Karlsruher Institut für Technologie

Sicherheit in drahtlosen Sensornetzen

Institut für Telematik, Fakultät für Informatik

Frohes Weihnachtsfest!!!

Drahtlose Sensor-Aktor-Netze

- Keine zentrale Infrastruktur, **dezentrales** Vorgehen
  - Keine zentralen Entitäten
    - Zumindest oft nicht ständig erreichbar
- **Selbstorganisierend**
  - Eingeschränkte Nutzerinteraktion (Sysadmin)
  - Eingeschränkte Wartungsmöglichkeiten
  - Systeme nach Ausbringung oft schwer zugänglich
  - Oft verbunden mit hoher Dynamik
    - Mobilität / „Schlafen“ ziehen Topologieänderungen nach sich
- **Unzuverlässiger Kommunikationskanal**
  - Drahtloses Medium stärker fehlerbehaftet als drahtgebundenes
- **Limitierte Ressourcen**
  - Rechenleistung, Energie-, Speicher- und Kommunikationskapazität
- **Unsicher**
  - Sensorknoten können beschädigt/entfernt/hinzugefügt werden
  - Abhören drahtloser Kommunikation
  - Klassische kryptographische Verfahren stoßen in WSANs wegen limitierten Ressourcen an ihre Grenzen

Trade-Off Energie-Sicherheit

Fragestellungen im Kontext Sicherheit

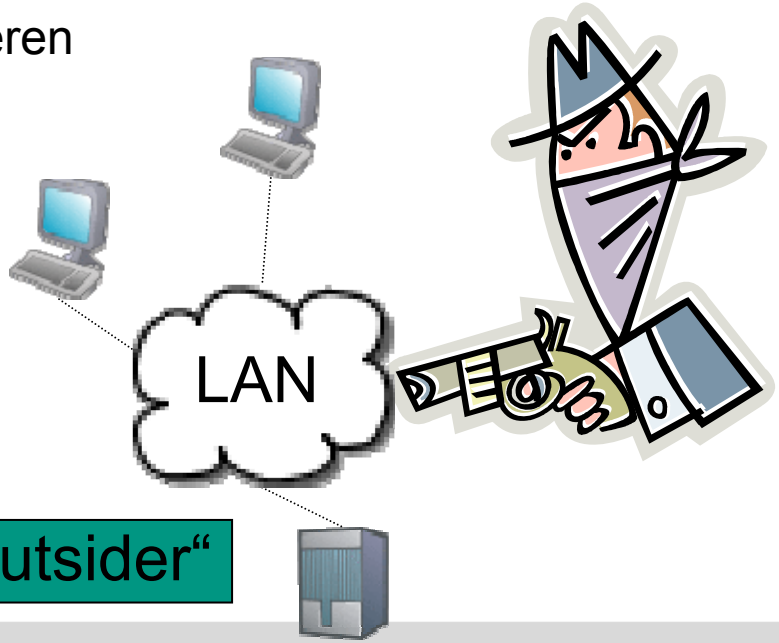
- Problem: Sicherheitsanforderung und Ressourcenknappheit **gegensätzliche** Beschränkung
  - Idee: Trade-Off Energie – Sicherheit
    - Probabilistische Metriken statt absoluter Größen
  - Wie können Trade-Offs parametrisiert werden?
  - Welcher Grad an Sicherheit ist erzielbar?

- Problem: **Energieabschätzung**
  - Welche Sicherheitsmaßnahme kostet wie viel Energie?
  - Genauigkeit von Simulationen oft unzureichend

- Problem: Sicherheit in Sensornetzen ist „anders“
  - Wie müssen sichere Anwendungen für Sensornetze entworfen werden?
  - Beispiel: Schlüsselverteilung ohne Infrastruktur und Vertrauensanker
  - → Auf Sensornetze **angepasster Sicherheitsprozess**

Angepasster Sicherheitsprozess: Angreifermodell

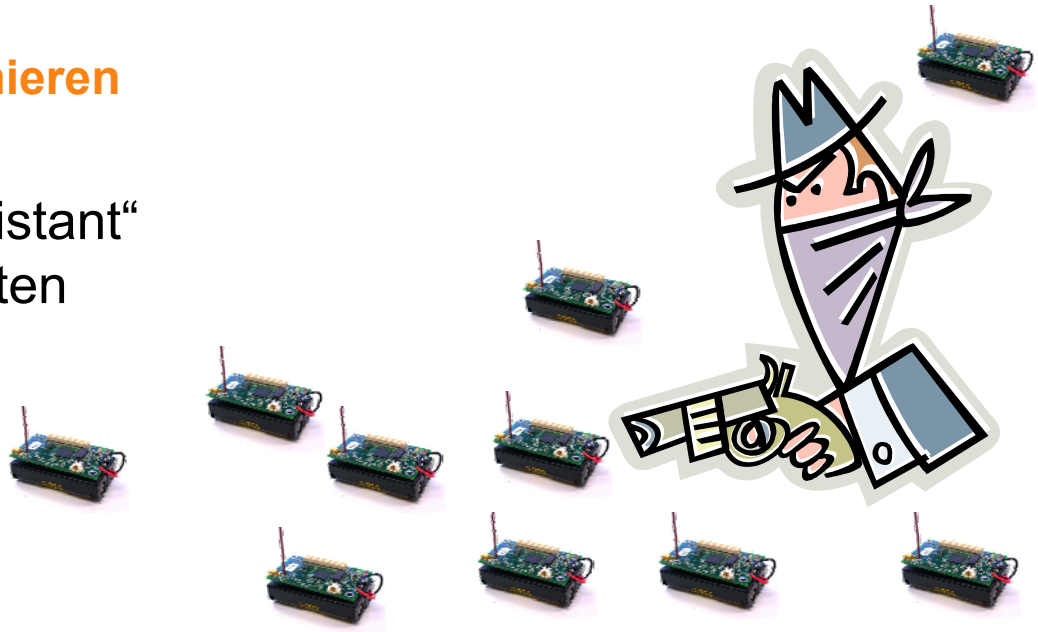
- „Klassisches“ Angreifermodell: In klassischen Protokollen häufig nicht explizit genannt, z.B. TLS (RFC 4346) oder IPSec (RFC 4301, 4303).
  - Implizit geht man häufig von einem Angreifer aus, den man „**Dolev-Yao**“ Angreifer nennt
  - Angreifer ist omnipräsent im Netz, kann sämtliche Kommunikation abhören
  - Kann eigene Dateneinheiten erzeugen und versenden
  - Kann fremde Dateneinheiten modifizieren
  - Kann allerdings nicht entschlüsseln oder verschlüsseln, ohne den Schlüssel zu kennen



Angreifer = „Outsider“

Was ist anders in Sensornetzen?

