



Seminarvortrag:

Sicherheit in Sensornetzen

Danat Pomeranets

Betreuer: Harald Vogt

Übersicht

- Sicherheit in Sensornetzen
- Denial Of Service
- SPINS
 - SNEP
 - μ TESLA
- Resümee

Warum Sicherheit?

- Feindliche Umgebungen
 - Schlachtfeldüberwachung
- Sicherheitskritische Anwendungen
 - Sensoren in Reaktoren, Feuermelder, ...
- Privatsphäre / Datenschutz
 - Home Healthcare

Mögliche Angriffe

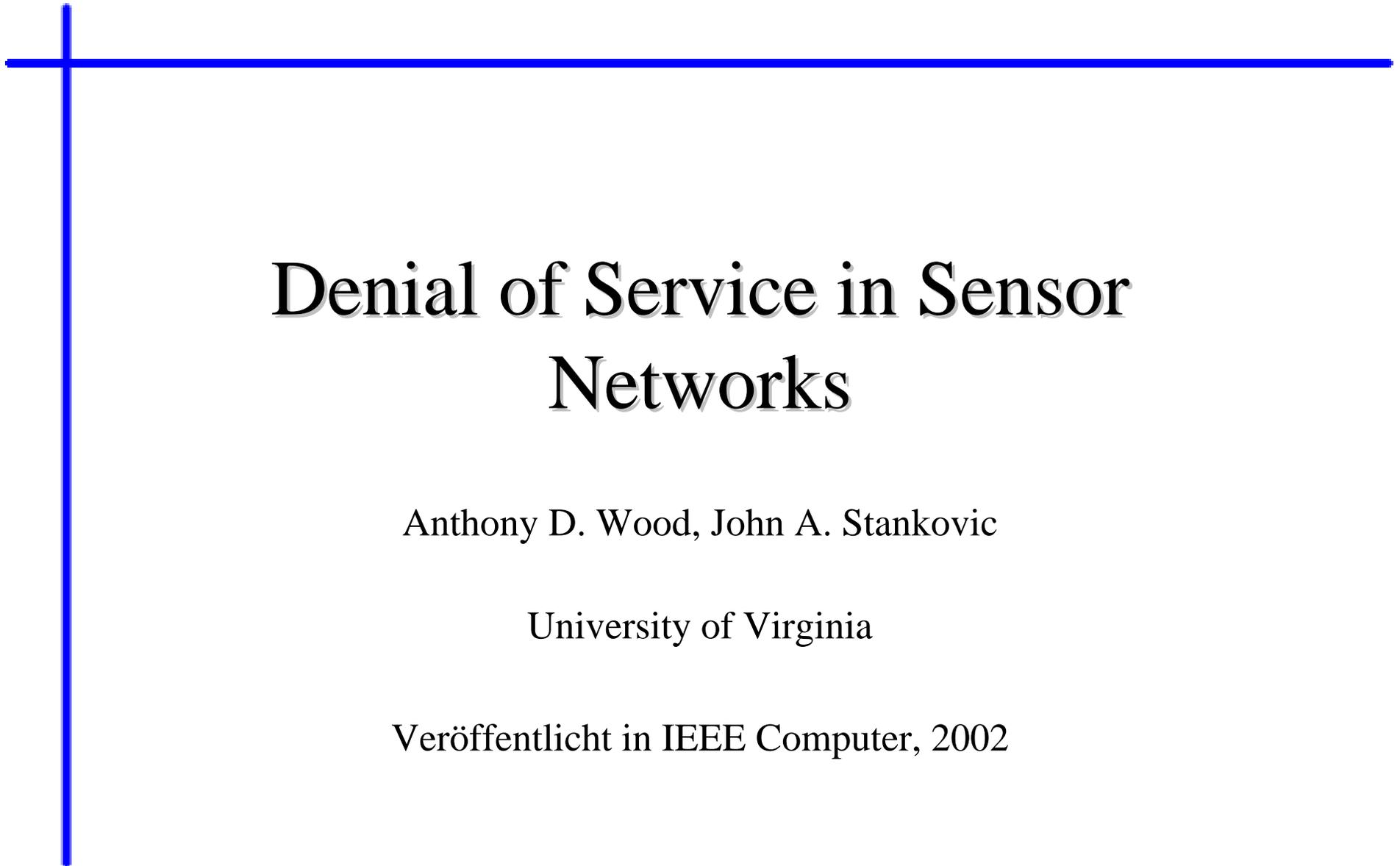
- Abhören (*eavesdropping*)
- Verkehrsanalyse
- Nachrichten
 - eigene erzeugen (*injection*)
 - manipulieren
 - wiederholen (*replay*)
- Denial Of Service
- ...

Anforderungen / Ziele

- Vertraulichkeit (*confidentiality*)
 - *Nachrichteninhalt für Gegner nicht lesbar*
- Integrität (*integrity*)
 - *Nachrichteninhalt wurde nicht nachträglich verändert*
- Authentizität (*authencity*)
 - *Der Absender ist der echte Kommunikationspartner*
- Verfügbarkeit (*availability*)
- Aktualität (*freshness*)
 - *Schutz gegen Replay*

Übersicht

- Sicherheit in Sensornetzen
- **Denial Of Service**
- SPINS
 - SNEP
 - μ TESLA
- Resümee



Denial of Service in Sensor Networks

Anthony D. Wood, John A. Stankovic

University of Virginia

Veröffentlicht in IEEE Computer, 2002

DoS: Definition

- Jedes Ereignis, das die Funktionalität des Netzwerks erheblich beeinträchtigt
→ Verfügbarkeit !
- Verursacht durch Hardware-Ausfälle, Software-Bugs, Umgebung, ...
- Angriff: wenn absichtlich hervorgerufen
→ Auf mehreren Netzwerk-Ebenen möglich

DoS: Schichten

- Physische Schicht
 - Störung der Kommunikation (*jamming*)
 - Eingriff in die einzelnen Knoten (*tampering*)
- Sicherungsschicht (Link Layer)
 - Kollisionen
- Netzwerk- und Routing-Schicht
 - Wurmlöcher, Schwarze Löcher, Homing, ...
- Transport-Schicht
 - Fluten (*flooding*)

