Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Authentischer und effizienter Datentransport in drahtlosen Sensornetzen

Studienarbeit am Institut für Telematik
Prof. Dr. M. Zitterbart
Fakultät für Informatik
Universität Karlsruhe (TH)

von

cand. inform.
Joachim Wilke

Betreuer:
Prof. Dr. M. Zitterbart
Dipl.-Inform. Erik Blaß

Tag der Anmeldung: 13. Februar 2006
Tag der Abgabe: 13. August 2006

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 13. August 2006

Inhaltsverzeichnis

1. Einleitung

Die technische Entwicklung der vergangenen Jahre und die damit fortschreitende Miniaturisierung eröffnet für die Computertechnik immer neuere Anwendungsgebiete. Ein aktueller Trend besteht in der Integration von Sensoren, Prozessoren, Funktechnik, Arbeits- und Energiespeichern in so genannte Sensorknoten. Diese völlig autarken Geräte sind in der Lage Netzwerke zu formen, ihre Umwelt sensorisch zu erfassen und Messdaten auszuwerten und weiterzuleiten.

Um ein möglichst breites Anwendungsfeld zu erschließen, geht die Entwicklung hin zu immer kleineren Sensoren, die sich unauffällig in unsere gewohnte Umwelt einfügen können. Denkbar sind so etwa neben Umwelt- und Gebäudeüberwachung auch medizinische [oEAS] und militärische [Saus05, Qadi] Applikationen.

Analysen in [RSPS02] haben gezeigt, dass die Übermittlung von Daten per Funk den größten Anteil am Energieverbrauch eines Sensorknotens ausmacht. Viele Energiesparmaßnahmen zielen daher darauf ab, die notwendige Datenkommunikation zu reduzieren. In vielen Anwendungsfällen benötigt die Basisstation nicht alle detaillierten Messwerte, sondern es reichen Approximationen und Zusammenfassungen, wie Durchschnitt, Minimum, Maximum oder ähnliches, aus. Ist dies der Fall, liegt eine sehr effektive Energiesparmaßnahme in Datenaggregation [EGHK99, IEGH01]. Hierbei leiten Netzknoten nicht jedes einzelne Datum weiter. Statt dessen aggregieren sie mehrere empfangene Werte mittels einer oder mehrerer, dem Anwendungszweck entsprechenden, Aggregationsfunktionen. Das daraus resultierende Aggregat wird dann an Stelle der detaillierten Werte übertragen.

In vielen Anwendungsbereichen ist Sicherheit von hoher Priorität, so etwa in den genannten Bereichen der Medizin oder dem Militär. Die Verwendung von Datenaggregation hat jedoch den Nachteil, dass jeder daran beteiligte Knoten die übertragenen Daten einsehen und beliebig verändern kann. Damit lässt sich nicht mehr einwandfrei sagen, ob das empfangende Aggregat aus den realen Messwerten der einzelnen Sensoren entstanden ist, oder ob einzelne Sensorknoten böswillig Manipulationen vorgenommen haben.

1.1 Zielsetzung der Arbeit

Zielsetzung dieser Arbeit ist es, ein Verfahren zu entwickeln, das trotz Einsatz von Datenaggregation ein gewisses Maß an Sicherheit hinsichtlich der Authentizität der Daten gewährleistet. Dabei handelt es sich um die eigentliche Herausforderung, denn wie im nächsten Kapitel genauer erklärt wird, sind Datenaggregation und Gewährleistung von Authentizität prinzipiell unvereinbare Anforderungen.

Weiterhin sollen einige Randbedingungen eingehalten werden. So soll das Verfahren für jede denkbare Aggregationsfunktion praktikabel sein. Außerdem soll der von der Anwendung abhängige Grad an notwendiger oder gewünschter Sicherheit über einen Systemparameter einstellbar sein. So kann ein optimaler Tradeoff zwischen erreichter Sicherheit und Energieverbrauch erreicht werden.

1.2 Gliederung der Arbeit

Das nächste Kapitel befasst sich mit einer detaillierten Analyse der Problemstellung und zeigt Ideen für eine Lösung auf. Abgeschlossen wird mit einem Überblick über den aktuellen Stand der Forschung auf diesem Gebiet.

Auf die eigene Problemlösung wird im Anschluss in Kapitel 3 eingegangen.