- Systeme befinden sich häufig an öffentlichen oder anderen leicht zugänglichen Orten
  - Angreifer kann physisch auf Systeme zugreifen
    - Kann Systeme evtl. einfach „klauen“ oder physisch zerstören
  - Manch anderes wird auch einfacher
    - **Speicher auslesen**
    - Komplett **re-programmieren**
    - **Korrumpieren**
  - Sogenannte „tamper-resistant“ Hardware für Sensorknoten zu teuer



Was ist anders in Sensornetzen?

- Andere Möglichkeit: Viren und Würmer?
  - Die im Internet häufigste Form der Korrumpierung
  - Auch für Sensoren denkbar: Angreifer findet Implementierungsfehler im Netzwerk-Stack („Buffer-Overflow“)
  - Damit kann Angreifer Systeme fernsteuern
- Angreifer korrumpiert eine Menge von Systemen, die danach zusammenarbeiten, „**byzantinische**“ Systeme

- Problem in Sensornetzen
  - Viele dieser Angriffe lassen sich nicht durch klassische IT-Security verhindern.

→ Herausforderung für Sicherheitsprotokolle im Sensornetz

- Erbringe einen Dienst sicher in Gegenwart von **korrumpierten** („bösen“) Systemen, z.B. sicher in Gegenwart von β=10% korrumpierten Systemen

Angreifer = „Insider“

Überblick über die weitere Agenda

- **Andere Wege: Probabilistische Sicherheit**
- Exkurs: Messung des Energieverbrauchs mit SANDbed
- Sicherheitsprozess am Praxisbeispiel FleGSens

1. Andere Wege: Probabilistische Sicherheit

- Wie viele Systeme korrumpiert Angreifer?
- Annahme: Netz mit n Systemen
  - Klar: Korrumpiert der Angreifer **n-1** oder **n** Systeme, gibt es keine sinnvolle Definition von Sicherheit mehr (Benutzer sollte andere Probleme zuerst lösen…)
  - Auch klar: Je mehr korrumpierte Systeme im Netz, desto schwieriger wird Sicherheit
  - Angreifer wird in der Realität nicht einfach alle Systeme korrumpieren können
    - Korrumpieren von Systemen „**kostet**“ den Angreifer etwas
      - Z.B. Zeit, Geld in Form der notwendigen Hardware, usw.
    - Angreifer verfügt nur über begrenzte Ressourcen oder will nur begrenzte Ressourcen für seinen Angriff ausgeben
  - Häufig geht man davon aus, dass der Angreifer **n‘<n**, d.h. maximal einen Prozentsatz **β%** zufällig korrumpiert.

Probabilistische Sicherheit

- „Absolute“ Sicherheit unter diesem Angreifermodell zu teuer

- Beispiel: authentische Aggregation von Daten im Netz
  - Aggregation soll Energie sparen
  - Verhindert Prüfung der Authentizität der Daten
  - Bei 100% Authentizität, keine Aggregation möglich
- Ansatz: **probabilistische** Authentizität
  - Systeme die Aggregieren werden mit bestimmter Wahrscheinlichkeit **p** durch andere Sensorknoten „überprüft“
- Resultat
  - Aggregate sind mit bestimmter Wahrscheinlichkeit **P(p, β)** < 100% authentisch
  - Realisiert **Trade-Off** zwischen Energie und Authentizität

Wilke, Blaß, Freiling, Zitterbart
A Framework for Probabilistic, Authentic Aggregation in WSNs.
Praxis der Informationsverarbeitung und Kommunikation (PIK), April 2009.

Wie viel Sicherheit ist möglich?

- **Energieverbrauch** bestimmt realisierbares Maß an Sicherheit
  - Sicherheitsmaßnahmen erhöhen Kommunikationsvolumen, Rechenaufwand, …
  - Energieverbrauch über Simulationen nur ungenau abschätzbar
    - Funkmedium schwer zu simulieren
    - Energiemodell von Sensorknoten ungenau, teilweise plattformabhängig

Genaue, plattformunabhängige Energiemessung notwendig

Überblick über die weitere Agenda

- Andere Wege: Probabilistische Sicherheit
- **Exkurs: Messung des Energieverbrauchs mit SANDbed**
- Sicherheitsprozess am Praxisbeispiel FleGSens

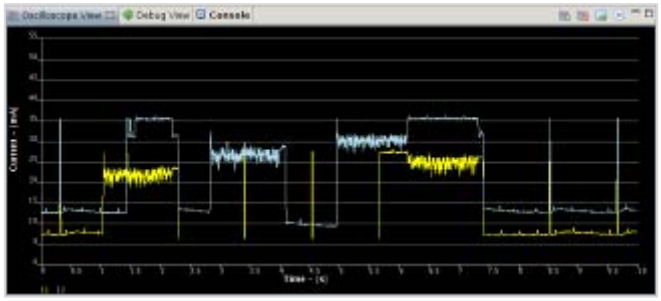
2. Exkurs: Messung des Energieverbrauchs

- Wünschenswert
  - Realisitische Energiemessung
  - Sensornetz-Applikation ohne Zusatzmodifikationen direkt auf der Hardware vermessen
    - Basis für die Entwicklung detaillierterer Energiemodelle

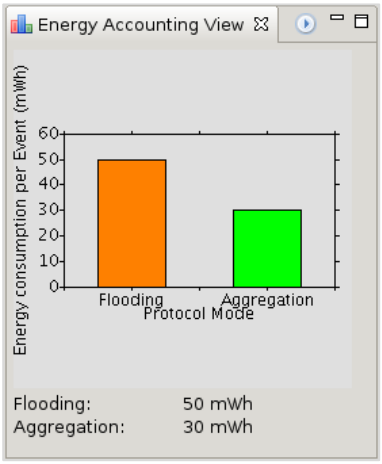
  → Sensornetz-Testbett **SANDbed**
    - Sensor-Actuator-Network Development Testbed