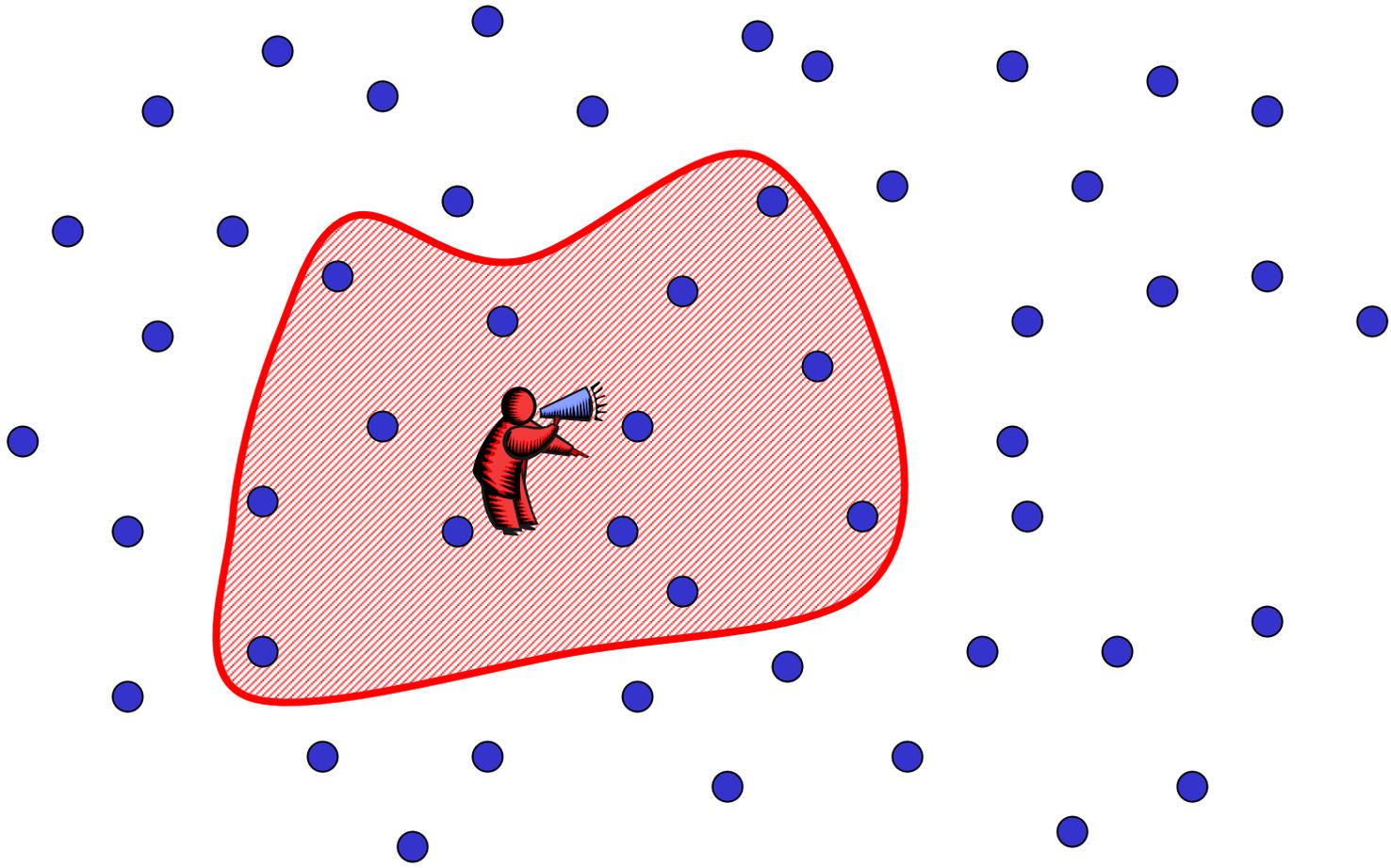
DoS: Physische Schicht

- Schutz gegen Eingriff in die einzelnen Knoten
 - Physischer Schutz (*tamper-proofing*)
 - Knoten verstecken
 - Auf Einbruchversuche in Hardware reagieren, z.B. gesamten Speicher löschen
- Interferenzen mit Störsignal
 - Einfach und effektiv
 - Können auch ohne “böse Absichten” auftreten

DoS: Jamming

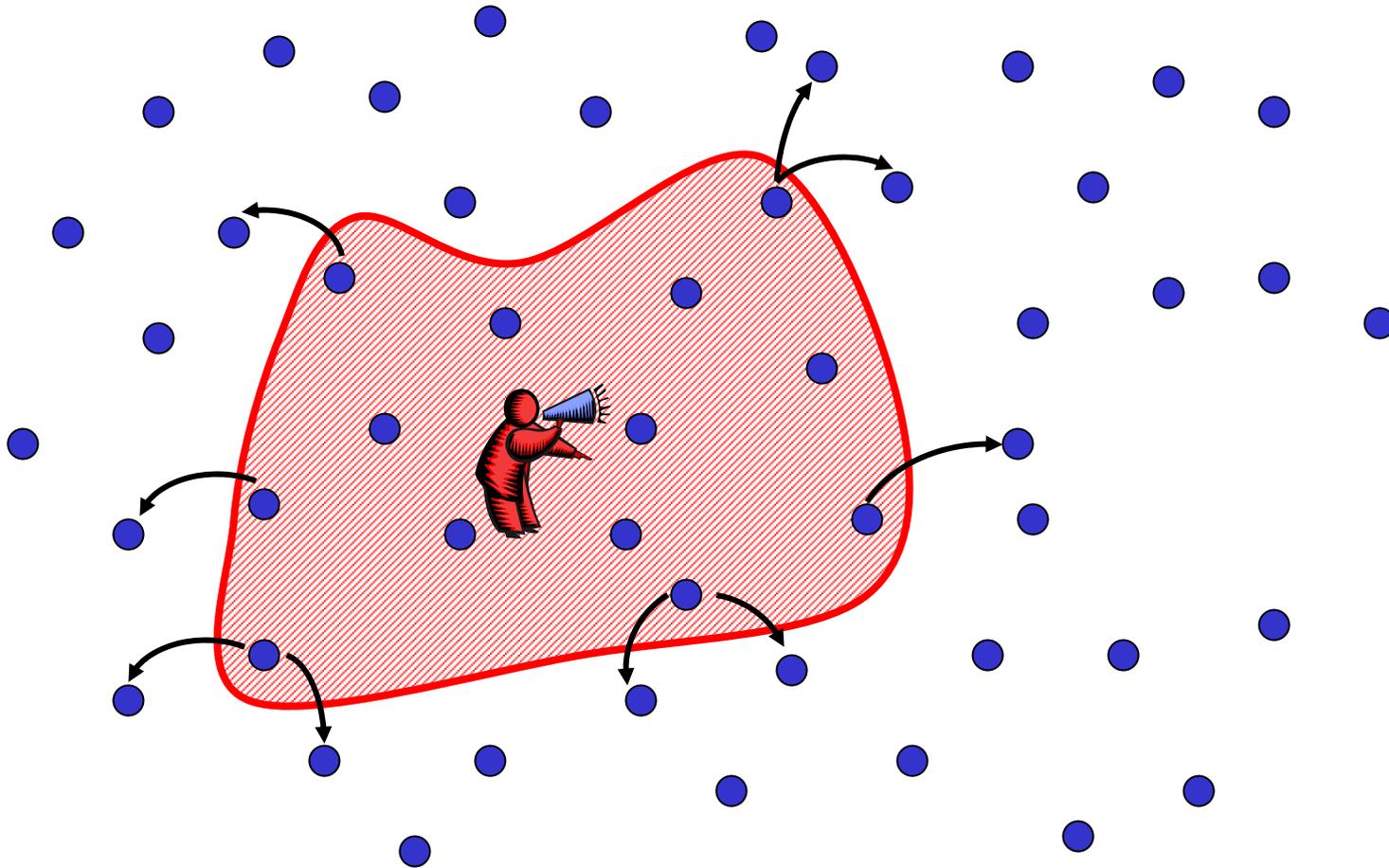
- Schutz gegen Störsignal
 - Frequenz wechseln:
 - Breitspektrum-Kommunikation
 - Frequenz-Hopping (z.B. bei Bluetooth)
 - Code-Spreading (z.B. UMTS)
 - Zu aufwendig für limitierte Sensor-Knoten
 - Im Schlaf das Ende abwarten
 - Andere Medien: z.B. optische Kommunikation
- In einem grossräumig verteilten Sensornetz ist ggf. nur ein Teilgebiet betroffen. Dann...