Abgeschlossen wird mit Kapitel 4 in dem Simulations- und Messergebnissen vorgestellt werden, die die Funktions- und Leisungsfähigkeit aufzeigen.

2. Analyse

Sensornetze bieten aufgrund ihrer Struktur vielfältige Angriffsmöglichkeiten. Insbesondere die drahtlose Kommunikation als auch der oft freie Zugang zu einzelnen Knoten eröffnen neue Sicherheitsrisiken die beachtet werden müssen [PeSW04]. Im Folgenden wird zwischen passiven und aktiven Angreifern unterschieden.

Die Übertragung von Daten in Sensornetzwerken erfolgt per Funk. Passive Angreifer sind in der Lage diesen Netzwerkverkehr zwischen den einzelnen Sensorknoten abzuhören. Obwohl die Übertragungsreichweite der Knoten begrenzt ist, geht man in der Regel davon aus, dass der Angreifer vom gesamten Netzwerkverkehr Kenntnis erlangen kann.

Aktive Angreifer erlangen im Gegensatz zu passiven Angreifern auch physikalischen Zugriff auf einen oder mehrere Teilnehmer des Sensornetzwerks. Dadurch können sie den gesamten Speicher dieser Knoten auslesen und erhalten so Kenntnis über darin gespeicherte Informationen wie Messdaten und kryptographische Schlüssel. Dies ermöglicht ihnen oft auch gefälschte Pakete erfolgreich in das Netzwerk einzuspeisen.

Ein Sensornetzwerk, das sensible Daten erfasst und überträgt, sollte sowohl gegen passive als auch aktive Angriffe resistent sein.

Um sich vor passiven Angreifern zu schützen, reicht eine einfache Verschlüsselung des gesamten Netzwerkverkehrs auf Basis eines für alle Knoten einheitlichen Schlüssels aus. Ohne diesen Schlüssel kann der Angreifer weder den abgehörten Datenverkehr entschlüsseln, noch kann er eigene Pakete in das Netzwerk einspeisen.

Handelt es sich jedoch um einen aktiven Angreifer ist es für diesen leicht den globalen Schlüssel zu erlangen. Er ist dann in der Lage gefälschte Pakete in das Netzwerk einzuspeisen. Eine andere Idee ist es, dass jeder Knoten einen eigenen Schlüssel mit der Basisstation gemeinsam hat. Das begrenzt den Schaden den der Angreifer anrichten kann, da er nur noch Pakete im Namen der kompromittierten Knoten versenden kann.

Ein großer Nachteil dabei ist, dass so auch die Datenaggregation erschwert wird. Ein auf diesem Weg verschlüsseltes Datenpaket kann nur von dem sendenden Knoten und der Basisstation entschlüsselt werden. Alle anderen Knoten können das nicht und

sind nicht ohne Weiteres in der Lage eine Aggregation durchzuführen. Das Paket wird unverändert an sein Ziel weitergeleitet werden und verursacht damit deutlich höhere Übertragungskosten als dies mit Datenaggregation der Fall wäre.

2.1 Anforderungen

Wünschenswert ist eine Verfahren, das zum einen Aggregationen ermöglicht ohne dadurch leicht angreifbar zu werden. Im Einzelnen sollte eine zufriedenstellende Lösung deshalb folgenden Punkten gerecht werden:

- Verschlüsselung des Datenverkehrs zum Schutz gegen Angreifer.
  Das Sensornetzwerk sollte sowohl gegen passive als auch aktive Angreifer resistent sein. Dem Angreifer soll es weder ermöglicht werden, Datenverkehr zwischen den Knoten abzuhören, noch soll es ihm möglich sein, gefälschte Pakete in das Netzwerk einzuspeissen und so dessen Betriebsablauf zu stören oder Messergebnisse zu verfälschen. Klar ist, das kein Protokoll in Gegenwart von aktiven Angreifern vollständige Sicherheit bieten kann. Erlangt der Angreifer auf einen Großteil der Knoten physikalischen Zugriff, kann er die Netzwerkfunktionen empfindlich stören.

- Nutzung von Datenaggregation zur Energieeinsparung
  Die Funkübertragung von Daten ist die energieaufwendigste Aktion jedes Sensorknotens. Trotz gegebenen Sicherheitsansprüche soll es daher möglich sein, mittels Datenaggregation das zu übertragene Datenvolumen zu reduzieren und damit durch die erzielte Energieeinsparung eine längere Lebenszeit des Sensornetzwerks zu ermöglichen.