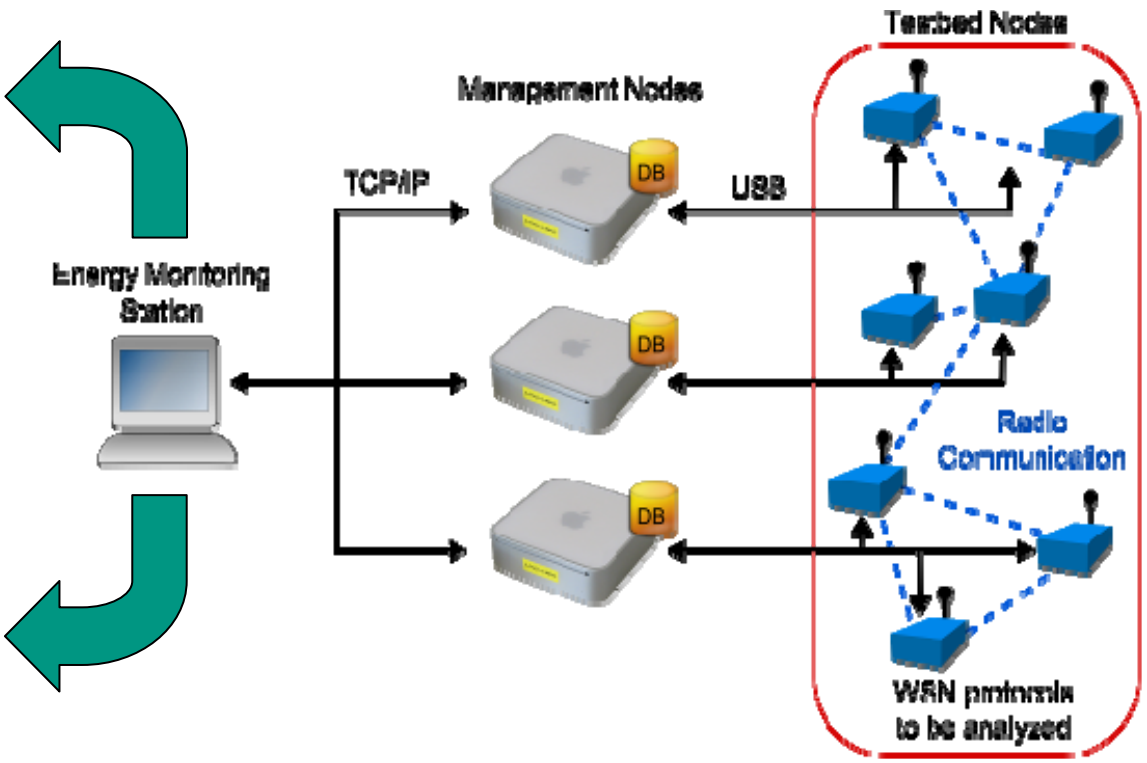
SANDbed - Gesamtarchitektur

- Detaillierte Energieverbrauchsanalyse einzelner Systeme



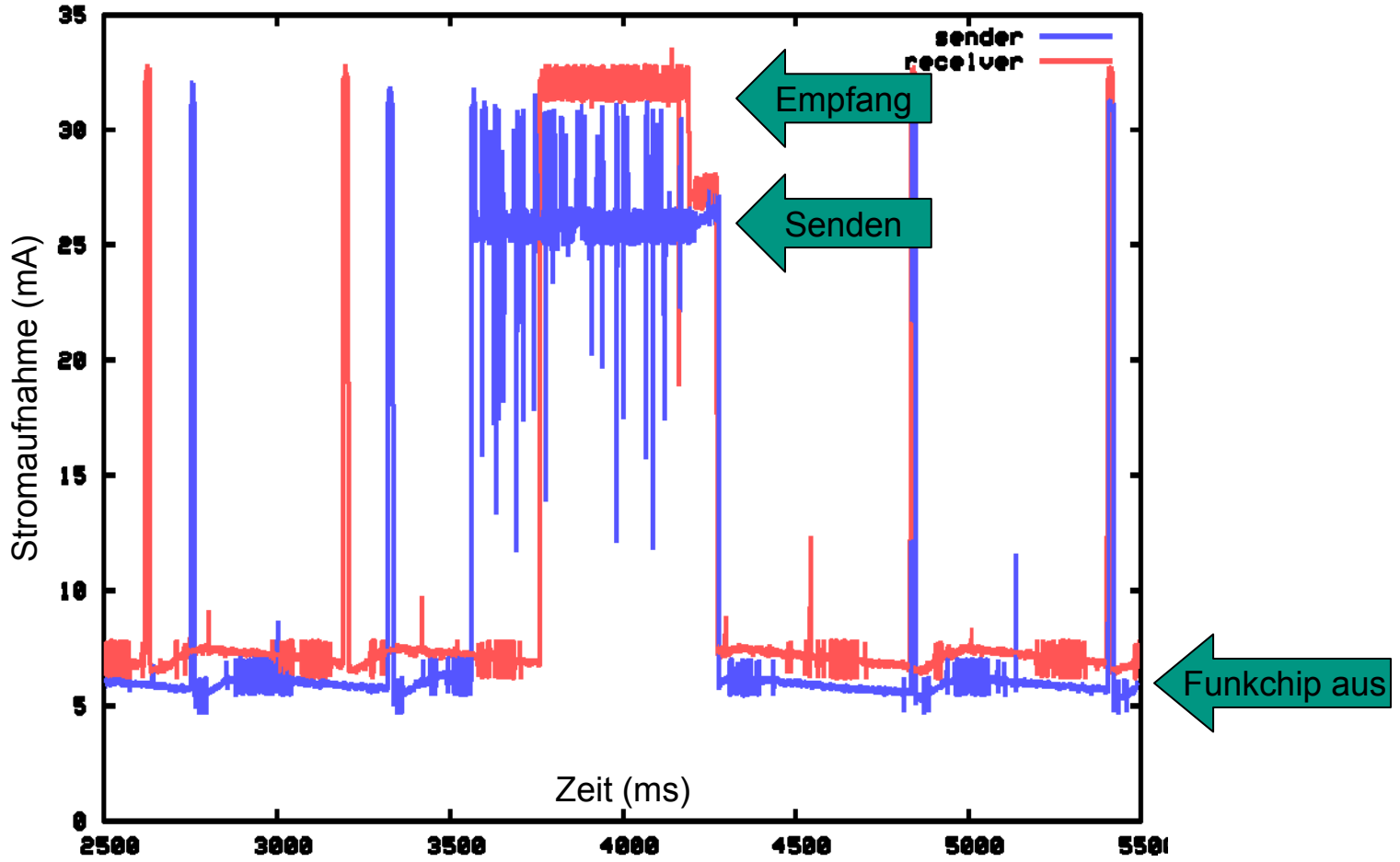
- Gesamtenergieverbrauch des Netzes





Energiemessung „Demo“

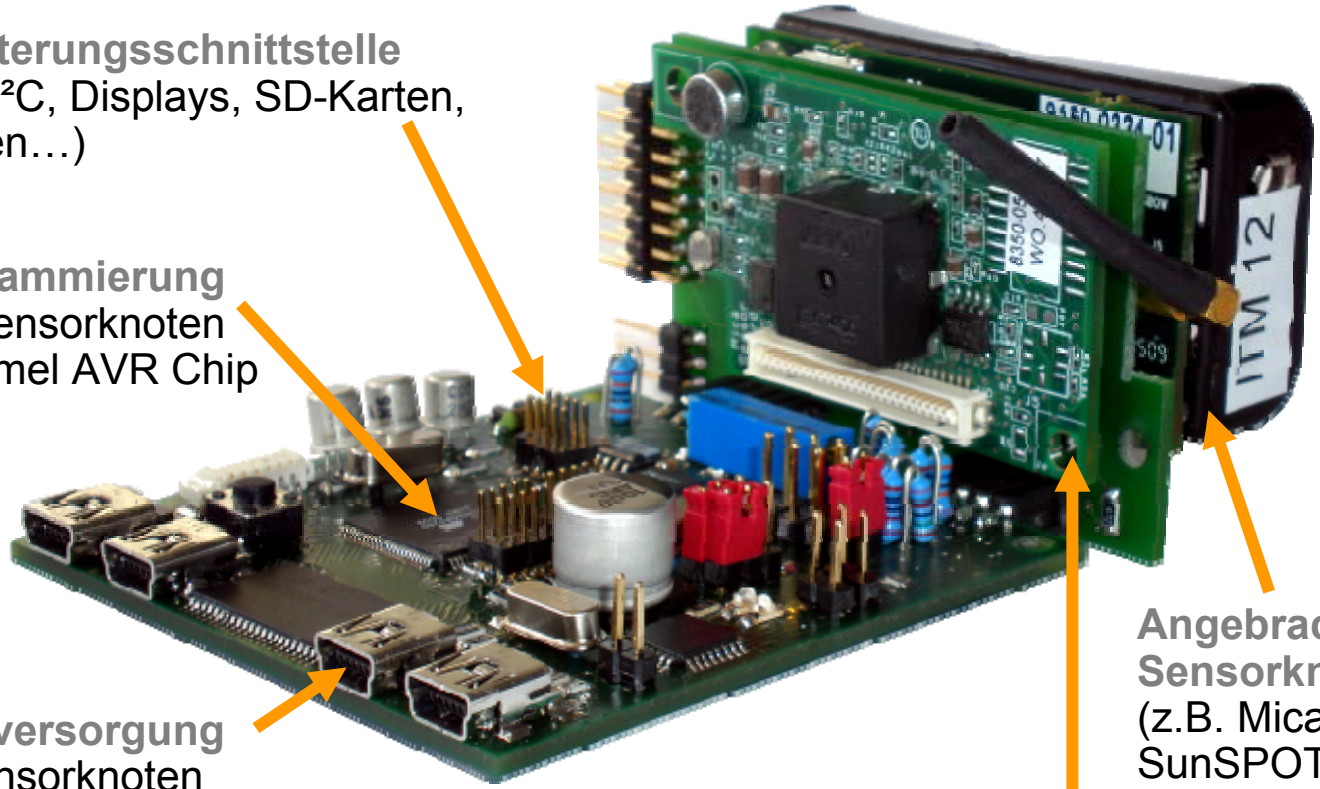
- Beispiel: Paketversand mit B-MAC auf MicaZ (Sampling 1kHz)



SANDbed – Sensor Network Management Device

Erweiterungsschnittstelle
(SPI, I²C, Displays, SD-Karten, Aktoren…)

Programmierung
von Sensorknoten mit Atmel AVR Chip



Stromversorgung
der Sensorknoten über USB oder per Batterie

Angebrachter Sensorknoten
(z.B. MicaZ, SunSPOT, …)

Sensorboard

Hergenröder, Horneber, Meier, Armbruster, Zitterbart
Distributed Energy Measurements in Wireless Sensor Networks,
ACM SenSys,Berkeley, USA, Nov 2009.