DoS: Jamming



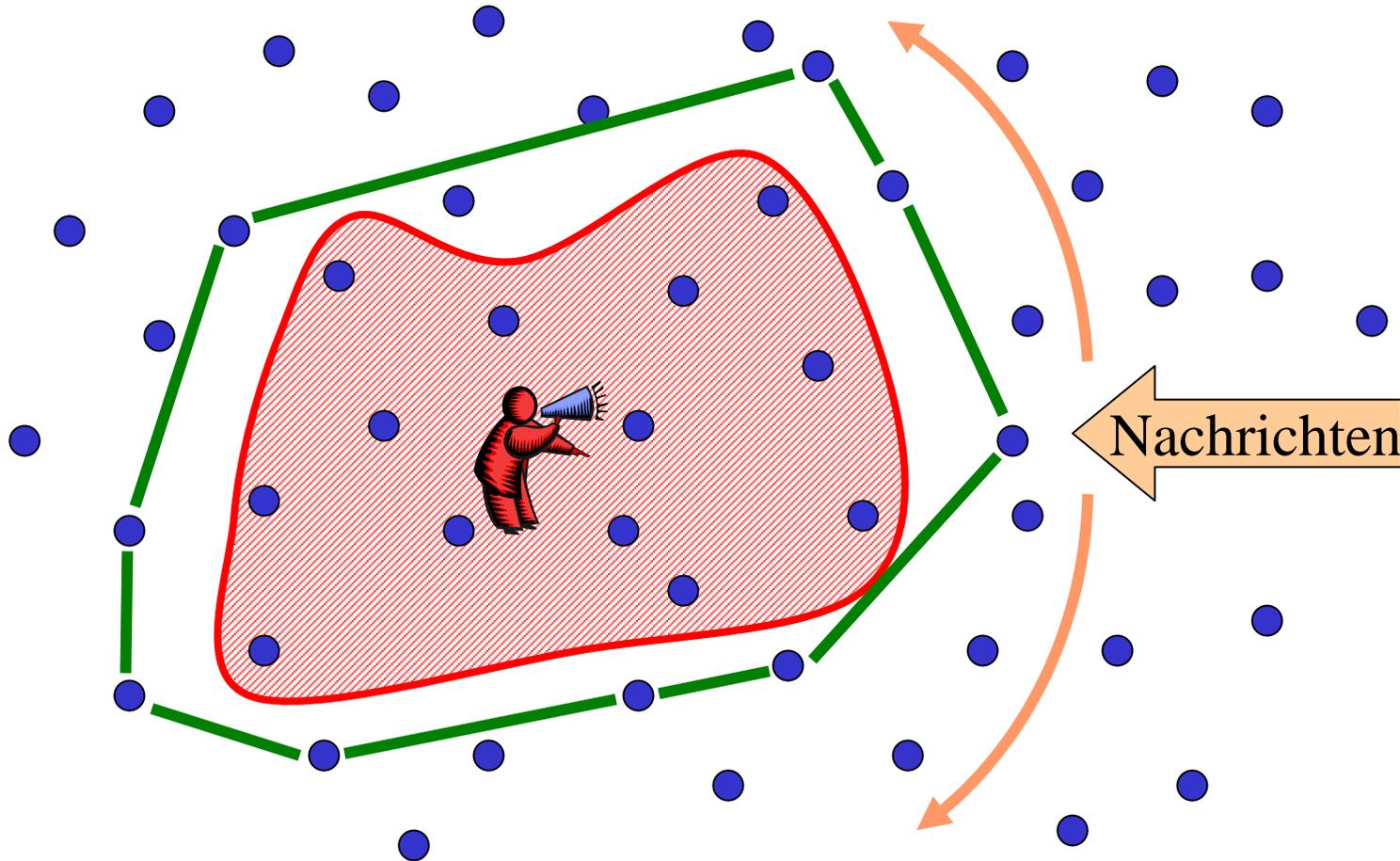
DoS: Jamming

Phase 1:



DoS: Jamming

Phase 2:



DoS: Link Layer

- Aufbrauchen der Energie durch Kollisionen
 - Einige Bits korrupt → Packet ganz neu senden
 - Kollisionen mit ACKs problematisch
- Schutz teilweise durch
 - Fehlerkorrigierende Codes
 - Random Back-Off (s. Ethernet)

DoS: Routing

Problem: jeder Knoten ist auch ein Router

Einige Beispiele für Fehlverhalten:

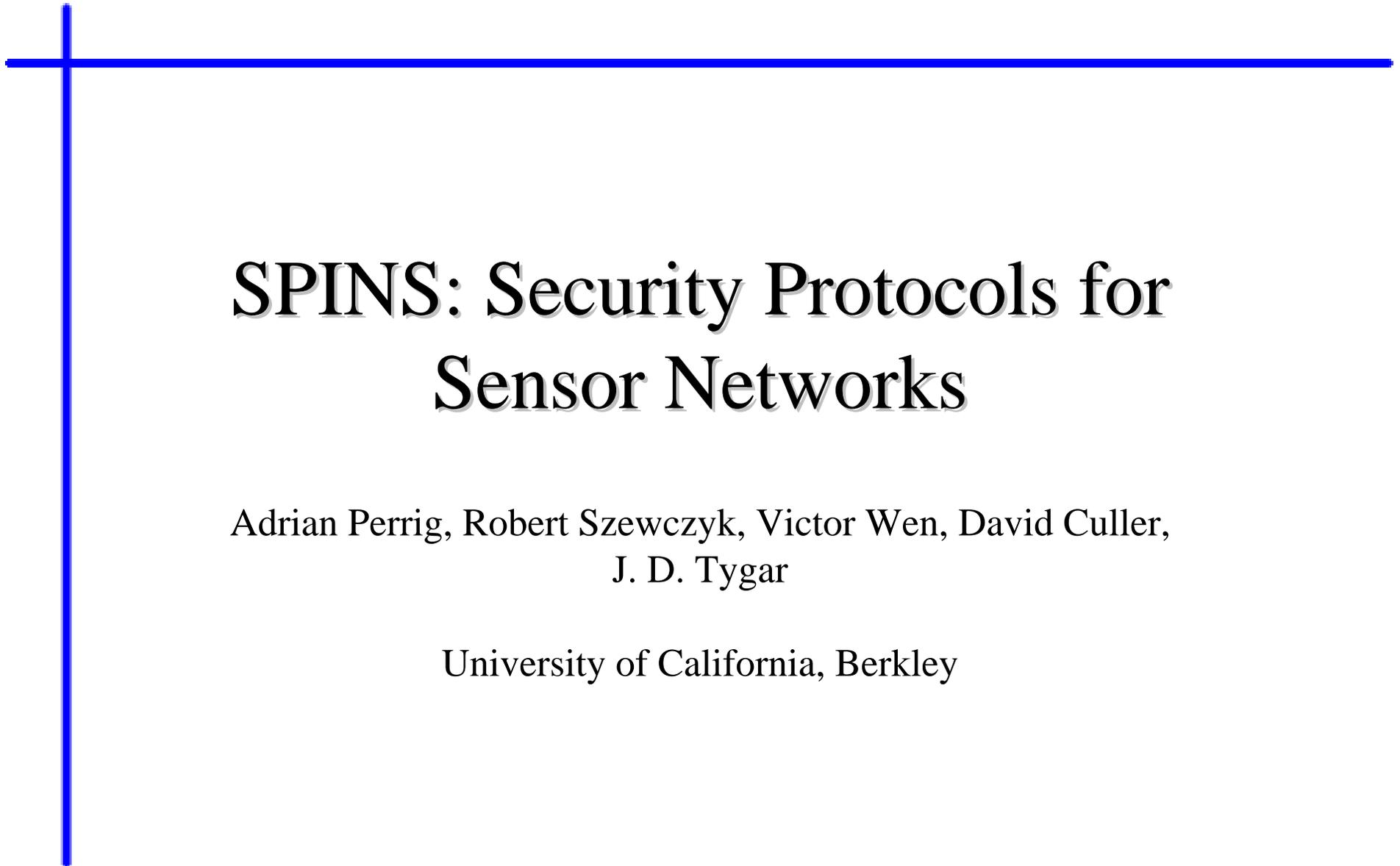
- Nachrichten verwerfen oder eigene bevorzugen
→ Mehrmals verschicken, mehrere Wege benutzen
- Nachrichten auf falschen Pfaden weiterleiten
(*Wurmloch*)
→ Knoten können Nachbarn überwachen/testen
- Alle Routing-Pfade zu sich holen
(*Schwarzes Loch*)
→ Knoten untereinander autorisieren (SNEP)

DoS: Fazit

- DoS-Attacken sind
 - schwer abzuwehren
 - oft sogar schwer festzustellen
- Grosse Anfälligkeit da jeder Knoten auch ein Router ist
- Sicherheit muss zur Designzeit und auf allen Ebenen beachtet werden
 - Prinzip: mach es dem Gegner so schwer wie möglich
→ Aufwand zum Gegner hin verschieben

Übersicht

- Sicherheit in Sensornetzen
- Denial Of Service
- **SPINS**
 - SNEP
 - μ TESLA
- Resümee



SPINS: Security Protocols for Sensor Networks

Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler,
J. D. Tygar

University of California, Berkley

Hardware – SmartDust

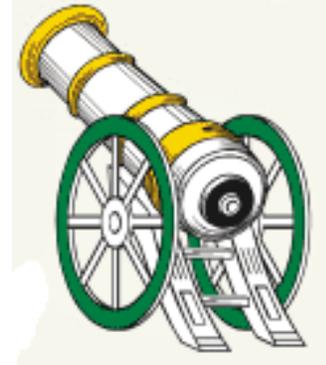
- 4 MHz, 8-bit CPU
- 8 KB Flash-Speicher
- 4,5 KB frei für Programmcode (TinyOS belegt 3,5 KB)
- 512 bytes SRAM
- 10 Kbps radio



Public-Key Kryptographie

... ist in Sensornetzen kaum einsetzbar:

- Beispiel: RSA-1024
- Signaturlänge = 128 byte, Schlüssellänge = 128 byte
- Schlüssel \leftrightarrow Speicher (*512 byte*)
- Berechnungen \leftrightarrow Energie & Rechenleistung
- Overhead pro Nachricht \leftrightarrow Energie & Bandbreite
(*typische Grösse: 30 byte*)



SPINS: Ziele

- Vertraulichkeit der Nachrichten
 - Authentizität und Integrität der Nachrichten
 - Aktualität der Nachrichten
 - Minimieren des Overheads der sicheren Kommunikation
- Minimieren des Energieverbrauchs

SPINS: Netzwerk-Modell

- Knoten:
 - Selbstorganisierend, Multihop-Routing, gemeinsame Zeit (synchrone Phasen)
- Eine Basisstation: unbeschränkte Energie
- Kommunikationsmöglichkeiten:
 - Knoten \rightarrow Basisstation (Sensorwerte)
 - Basisstation \rightarrow Knoten (Anfragen)
 - Basisstation \rightarrow Alle Knoten (Anfragen, Updates)
- Kommunikationsprimitive: Lokaler Broadcast

SPINS: Vertrauensmodell

- Einzelne Knoten sind nicht vertrauenswürdig
 - physisch leicht manipulierbar
 - drahtlose Kommunikation störungsanfällig
- Basisstation ist vertrauenswürdig
 - gemeinsame Schlüssel mit allen Knoten
 - zentraler Schwachpunkt
- Knoten vertrauen sich selbst
 - Sensorwerte und lokale Uhr (kleiner Zeitdrift)
- Nachrichten erreichen ihr Ziel mit $WSK > 0$

SNEP vs. μ TESLA

SPINS besteht aus 2 Protokollen:

1. **SNEP: Secure Network Encryption Protocol**
 - Vertraulichkeit, Authentizität, Aktualität zwischen Knoten und der Basisstation
2. **μ TESLA: Micro Timed Efficient Stream Loss-tolerant Authentication**
 - Authentizität für Broadcast von Basisstation
 - Nur symmetrische Kryptographie
 - Asymmetrie durch Zeitverschiebung

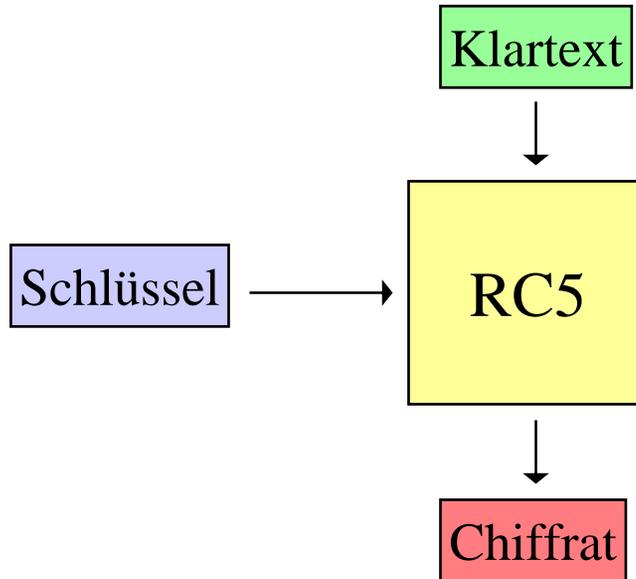
Übersicht

- Sicherheit in Sensornetzen
- Denial Of Service
- SPINS
 - **SNEP**
 - μ TESLA
- Resümee