- Unabhängigkeit des Verfahrens von den genutzen Aggregationsfunktionen
  Aggregation ist in einem Sensornetzwerk oft nicht nur auf das Bilden von Summen und Durchschnitten beschränkt. Eine optimale Lösung ist unabhängig von der oder den verwendeten Aggregationsfunktionen einsetzbar.

- Betrugsversuche durch kompromittierte Knoten müssen erkannt werden.
  Ein Betrugsversuch eines Aggregators muss erkannt werden. Wünschenswert wäre dabei das Identifizieren des kompromittierten Knotens um diesen aus dem Netzwerk auszuschliessen.

2.2 Annahmen

Sowohl der Aufbau des Aggregationsbaums als auch die Schlüsselverteilung werden im Folgenden als gegeben betrachtet. Insbesondere verfügt jeder Knoten im Netzwerk über paarweise Schlüssel mit seinen Vorgängern im Aggregationsbaum. Hierfür stehen effiziente Verfahren zur Verfügung [BlZi06, ZhSJ03].

Weiterhin wird angenommen, dass Pakete im Sensornetzwerk entlang des aufgebauten Aggregationsbaums „geroutet“ werden. Korrumpierte Blattknoten, die falsche Messwerte versenden, werden nicht berücksichtigt, da keine Absicherung gegen solch einen Betrugsversuch möglich ist. Gleiches gilt für die Datensenke. Im Folgenden werden Blattknoten und Senke somit als ehrlich betrachtet.

2.3 Stand der Forschung

Die folgenden Abschnitte bieten einen Überblick über vorhandene Forschungsarbeiten im Bereich der sicheren Aggregation.

[ZSJN04] schlägt ein Verfahren vor, um in Gegenwart von bis zu k kompromittierten Knoten jedes von diesen Knoten gefälschte Datenpaket zu entdecken. Dazu fordert es, dass jedes gemessene Ereignis von mindestens k+1 Knoten bestätigt wird. Zwar wird so jede gefälschte Information erfolgreich herausgefiltert, jedoch ist auch ein deutlich erhöhter Kommunikationsaufwand erforderlich. Mögliche Einsparpotentiale durch Aggregation werden nicht betrachtet.

In [DiFo04] werden Datenpakete zum Schutz vor passiven Angreifern verschlüsselt. Die Aggregatoren entschlüsseln die Pakete, berechnen das Aggregat und senden das Ergebnis mit einem neuen Schlüssel codiert weiter. Ein kompromittierter Knoten kann hier jedoch Schaden anrichten. Um dies zu verhindern setzt das Verfahren einen externen Mechanismus vorraus, der solche Knoten erkennt und ausschliesst.

Ohne solch einen Mechanismus kommt [PrSP03] aus. Hier wird das von einem Aggregator berechnete Datum mittels spezieller von der Aggregationsfunktion abhängigen Verfahren überprüft, etwa mittels Stichproben aus dem Datenbestand der aggregiert wurde. Damit wird sichergestellt, dass das Ergebnis mit einer bestimmten Wahrscheinlichkeit innerhalb eines definierten Intervalls um den tatsächlichen Aggregationswert liegt und daher wahrscheinlich nicht wesentlich manipuliert wurde. Da das Prüfungsverfahren von der verwendeten Aggregationsfunktion abhängig ist, lässt es sich nicht auf beliebige Aggregationsfunktionen erweitern.

Die Verwendung von privaten Homomorphismen nutzen [WeGS06], [CaMT05] und [AcGW05]. Ein privater Homomorphismus ist eine Verschlüsselungsfunktion die es erlaubt bestimmte Rechenoperationen direkt auf den verschlüsselten Daten durchzuführen (etwa Addition oder Multiplikation). Dadurch können Knoten empfangene verschlüsselte Datenpakete aggregieren und weitersenden, ohne dass sie selbst vom Inhalt Kenntnis erlangen. Die größte Einschränkung dieses Verfahrens ist, dass es nur auf eine beschränkte Gruppe von Aggregationsfunktionen anwendbar ist.