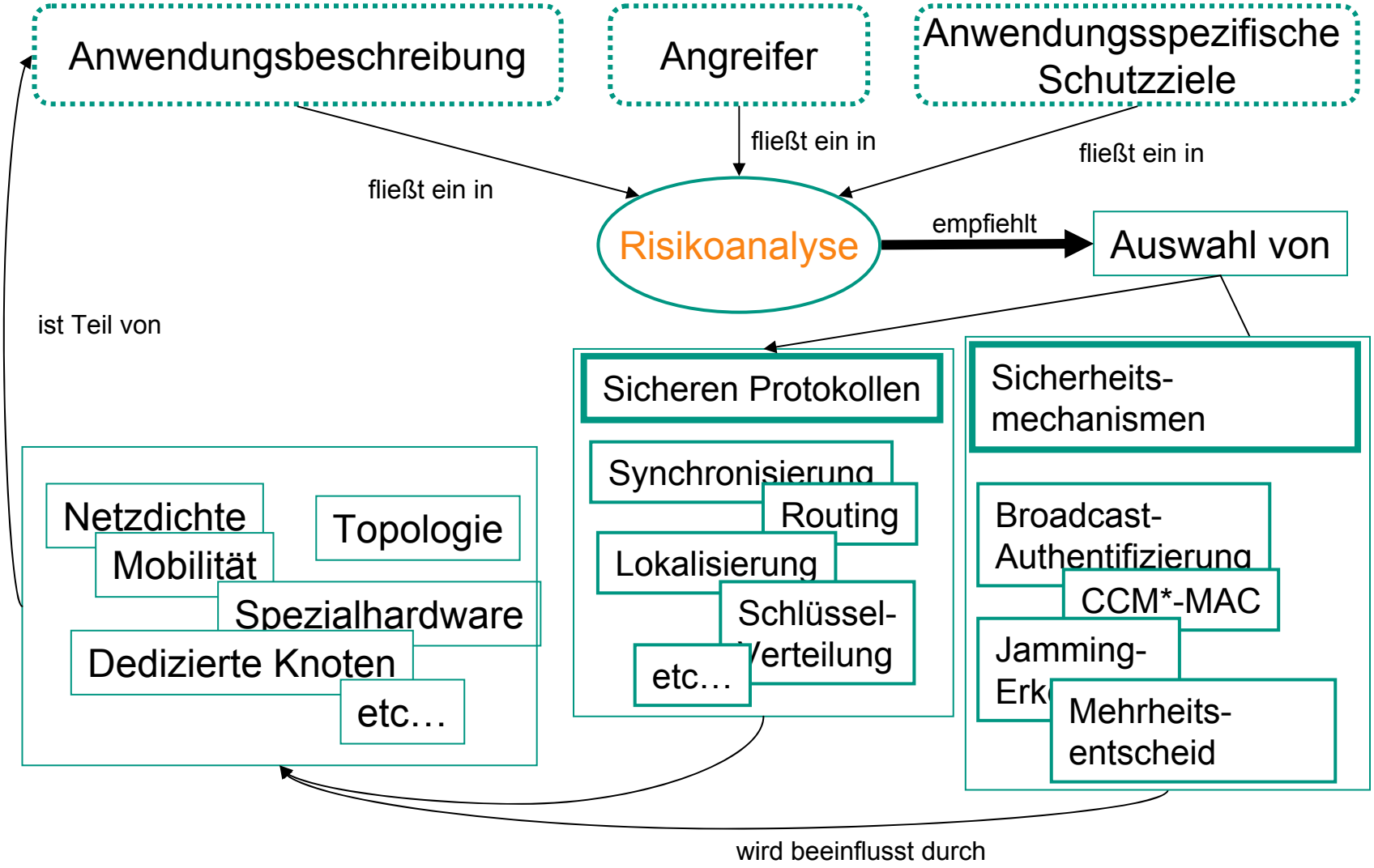
Überblick über die weitere Agenda

- Andere Wege: Probabilistische Sicherheit
- Exkurs: Messung des Energieverbrauchs mit SANDbed
- **Sicherheitsprozess am Praxisbeispiel FleGSens**

3. Sicherheitsprozess in Sensornetzen

- Was brauchen wir um eine Anwendung abzusichern ?
  - Angreifermodell
    - Intern, extern, aktiv oder passiv ?
    - Anzahl der korrumpierbaren Sensorknoten ?
  - Anwendungsspezifische Schutzziele
    - Welche Ziele hat ein Angreifer, der Schaden anrichten will ?
  - Anwendungsbeschreibung
    - Welche Komponenten sind abzusichern ?
  - Risikoanalyse
    - Wie wahrscheinlich ist ein Angriff ?
    - Wieviel Schaden richtet ein Angriff an ?
  - Sicherheitsmaßnahmen und „sichere Protokolle“
    - Wieviel Overhead kostet mich ein sicheres Protokoll ?

