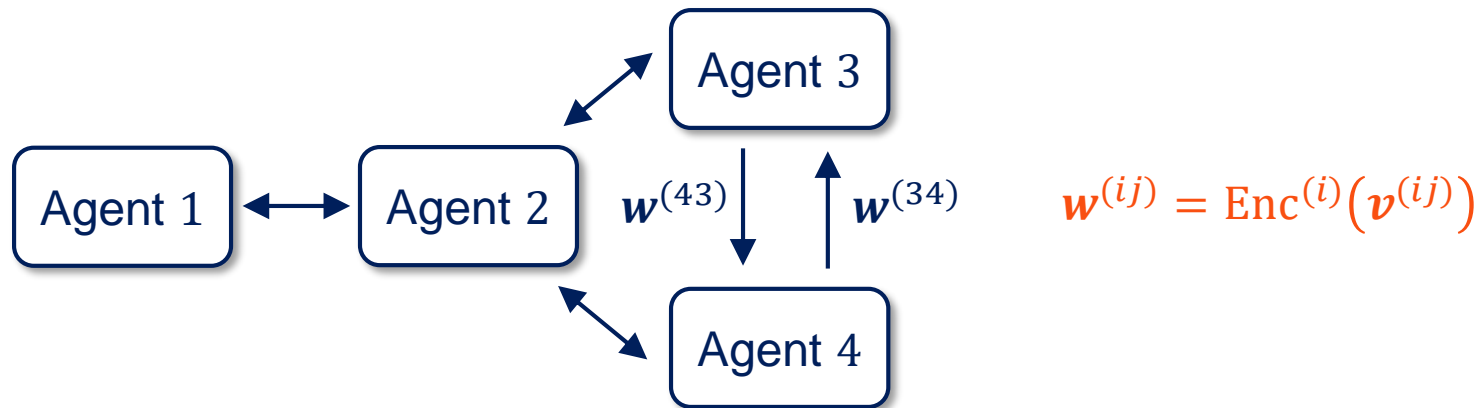
## Probleme und Ausblick

- Cloud-basierte verschlüsselte Regelungen erfordern **viel Rechenleistung außerhalb der Cloud** (asymmetrische Verschlüsselung + Lokalisierung)
- Unverschlüsselte Übermittlung der Region  $i$  birgt **Sicherheitsrisiken**



## Probleme und Ausblick

- Verteilte verschlüsselte Regelung basiert auf der Annahme, dass es unmöglich / schwierig ist  $\mathbf{x}^{(j)}$  aus  $\mathbf{v}^{(ij)} = \mathbf{K}^{(ij)} \mathbf{x}^{(j)}$  zu rekonstruieren
- Kenntnisse über die Dimension von  $\mathbf{K}^{(ij)}$  sind jedoch häufig bekannt



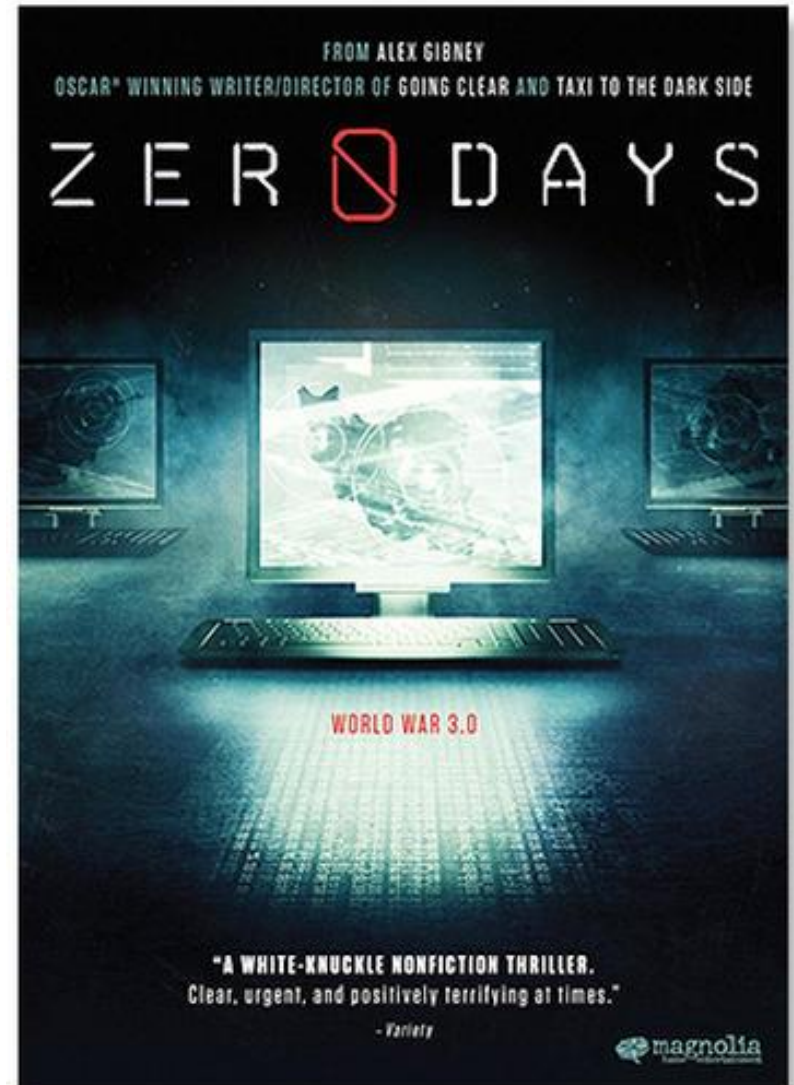
- Bisher wird nahezu ausschließlich das Paillier-Kryptosystem verwendet
- Andere homomorphe Verschlüsselungen (partiell, gelevelt, voll) als auch Secret Sharing und Multi-Party Computation bieten großes Potenzial