Als Letztes wird Secure Aggregation for Wireless Networks (SAWN) aus [HuEv03] vorgestellt. Da es von den bisher vorgestellten Verfahren die Anforderungen aus Abschnitt 2.1 am besten erfüllt, wird es im nächsten Abschnitt ausführlicher vorgestellt.

2.3.1 Secure Aggregation for Wireless Networks (SAWN)

Das Verfahren von Hu und Evans verwendet verzögerten Aggregation und Authentizität. Verzögerte Aggregation bedeutet, dass Sensordaten entlang des Aggregationsbaumes nicht direkt vom Elternknoten aggregiert werden, sondern dieser die Daten lediglich unverändert weiterleitet. Die eigentliche Aggregation erfolgt später. In diesem Verfahren findet sie im Großelternknoten des betreffenden Sensors statt. Verzögerte Authentifikation bedeutet, dass die Authentizität von Nachrichten nicht direkt beim Empfangen der Nachricht überprüft werden kann, sondern erst nach einer bestimmten zeitlichen Verzögerung.

Zur Realisierung werden weitergehende Voraussetzungen an das Sensornetzwerk gestellt, als in Abschnitt 2.1 beschrieben. Zusätzlich gefordert werden:

- Die Sendeleistung der Basisstation ist deutlich grösser als die der einzelnene Sensorknoten. Sie kann damit Broadcast-Nachrichten versenden, die alle Sensorknoten direkt empfangen können.
- Das verwendete Netzwerkprotokoll garantiert einen zuverlässigen Nachrichtentransport.

2.3.1.1 Senden- und Empfangen von Nachrichten

Zur Authentifizierung der von der Basistation gesendeten Daten wird das μTESLA-Protokoll [PSWC+01] verwendet. Dieses Protokoll ist speziell auf die Verwendung in Sensornetzwerken mit ihren eingeschränkten Resourcen zugeschnitten. Dabei erstellt die Basisstation eine Liste von Schlüsseln $K_i = F(K_{i+1})$, die mit einer öffentlichen Einwegfunktion F berechnet wurden.

Alle Sensorknoten werden initial mit K_0 versorgt. Dies ermöglicht ihnen die Authentifizierung weiterer Nachrichten der Basisstation. Dazu verschlüsselt sie ihre erste Nachricht an die Sensorknoten mit dem Schlüssel K_1, versendet die verschlüsselte Nachricht und veröffentlicht im Anschluss den verwendeten Schlüssel K_1. Die Sensorknoten können damit die empfangene Nachricht entschlüsseln und die Authentizität überprüfen, da $F(K_1) = K_0$ gelten muss. Dies wird fortgesetzt, bis der letzte Schlüssel K_n erreicht wird. In dieser letzten Nachricht kann eine neue Schlüsselliste initialisiert werden, mit der dann weitere Übertragungen möglich sind.

Damit ein Sensorknoten nicht nur Nachrichten der Basisstation empfangen und authentifizieren kann, sondern selbst auch Daten versenden kann, ist ein weiterer Schlüssel notwendig. Jeder Sensorknoten i besitzt dazu einen symmetrischen Schlüssel K_{iS} der die Basis sämtlicher Übertragungen des Knoten darstellt. Möchte der Knoten eine Nachricht Richtung Basisstation senden, verschlüsselt er diese mit einem auf der Basis von K_{iS} berechneten temporären Schlüssel $K_{ic} = E(K_{iS}, c)$ der sich aus der Verschlüsselung eines Zählers c mit dem Schlüssel K_{iS} ergibt.

2.3.1.2 Sichere Aggregation der Messdaten

Die Übertragung, Authentifikation und Aggregation von Messdaten läuft wie folgt ab: Jeder Blattknoten i im Aggregationsbaum sendet seinen Messwert D_i zusammen mit seiner Knoten-ID ID_i und einem Authentifizierungscode $\mathrm{MAC}(D_i, ID_i, K_{ic})$. Der zur Berechnung des Authentifizierungscodes verwendete Schlüssel K_{ic} ist initial lediglich dem Knoten und der Basisstation bekannt. Der Elternknoten der die Nachricht empfängt, speichert sie zunächst ab, da er ohne den Schlüssel K_{ic} nicht die Authentizität der Nachricht sicherstellen kann. (Erst wenn die Basisstation den Schlüssel offenlegt, kann der Knoten die Authentizität überprüfen und im Falle eines Fehlers einen Alarm auslösen.)