Sicherheitsprozess in Sensornetzen

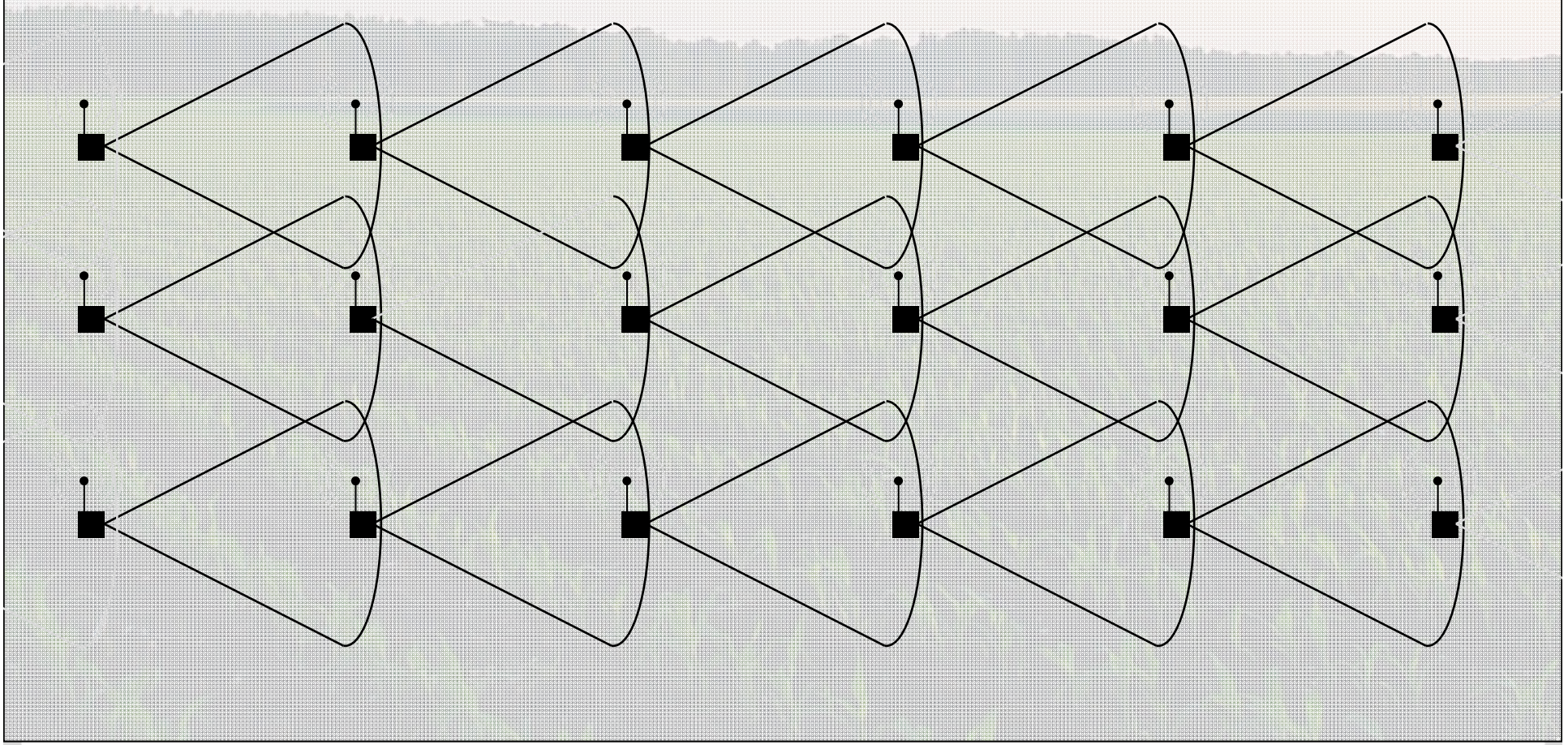


FleGSens - Anwendungsbeschreibung

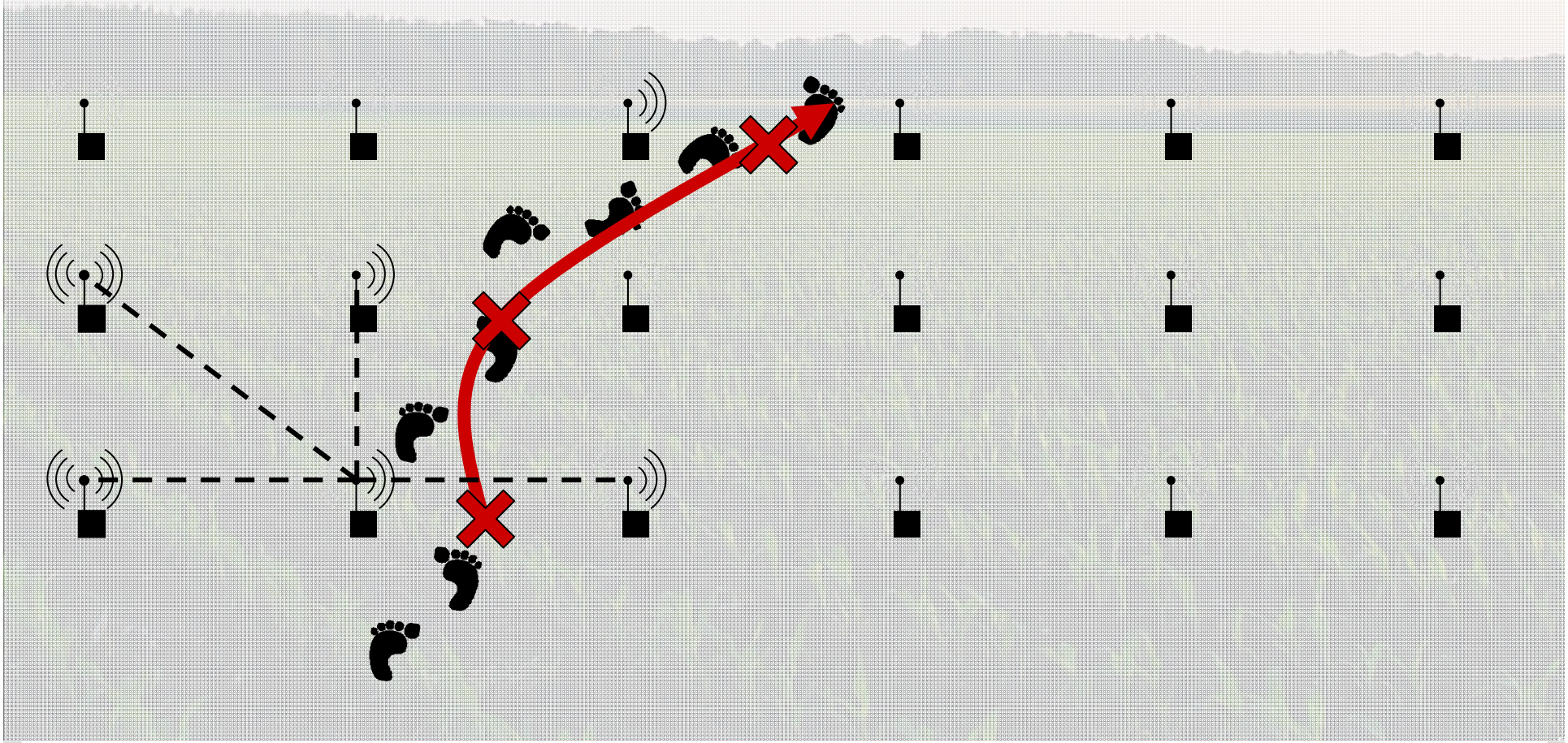
- Grenz und Liegenschaftsüberwachung durch drahtlose Sensornetze
  - Erkennung von Übertritten an einer grünen Grenze
  - Fokus auf Informationssicherheit, nicht Sensorik
  - 200 Sensorknoten (10% Knotenausfallrate, 5% korrumpierte Knoten)
  - Simulative Betrachtung mit bis zu 2000 Sensorknoten

- Funktionale Anforderungen
  - Ortung von Eindringlingen auf 10m, Meldung innerhalb von 5sec
  - Lebensdauer: 7 Tage
  - Angriffe mit IT-Mitteln zur Störung des Betriebes

FleGSens - Anwendungsbeschreibung



FleGSens - Anwendungsbeschreibung



FleGSens - Angreifermodell

- Dolev-Yao Modell
  - „Man-in-the-middle Angreifer“
  - Abhören jeglicher Kommunikation im Netz
  - Erzeugen, Modifizieren von Nachrichten und Maskerade
  - Angreifer kann kryptografische Algorithmen nicht effizient berechnen
  - Angreifer kann Schlüsselmaterial nicht erraten
- Erweiterungen in FleGSens
  - Hardware-Manipulation
    - Physischer Zugriff auf die Sensorknoten
    - Auslesen und Neu-Programmierung der Sensorknoten
    - Sensorknoten nicht „tamper-resistant“
  - Korrumpierung durch Würmer
    - Korrumpierung durch „Over-the-Air Programming“
  - Angreifer kennt alle Protokollabläufe

FleGSens – Anwendungsspezifische Angriffs- und Schutzziele

- Verhindern von Alarmen
  - Angriff auf die Weiterleitung von Nachrichten
  - Verfälschen von Zeitstempeln und Uhren
  - Zerstörung von Knoten
- Verzögern von Alarmen
- Verfälschen von Alarmen
  - Lokalisationsinformation oder Zeitinformation
- Erzeugung von Fehlalarmen
- Denial of Service
  - Angriffe basierend auf Fluten von Nachrichten
- Partitionierung des Netzes
- Verkehrsanalyse
  - Ziel: Identifikation „wichtiger“ Knoten