SNEP: Überblick

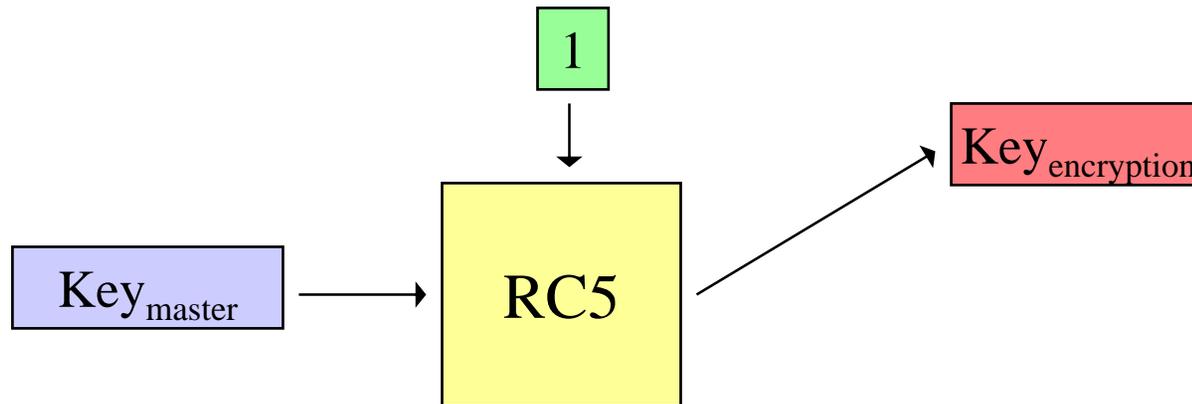
- Nur symmetrische Kryptographie
 - Nur eine kryptographische Funktion für:
 - Schlüsselgenerierung
 - Verschlüsselung und Entschlüsselung
 - MAC: Signieren von Nachrichten
 - Pseudo-Zufallsgenerator
- Einsparungen im Quellcode

RC5



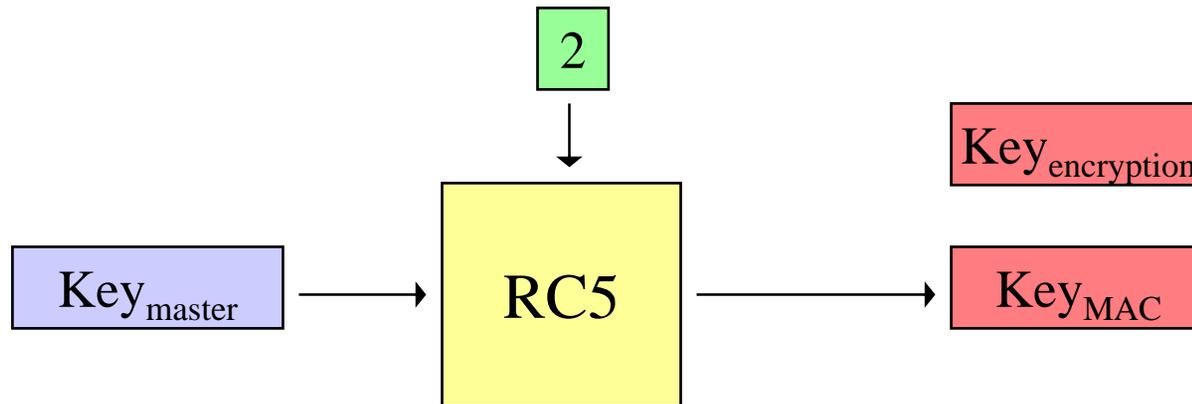
- Blockchiffrieralgorithmus, entwickelt von Ron Rivest, 1995
- Funktionsweise: Datenabhängige Rotationen
- Wort-Grösse, Rundenanzahl und Schlüssellänge wählbar
- Chiffrat gleich lang wie der Klartext (!)
- Algorithmus kompakt beschreibbar → braucht wenig Speicher

Schlüsselgenerierung



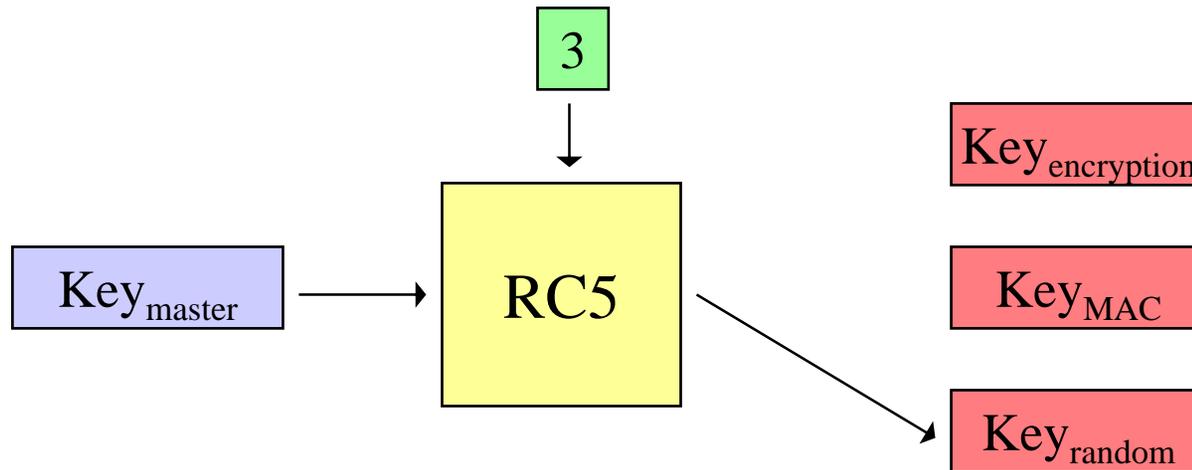
- **Jeder Knoten besitzt einen gemeinsamen Master-Key mit der Basisstation (wird vor dem Einsatz des Sensornetzes festgelegt)**
- Alle anderen Schlüssel werden aus dem Master-Key generiert:
 - *Verschlüsselungs-Schlüssel*