Obwohl die Authentizität der empfangenen Nachrichten noch nicht überprüft wurde, erzeugt der Knoten, sobald er alle Messwerte empfangen hat, das Aggregat $Aggr$ dieser Werte. Anschliessend leitet er die empfangenen Messwerte D_{i_1}, D_{i_2}, ..., die IDs der beteiligten Knoten ID_{i_1}, ID_{i_2}, ... und einen Authentifizierungscode $\mathrm{MAC}(K_{jc}, Aggr)$ an seinen Elternknoten weiter. Das berechnete Aggregat selbst wird nicht weitergeleitet.

Der nachfolgende Knoten, also der Großelternknoten der betrachteten Blattknoten, führt nun ebenfalls eine Aggregation der empfangenen Messwerte D_{i_1}, D_{i_2}, ... durch.

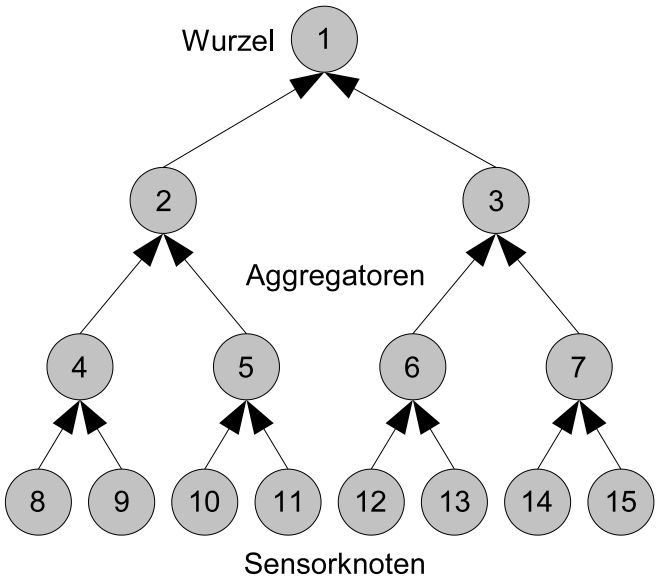


Abbildung 2.1: Aggregationsbaum mit 15 Sensorknoten

Um sicherzustellen, dass dem vorherigen Knoten keine (absichtlichen) Fehler unterlaufen sind, berechnet er zusätzlich das Aggregat der Werte, die diesen Knoten passiert haben. Dies muss er für jeden seiner Kindknoten durchführen.

Dieses Verfahren wird rekursiv fortgesetzt, bis die aggregierten Werte die Wurzel des Aggregationsbaums erreicht haben.

2.3.1.3 Beispiel

Betrachtet wird das Sensornetz aus Abbildung 2.1. Die Knoten 8-15 möchten ihre Messwerte der Basisstation (Knoten 1) mitteilen. Bei der verwendeten Aggregationsfunktion handelt es sich um den Durchschnitt. Die versendeten Nachrichten sind in Tabelle 2.1 aufgeführt.

Gut zu erkennen ist die Tatsache, dass die Größe der übertragenen Nachrichten ab Schritt 2 nicht mehr weiter zunimmt. Dagegen steigt die Größe der Nachricht der Basisstation in Schritt 4 linear mit der Anzahl der im Netz aktiven Knoten.

2.3.1.4 Bewertung

Dieses Verfahren kann somit beliebige Aggregationsfunktionen verarbeiten. Nachteile sind neben der Notwendigkeit von wide-area Broadcasts, die einen leistungsstarken Wurzelknoten erfordern, auch die fehlende Skalierbarkeit für größere Netzwerke. Die größe des Broadcasts steigt linear mit der Zahl der Knoten im Netzwerk. Zudem können zwei im Aggregationsbaum direkt hintereinander sitzende kompromittiere Knoten Einfluß auf die Aggregation nehmen und gefälschte Daten einspeisen ohne dass dies entdeckt werden kann.