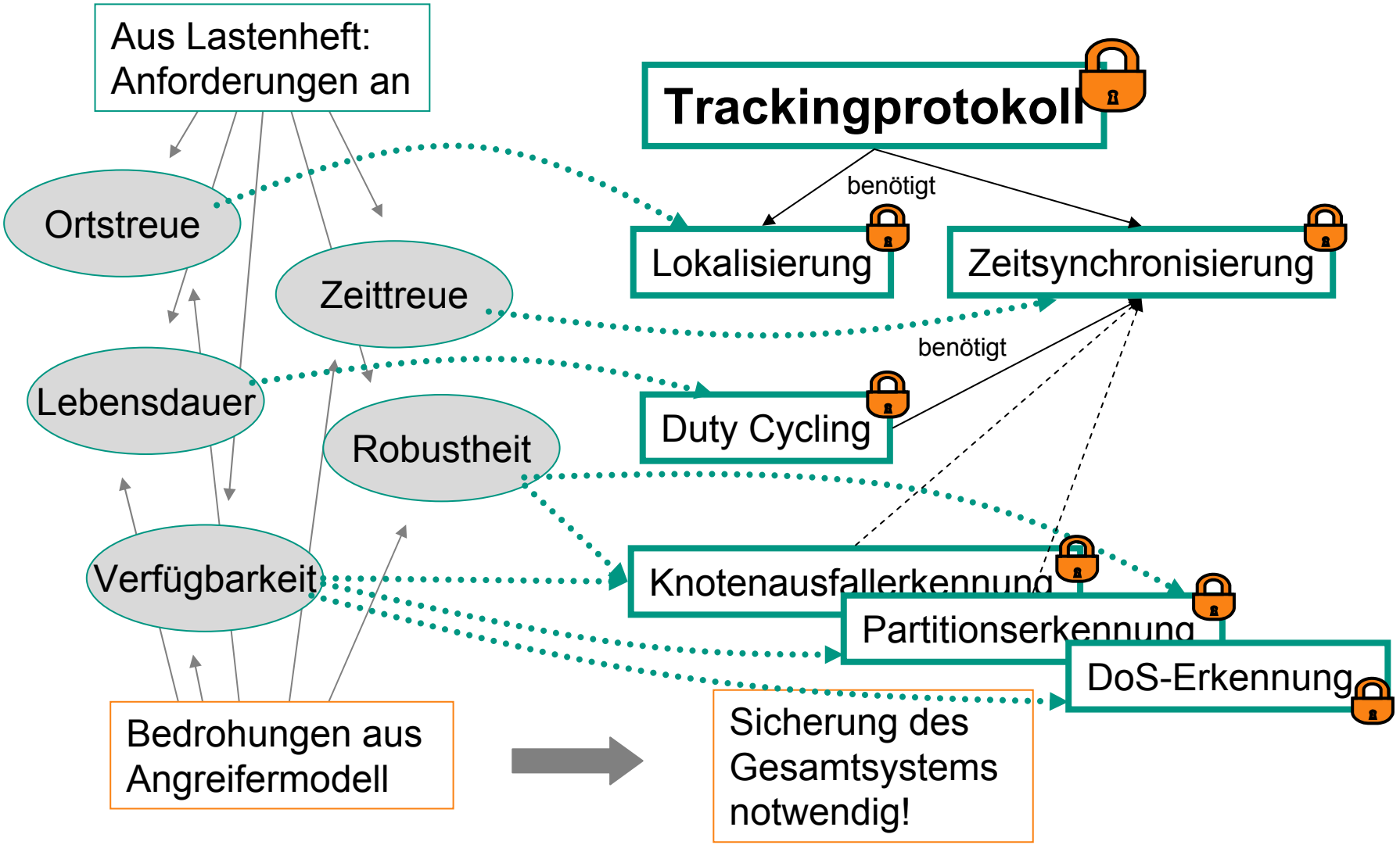
FleGSens - Aufwands- und Schadensanalyse

Angriffsziel	Angriff	Aufwand	Schaden	Gegenmaßnahme
Verhindern von Alarmen	Black-Hole-Attacke	Hoch	Hoch	Fluten
	Verfälschen von Zeitstempeln	Gering	Hoch	Message Authentication Codes
Verzögern von Alarmen	Nachrichten aufhalten	Hoch	Hoch	Fluten
Verfälschen von Alarmen	Ortsinformation verfälschen	Gering	Hoch	Message Authentication Codes
	Zeitstempel verfälschen	Gering	Hoch	Message Authentication Codes
Erzeugen von Fehlalarmen	Event-Nachrichten verfälschen	Mittel	Mittel	Sicheres Trackingprotokoll
	Korrumpierung von Knoten	Hoch	Mittel	Aggregation / Mehrheitsentscheid
Denial of Service	Flooding	Hoch	Hoch	DoS-Erkennung (keine Verhinderung)
Partitionierung des Netzes	Korrumpierung von Knoten	Hoch	Hoch	Keine möglich Partitionserkennung
	Injektion falscher Zeit-Informationen	Gering	Hoch	Sichere Zeitsynchronisierung
Verkehrsanalyse	Abhören des Verkehrs	Gering	Gering	Verschlüsselung

FleGSens - Sicherheitsmechanismen und -protokolle

- Einsatz von symmetrischer Kryptografie
  - Asymmetrische Kryptografie „zu teuer“
    - Symmetrisches Verschlüsseln (AES/128Bit):<300µs
    - Asymmetrisches Verschlüsseln (ECC/160Bit): ca. 3s
  - Nutzung des AES-Coprozessors auf den Sensorknoten
  - Paarweise symmetrische Schlüssel für Kommunikationspartner
    - Schlüsselaustauschprotokoll notwendig (HARPS)
- Einsatz von Broadcast-Authentifikationsprotokollen
  - Authentifikation von Basisstation und Senken (µTESLA)
- Nutzung von Message Authentication Codes (MACs)
  - AES-Coprozessor liefert MAC in <300µs (CCM*-MAC)

Entwurf einer sicheren Architektur am Beispiel FleGSens



FleGSens Impressionen

Rothenpieler, Krüger, Buschmann, Pfisterer, Fischer, Dudek, Haas, Zitterbart
FleGSens – A Wireless Sensor Network for Border Surveillance.
ACM Sensys09 Proceedings.

Resümée

- Absolute Sicherheit in Sensornetzen schwer erreichbar
  - Angreifer kann Knoten korrumpieren
  - Probabilistische Verfahren aber oft ausreichend
- Sicherheit in Sensornetzen im Spannungsfeld Energie – Sicherheit zu betrachten
  - Hierfür genaue Energiemessung notwendig
- Spezieller Sicherheitsprozess in Sensornetzen notwendig
  - Für jedes einzelne Protokoll ist eine Sicherheitsanalyse durchzuführen
  - Jedes Protokoll muss einzeln vor Angriffen geschützt werden
    - Insider-Angriffe hier die größte Gefahr
  - Sicherheitsüberlegungen müssen zeitgleich mit dem Entwurf der Anwendung durchgeführt werden