Schlüsselgenerierung



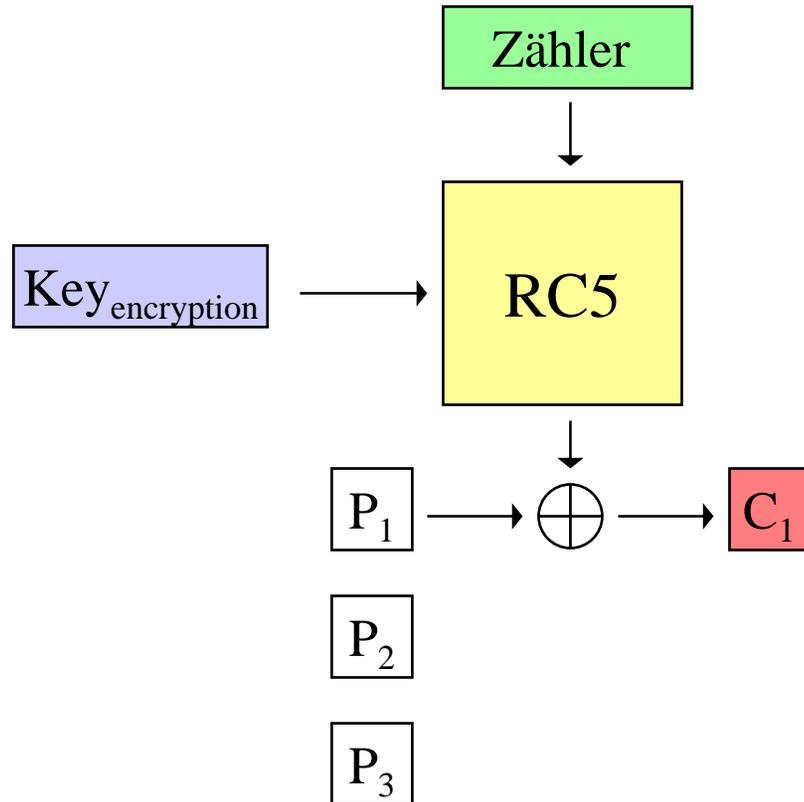
- **Jeder Knoten besitzt einen gemeinsamen Master-Key mit der Basisstation (wird vor dem Einsatz des Sensornetzes festgelegt)**
- Alle anderen Schlüssel werden aus dem Master-Key generiert:
 - *Verschlüsselungs-Schlüssel*
 - *MAC-Schlüssel*

Schlüsselgenerierung



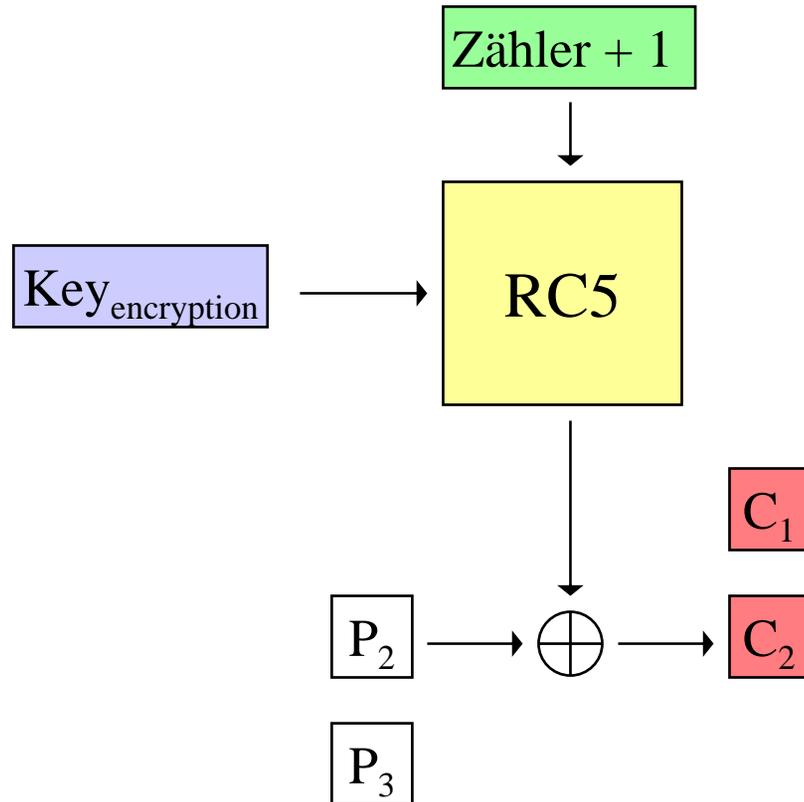
- **Jeder Knoten besitzt einen gemeinsamen Master-Key mit der Basisstation (wird vor dem Einsatz des Sensornetzes festgelegt)**
- Alle anderen Schlüssel werden aus dem Master-Key generiert:
 - *Verschlüsselungs-Schlüssel*
 - *MAC-Schlüssel*
 - *Schlüssel(=Initialwert) für den Pseudo-Zufallsgenerator*

RC5 Verschlüsselung



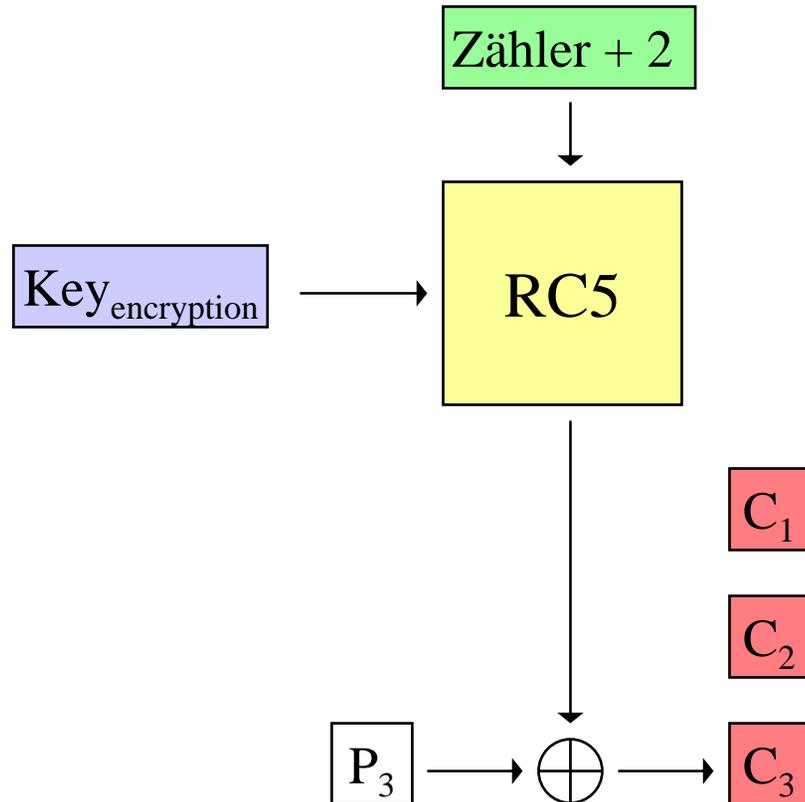
- Notation: $\{P\}_{\langle K_{\text{encr}}, \text{Zähler} \rangle}$
- RC5 wird als Stream-Cipher eingesetzt