Schritt	Sender	→	Empfänger	Paketinhalt		
1	8	→	4	D_8	ID_8	$\text{MAC}(K_{8,1}, D_8)$
	9	→	4	D_9	ID_9	$\text{MAC}(K_{9,1}, D_9)$
	10	→	5	D_{10}	ID_{10}	$\text{MAC}(K_{10,1}, D_{10})$
	11	→	5	D_{11}	ID_{11}	$\text{MAC}(K_{11,1}, D_{11})$
	12	→	6	D_{12}	ID_{12}	$\text{MAC}(K_{12,1}, D_{12})$
	13	→	6	D_{13}	ID_{13}	$\text{MAC}(K_{13,1}, D_{13})$
	14	→	7	D_{14}	ID_{14}	$\text{MAC}(K_{14,1}, D_{14})$
	15	→	7	D_{15}	ID_{15}	$\text{MAC}(K_{15,1}, D_{15})$
2	4	→	2	D_8	ID_8	$\text{MAC}(K_{8,1}, D_8)$
				D_9	ID_9	$\text{MAC}(K_{9,1}, D_9)$
						$\text{MAC}(K_{4,1}, Aggr(D_8, D_9))$
	5	→	2	D_{10}	ID_{10}	$\text{MAC}(K_{10,1}, D_{10})$
				D_{11}	ID_{11}	$\text{MAC}(K_{11,1}, D_{11})$
						$\text{MAC}(K_{5,1}, Aggr(D_{10}, D_{11}))$
	6	→	3	D_{12}	ID_{12}	$\text{MAC}(K_{12,1}, D_{12})$
				D_{13}	ID_{13}	$\text{MAC}(K_{13,1}, D_{13})$
						$\text{MAC}(K_{6,1}, Aggr(D_{12}, D_{13}))$
	7	→	3	D_{14}	ID_{14}	$\text{MAC}(K_{14,1}, D_{14})$
				D_{15}	ID_{15}	$\text{MAC}(K_{15,1}, D_{15})$
						$\text{MAC}(K_{7,1}, Aggr(D_{14}, D_{15}))$
3	2	→	1	D_4	ID_4	$\text{MAC}(K_{4,1}, D_4)$
				D_5	ID_5	$\text{MAC}(K_{5,1}, D_5)$
						$\text{MAC}(K_{2,1}, Aggr(D_4, D_5))$
	3	→	1	D_6	ID_6	$\text{MAC}(K_{6,1}, D_6)$
				D_7	ID_7	$\text{MAC}(K_{7,1}, D_7)$
						$\text{MAC}(K_{3,1}, Aggr(D_6, D_7))$
4	1	→	Broadcast	$K_{4,1}$	$K_{5,1}$	$K_{6,1}$
				$K_{7,1}$	$K_{8,1}$	$K_{9,1}$
				$K_{10,1}$	$K_{11,1}$	$K_{12,1}$
				$K_{13,1}$	$K_{14,1}$	$K_{15,1}$

Tabelle 2.1: Ablauf einer Aggregation mit SAWN

2.4 Zusammenfassung

Keines der Verfahren erfüllt alle in Abschnitt 2.1 genannten Anforderungen an ein zufriedenstellendes Verfahren. Mit dem im nächsten Kapitel vorgestellten Verfahren „Extended Secure Aggregation for Wireless Networks“ (ESAWN) wird dies erreicht.

3. Entwurf

Das im Folgenden vorgestellte Verfahren basiert auf redundanter Berechnung von Aggregationsergebnissen und dessen Vergleich durch Dritte. Diese Redundanz wird durch Zeugen erreicht, auf die im Abschnitt 3.2 genauer eingegangen wird.

3.1 Schutz vor passiven Angreifern

Eine der Voraussetzungen aus Abschnitt 2.2 ist es, dass jeder Knoten Schlüssel zur sicheren Kommunikation mit allen Knoten besitzt, mit denen er kommunizieren muss. Dies umfasst sämtliche Knoten, die sich zwischen ihm und der Wurzel, also auf seinem Aggregationspfad befinden. Damit wird eine „Knoten-zu-Knoten-Authentizität“ erreicht, die jedem Knoten Klarheit über den direkten Ursprung eines Datenpakets verschafft. Kommunizieren zwei Knoten direkt miteinander gewährleistet diese Art von Authentizität, dass einer der Knoten feststellen kann ob ein empfangenes Datenpaket tatsächlich von dem anderen Knoten stammt. Außerdem ist die Vertraulichkeit der Daten gewährleistet, denn Datenpakete werden mit dem jeweiligen Schlüssel verschlüsselt. So wird sowohl ein Mithören von passiven Angreifern als auch von anderen Knoten innerhalb des Sensornetzwerks verhindert.