Karlsruher Institut für Technologie

Institut für Telematik
Prof. Dr. Martina Zitterbart

Vielen Dank für Ihre Aufmerksamkeit

Universität Karlsruhe (TH)
Research University · founded 1825

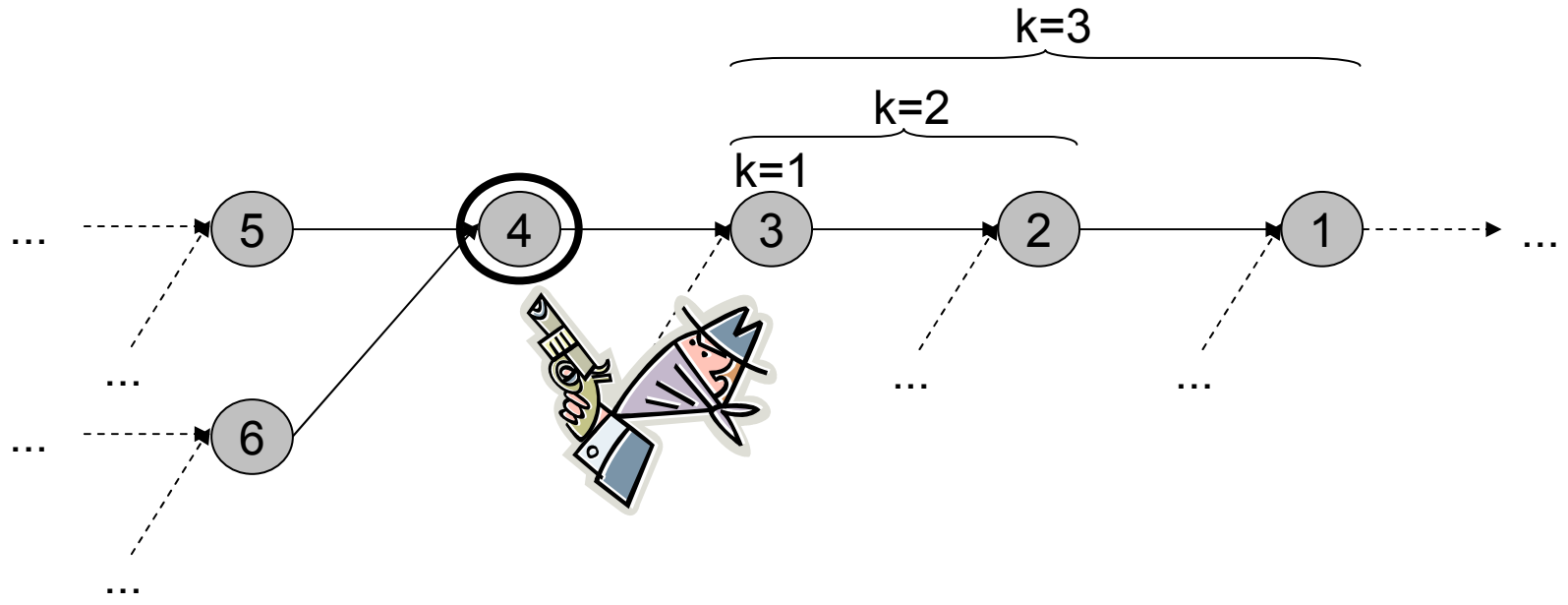
Prof. Dr.
Martina Zitterbart

Zirkel 2 · Geb. 20.20 · 76128 Karlsruhe
Tel.: +49 721 608 – 64 00 · Fax: - 67 89
E-Mail: zit@tm.uka.de

Backup

Probabilistische Sicherheit – ESAWN

- Extended Secure Aggregation for WSNs
- Annahme: Anteil **β** Knoten korrumpiert
  - Pro Aggregationsüberprüfung maximal **k** Knoten korrumpiert
- Beispiel: Aggregationsbaum, Einfluss von k



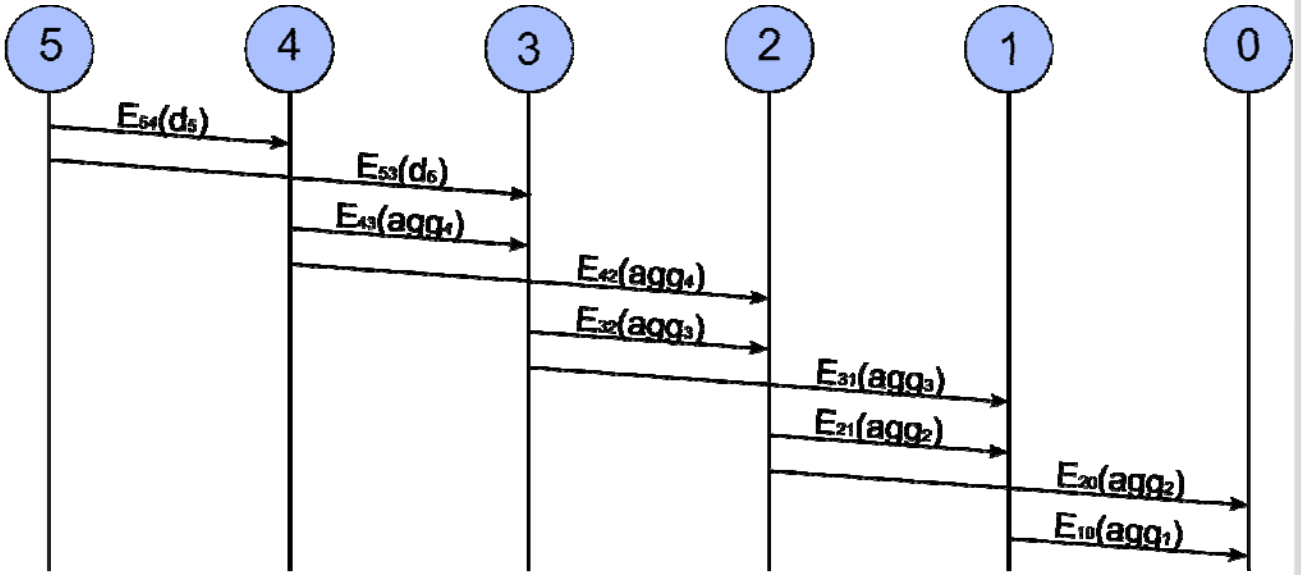
ESAWN-1

- ESAWN-1 (Basisvariante)
  - Ablauf für jeden Aggregationsknoten *4*
    - Knoten *4* wird durch k **Vorgänger** *3*, *2*, *1* überprüft
    - Knoten *5* und *6* senden Daten an *k* Vorgänger — **Aufheben der Aggregation**
    - Vorgänger **überprüfen** Aggregat
      - Fehlerhaftes Aggregat entdeckt → Alarm
  - Energieverbrauch verringern: **überprüfe nur mit *p%***

Senke („Vertrauensanker“)

→ **Trade-Off** durch Wahl von **k** und **p**

→ **P(p, k)**



ESAWN-Protokollvarianten

- ESAWN-1
  - Einfachste Möglichkeit, Verwendung von Zeugen zur **Erkennung** korrumpierter Aggregate, Aggregation wird abgebrochen
- ESAWN-2
  - Einbeziehung von weiteren Zeugen um Mehrheitsentscheid zu ermöglichen (**Behebung** korrumpierter Aggregate)
- ESAWN-NR
  - Protokollierung verschlüsselter Kommunikation auf den beteiligten Systemen, Aufdeckung von Schlüsseln zur **Identifizierung** korrumpierter Systeme

→ **Trade-Off** durch Wahl der Protokollvariante