RC5 Verschlüsselung



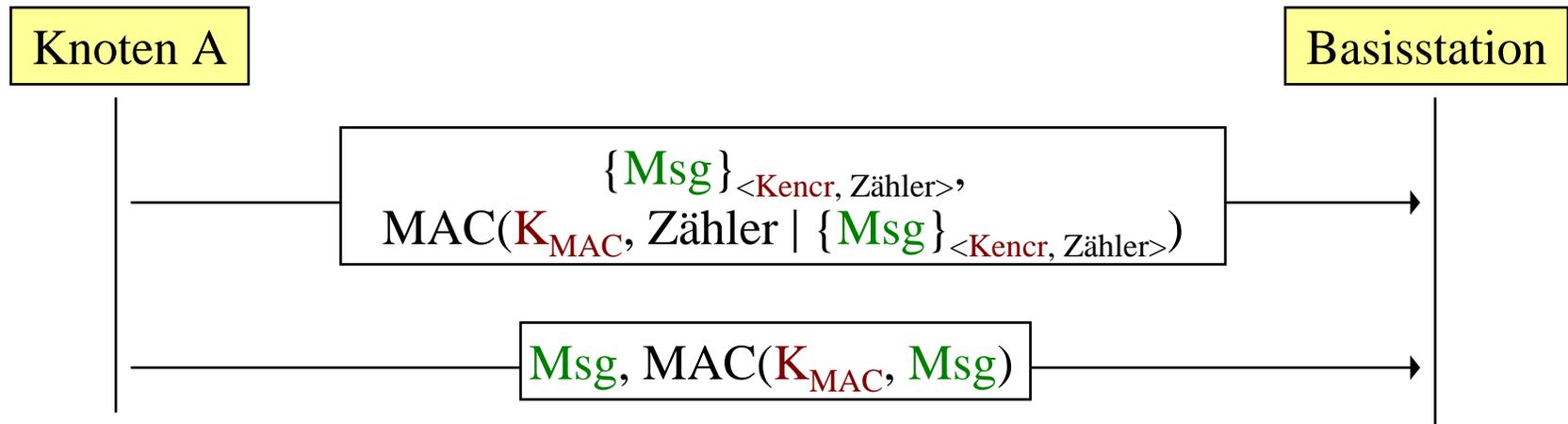
- Notation: $\{P\}_{\langle K_{\text{encr}}, \text{Zähler} \rangle}$
- RC5 wird als Stream-Cipher eingesetzt
- Knoten und Basisstation haben gleichen Zähler-Wert
- Entschlüsselung identisch

RC5 Verschlüsselung



- Notation: $\{P\}_{\langle K_{\text{encr}}, \text{Zähler} \rangle}$
- RC5 wird als Stream-Cipher eingesetzt
- Knoten und Basisstation haben gleichen Zähler-Wert
- Entschlüsselung identisch
- bei Verlust der Nachrichten: Probiere die nächsten paar Zählerwerte
- Notfalls: explizite Zähler-Resynchronisation (authentisch aber nicht verschlüsselt)

Authentifizierung & Verschlüsselung



- MAC: RC5 im *Cipher Block Chaining* - Modus
- Authentifizierung funktioniert auch ohne Verschlüsselung
- Für verschlüsselte Nachrichten wird der Zähler im MAC eingebaut
→ Aktualität, Schutz gegen Replay
- Basisstation speichert aktuellen Zählerwert für jeden Knoten

SNEP: Zusammenfassung

- Eine kryptographische Funktion für alles: RC5
- Knoten und die Basisstation haben jeweils:
 - gemeinsamen Master-Key
 - synchrone Zähler (Resynchronisation bei Bedarf)
- Vertraulichkeit, Integrität, Authentizität, Aktualität zwischen Knoten und Basisstation

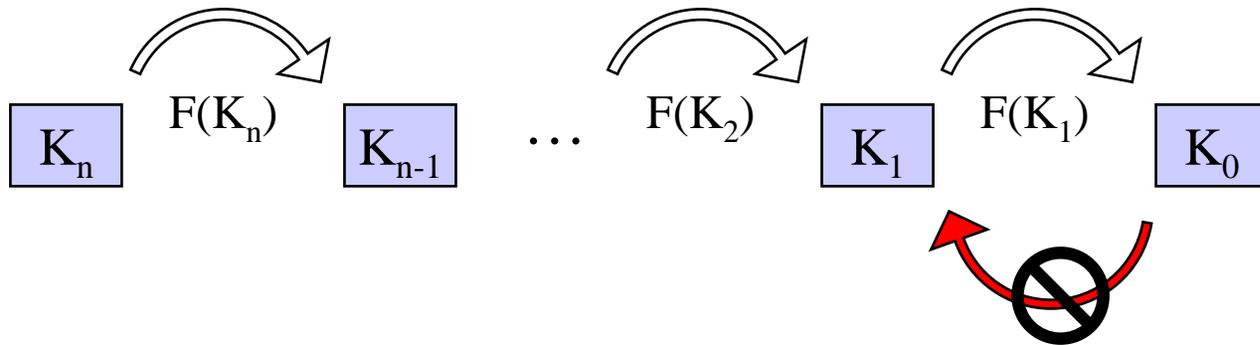
Übersicht

- Sicherheit in Sensornetzen
- Denial Of Service
- SPINS
 - SNEP
 - **μTESLA**
- Resümee

μ TESLA: Überblick

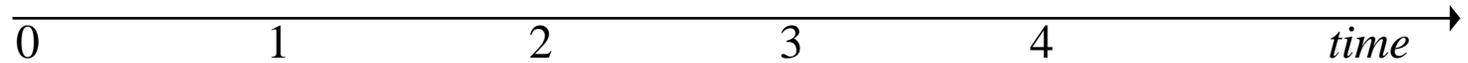
- Erlaubt authentischen Broadcast (nur Basisstation)
- Muss Asymmetrie einführen, um Fälschungen zu vermeiden
- Wir erinnern uns: Asymmetrie durch Digitale Signaturen mit PK-Kryptographie ist zu teuer
 - Berechnungen, Speicherplatz und Kommunikation
- Asymmetrie durch verzögerte Schlüssel-Vergabe

μ TESLA: Schlüsselkette



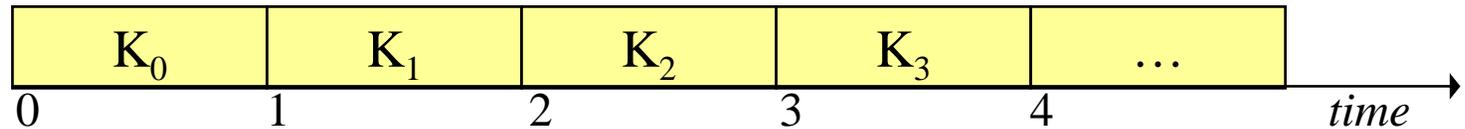
- Hauptidee: Schlüsselkette mit Einweg-Funktion
- K_0 ist der Endwert
- K_0 kann sich jeder Knoten bei der Basisstation beziehen (über SNEP)