Schutz vor aktiven Angreifern bietet dies aber nur beschränkt. Gelingt es einem Angreifer durch physikalischen Zugriff den Speicher eines Knotens i auszulesen, gelangt er in den Besitz der Schlüssel mit dem i die Kommunikation mit seinen verschiedenen Kommunikationspartnern verschlüsselt. Er kann Nachrichten von i fälschen und etwas völlig anderes als das korrekte Aggregat der Eingangsdaten weitermelden. Eine „Ende-zu-Ende-Authentizität“ welche der Senke die Authentizität der Daten auch über mehrere Aggregatoren hinweg, garantieren würde, ist so folglich nicht zu erreichen. Da die Aggregatoren nicht unbedingt vertrauenwürdig sind und Werte auf dem Weg zur Wurzel verändert haben könnten, kann die Wurzel nicht zweifelsfrei feststellen ob das erhalten Aggregat tatsächlich aus den ursprünglichen Messwerten berechnet wurde.

Um solch einen kompromittierten Knoten zu entdecken, kann man Zeugen benutzen, wie sie im nächsten Abschnitt beschrieben werden.

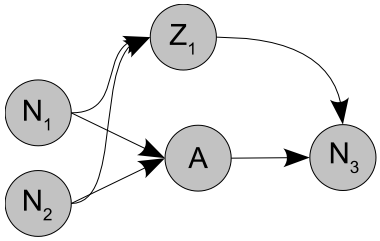


Abbildung 3.1: Aggregation mit einem Zeugen

3.2 Zeugen

Grundlegende Idee dieses Vefahrens ist es, jedem Knoten der Aggregationen durchführt, einen oder mehrere Zeugen zuzuweisen, die dessen Aggregationen überprüfen. Dazu muss jeder Knoten der Daten an einen Aggregator schickt, diese Daten auch an die dazugehörigen Zeugen schicken.

3.2.1 Verwendung eines Zeugen

Die Zeugen können damit die vom Aggregator durchgeführte Aggregation selbst nachvollziehen. In Abbildung 3.1 läuft das zum Beispiel wie folgt ab: Die Knoten N_1 und N_2 schicken ihr Datum wie bei einer normalen Aggregation an den Aggregator A und zusätzlich an den Zeugen Z_1. Sowohl Aggregator A als auch Zeuge Z_1 berechnen nun nach der vorgegebenen Aggregationsfunktion das Aggregat der empfangenen Daten und teilen es N_3, dem nächsten Knoten der Aggregationskette, mit. N_3 geht nur dann von einer fehlerfreien Aggregation aus, wenn er sowohl von A als auch von Z_1 das gleiche Ergebnis übermittelt bekommt.

Versucht der Aggregator zu betrügen, fällt dies N_3 beim Vergleich der empfangenen Aggregate auf. Der Knoten der einen Betrug bemerkt, muss dies der Senke melden. Eine Möglichkeit ist das Senden der Meldung an den Vaterknoten, also die Weiterleitung entlang des Aggregationspfades. Liegt auf dem Weg zwischen Zeuge und Wurzel ein weiterer böser Knoten, kann dieser die Meldung abfangen. Eine zweite Möglichkeit ist das Broadcasten der Meldung. Dadurch steigt die Wahrscheinlichkeit einen Kommunikationspfad zwischen Wurzel und Zeugen zu finden, der keinen bösen Knoten enthält. Diese Möglichkeit ist zwar zuverlässiger aber auch wesentlich teurer als die erste Möglichkeit. Nach dieser Meldung sollte der Aggregator seine Arbeit einstellen, dadurch fällt anderen Knoten auf, dass es an dieser Stelle ein Problem gab.

Wenn A *und* Z_1 gemeinsam einen Betrug versuchen, bleibt dieser jedoch unentdeckt.