**That's it!**

**Vielen Dank für ihre  
Aufmerksamkeit!**

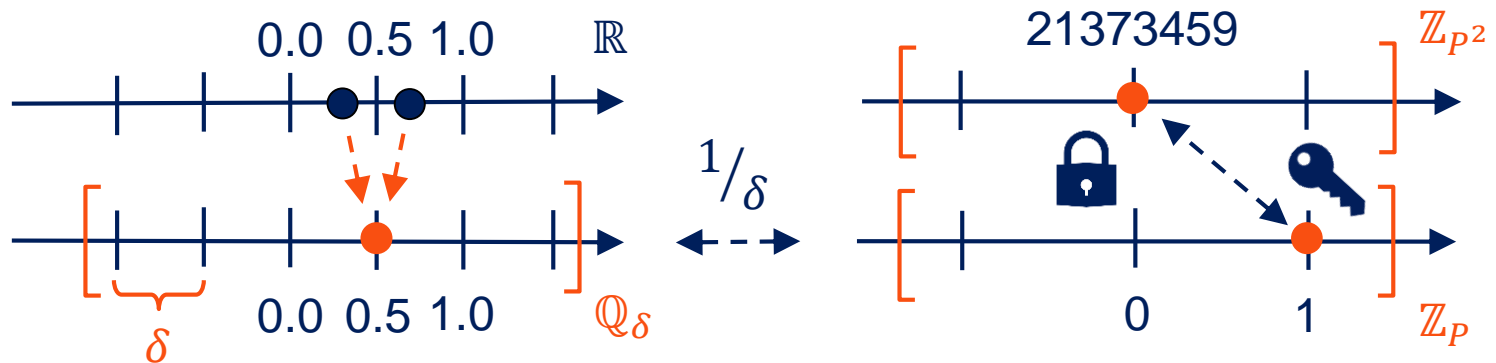
**Fragen?**

**Spannende Doku über Stuxnet →**



## Backup: Quantisierung

- Kryptosysteme erlauben **Verschlüsselung ganzer Zahlen** endlicher Größe
- Verschlüsselung erfordert **Quantisierung** (mit Bereich & Auflösung)



- Problem: Quantisierung kann zu **Instabilität** führen [Delchamps1990]
- Gewährleistung von Stabilität durch **robuste Regelung**

## Backup: Paillier Kryptosystem

- Wähle **zwei große Primzahlen**  $p_1$  und  $p_2$ , so dass die Produkte  $p_1p_2$  und  $(p_1 - 1)(p_2 - 1)$  keinen gemeinsamen Teiler außer 1 aufweisen
- Der **öffentliche Schlüssel** ist dann  $P = p_1p_2$
- Der **geheime Schlüssel**  $S$  ist kleinstes gem. Vielf. von  $p_1 - 1$  und  $p_2 - 1$
- Die **Verschlüsselung** ganzer Zahlen  $z \in \{0, \dots, P - 1\}$  erfolgt nun über

$$\text{Enc}(z, r) := (P + 1)^z r^P \bmod P^2,$$

wobei  $r$  eine **Zufallszahl** aus einer Untermenge von  $\{0, \dots, P - 1\}$  ist

- Die **Entschlüsselung** eines Ciphertexts  $c \in \{0, \dots, P^2 - 1\}$  geschieht über

$$\text{Dec}(c) := \left\lfloor \frac{(c^S \bmod P^2) - 1}{P} \right\rfloor (S^{-1} \bmod P) \bmod P$$

- Die Verschlüsselung ist umkehrbar in dem Sinne, dass  **$\text{Dec}(\text{Enc}(z, r)) = z$**  für alle  $z \in \{0, \dots, P - 1\}$  gilt

## Backup: Homomorphe Eigenschaften

- Per Konstruktion erhalten wir

$$\text{Enc}(z_1 + z_2, r_1 r_2) := \text{Enc}(z_1, r_1) \text{Enc}(z_2, r_2) \bmod P^2,$$

für alle  $z_1, z_2 \in \{0, \dots, P - 1\}$  mit  $z_1 + z_2 \in \{0, \dots, P - 1\}$

- Diese Eigenschaft erlaubt **verschlüsselte Addition** und entspricht der Operation  $\alpha(\text{Enc}(z_1), \text{Enc}(z_2))$
- Darüber hinaus ergibt sich

$$\text{Enc}(z_1 z_2, r) := \text{Enc}(z_2, r)^{z_1} \bmod P^2,$$

für alle  $z_1, z_2 \in \{0, \dots, P - 1\}$  mit  $z_1 z_2 \in \{0, \dots, P - 1\}$

- Diese Eigenschaft erlaubt **Multiplikationen mit einem verschlüsselten Faktor** und entspricht der Operation  $\mu(z_1, \text{Enc}(z_2))$