μ TESLA: Protokoll



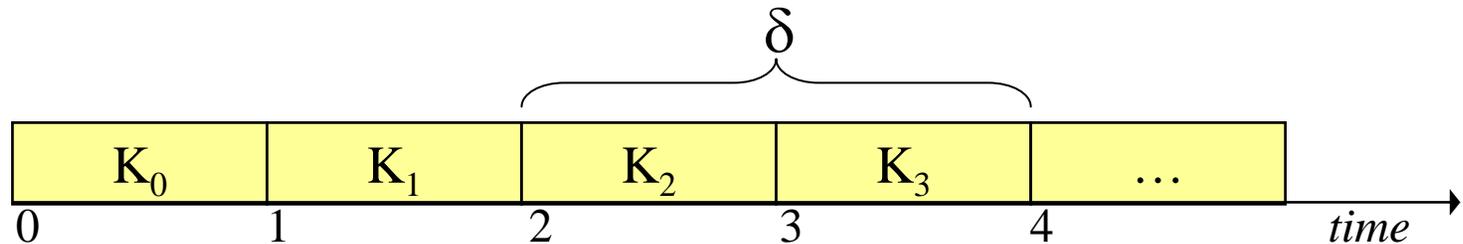
- Teile die Zeit in Intervalle ein

μ TESLA: Protokoll



- Teile die Zeit in Intervalle ein
 - Benutze den Schlüssel K_i im Interval i
- Alle Nachrichten, die im Interval i geschickt werden, benutzen K_i für die MAC-Berechnung

μ TESLA: Protokoll



- Teile die Zeit in Intervalle ein
- Benutze den Schlüssel K_i im Interval i
- **Trick: K_i wird erst nach einer Verzögerung, d.h. im Zeitintervall $i + \delta$ veröffentlicht**
- Knoten überprüfen die Authentizität von
 - K_i durch das Berechnen von $F(K_i)$ oder $F(F(K_i))$ oder ...
 - Nachricht N durch das Berechnen von $\text{MAC}(K_i, N)$

μTESLA: Zusammenfassung

- Wichtige Parameter: Zeitintervall, Verzögerung
- Verzögerung $>$ RTT um die Integrität sicherzustellen
- Dadurch maximale Verzögerung definiert, nach der eine Nachricht im Knoten bearbeitet werden kann
- Knoten müssen Nachrichten puffern, bis der Schlüssel veröffentlicht wird
- Benötigt schwach synchronisierte Uhren
- Wenn eine Schlüssel-Kette aufgebraucht ist: Knoten mit einer neuen Kette initialisieren!

SPINS: Zusammenfassung

Vorteile:

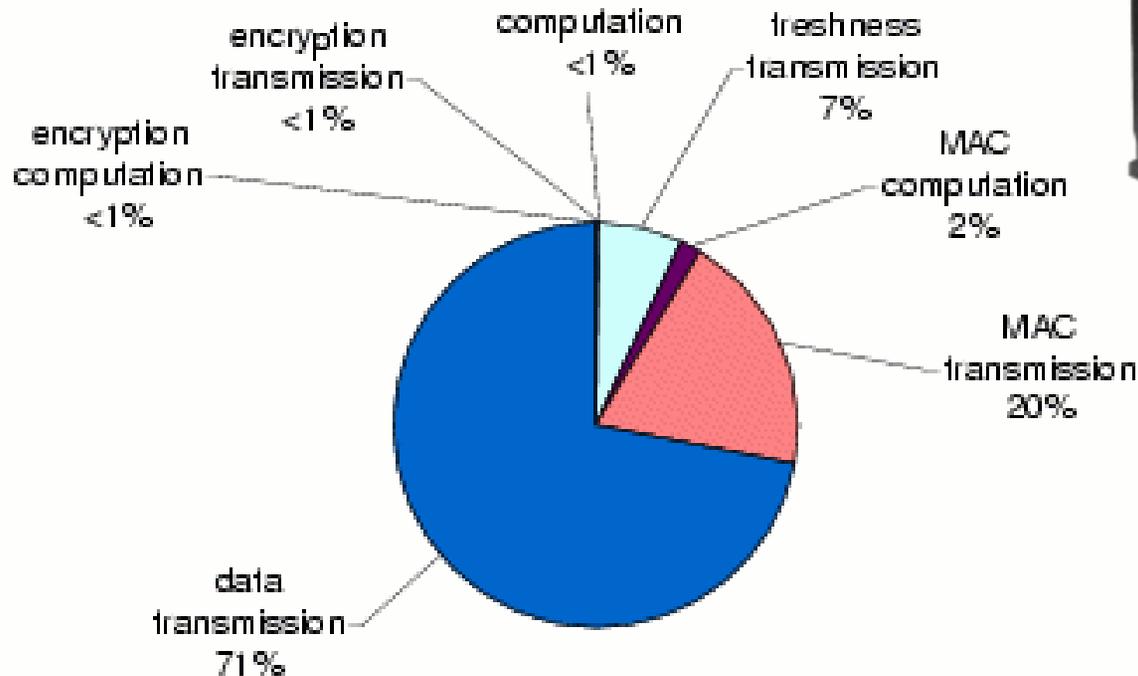
- Anforderungen erfüllt

Nachteile:

- Netz muss synchron sein
- Eindeutige IDs, z.B. Position
- Vordefinierter Master-Key pro Knoten
- Basisstation
 - Speicher: Master keys & Zähler, Schlüsselkette für Broadcasts
 - Flaschenhals der Kommunikation und der Sicherheit

SPINS: Auswertung

Energieverbrauch:



→ Die meiste Energie wird durch den Kommunikations-Overhead (8-byte MAC pro Packet) verbraucht

Übersicht

- Sicherheit in Sensornetzen
- Denial Of Service
- SPINS
 - SNEP
 - μ TESLA
- **Resümee**

Resümee

- Wir haben gesehen:
 - Schutz auf den unteren Schichten des OSI-Modells
 - Symmetrische Kryptographie für Vertraulichkeit und Authentizität (auch bei Broadcast)
- Weitere Forschungsthemen
 - Management von Schlüsseln on-the-fly (nicht vor dem Einsatz)
 - Sichere Routing-Protokolle
 - Public-Key Infrastruktur ohne zentrale Knoten (selbstorganisierend, à la PGP)

Resümee

- Sicherheit in Sensornetzen im begrenzten Rahmen möglich
- Sicherheit braucht Ressourcen
- Sicherheit bedeutet Overhead
- Sicherheit bereits zur Designzeit wichtig
 - Netz-Topologie, Basisstation, Routing, ...



Referenzen

- Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar. *SPINS: Security Protocols for Sensor Networks*. Mobile Computing and Networking, 2001.
- Anthony D., John A. Stankovic. *Denial of Service in Sensor Networks*. IEEE Computer, 2002.
- Ronald L. Rivest. *The RC5 Encryption Algorithm*. Available via anonymous ftp:
<ftp://theory.lcs.mit.edu/pub/rivest/rc5/rc5.ps>, 1995.