3.2.2 Situation mit k Angreifern

In vielen Fällen ist ein Zeuge nicht ausreichend. Hat ein Angreifer zwei Knoten kompromittiert und handelt es sich dabei um einen Aggregator und seinen Zeugen, ist der Angreifer wie zuvor in der Lage das Aggregat zu fälschen. Durch die Verwendung von mehr als einem Zeugen lässt sich aber auch solch ein Betrugsversuch aufdecken. Es reicht aus, wenn mindestens ein Zeuge den Betrug entdeckt. In Gegenwart von k Angreifern können also bis zu k Zeugen notwendig sein, um einen Betrugsversuch garantiert zu entdecken.

k stellt also einen Parameter da, mit dem die Zahl der Zeugen je Aggregator variiert werden kann. Abhängig von der konkreten Anwendung kann dieser Parameter so gewählt werden, dass das „nötige“ Maß an Sicherheit erreicht wird. Dabei muss beachtet werden, dass die Energiekosten mit steigenden k ebenfalls ansteigen, da mehr Pakete verschlüsselt und versendet werden müssen. Spätestens wenn mehr Pakete verschickt werden als dies ohne Aggregation nötig wäre, ist das Verfahren nicht mehr sinnvoll, da dann auch die Energiekosten der Aggregation die der Datenübertragung ohne Aggregation übersteigen.

Die Nutzung von k Zeugen ermöglicht es, jeden Betrugsversuch mit bis zu k Angreifern zu erkennen. Beachtet werden muss allerdings, dass es nicht möglich ist, den oder die kompromittierten Knoten zweifelsfrei zu identifizieren. So könnten Zeugen auch fälschlicherweise einen Betrugsversuch melden und so für einen Fehlalarm sorgen. Dieses Verfahren schützt also nicht generell vor Störversuchen durch Angreifer, sondern nur vor dem Versuch verdeckt Manipulationen an den Aggregationsdaten durchzuführen.

3.2.3 Zuweisung der Zeugen

Die Auswahl der Zeugen für jeden potentiellen Aggregator erfolgt während der Aufbauphase des Aggregationsbaum, also vor jeglicher Datenübertragung. Kandidaten sind alle Knoten die zwischen dem Aggregator und der Wurzel auf dem Aggregationspfad liegen. Dem Aggregator nahe liegende Knoten werden dabei bevorzugt ausgewählt, um den Kommunikationsaufwand möglichst niedrig zu halten.

Enthält der Aggregationspfad weniger als k Knoten wird k für diesen Aggregator entsprechend reduziert. Dies reicht aus, da in diesem Fall einer der Zeugen der Wurzelknoten ist, der per Definiton vertrauenswürdig ist.

Die richtige Wahl der Zeugen ist wichig um die Korrektheit des Verfahrens sicherzustellen. Werden Knoten als Zeugen bestimmt, die nicht Elternknoten des betrachteten Knoten sind, kann ein kompromittierter Knoten den Zeugen leicht austricksen. Dazu muss er dem Zeugen lediglich das korrekte Aggregat agg senden, während er entlang der Aggregationskette das gefälschte Aggregat agg' verbreitet. Ist der Zeuge dagegen selbst Teil der Aggregationskette, fällt solch ein Betrugsversuch auf. Diese rekursiv aufbauende Überprüfung und Kontrolle ermöglicht es der Basistation, die Authentizität und Korrektheit des Aggregats sicherzustellen.

3.2.4 Wahrscheinlichkeit einer Überprüfung

Um die Energiekosten zu reduzieren, kann über einen Parameter p die Wahrscheinlichkeit festegelegt werden, mit der ein Aggregator von seinen Zeugen überprüft wird. Mit z.B. $p = 0,5$ wird im Mittel nur noch jede zweite Aggregation eines jeden Aggregators durch Zeugen abgesichert. Dies spart Aggregationsvorgänge, Verschlüsselungen und Übertragung von zusätzlichen Paketen.

Wichtig dabei ist, dass die Kindknoten des betrachteten Aggregationsknoten synchronisiert arbeiten. Entweder übertragen alle ihre Daten auch an den oder die korrespondierenden Zeugen oder keiner. Mit einem Teil der Daten kann ein Zeuge das Aggregat nicht oder nur näherungsweise berechnen.

Es gibt verschiedene Möglichkeiten diese Synchronisation zu erreichen. Eine Möglichkeit ist es, für jeden $\lceil 1/p \rceil$-ten Wert eine Überprüfung durchzuführen. Mit $p = 0,5$

wird so genau jedes zweite Aggregat des Knoten überprüft. Dies ermöglicht es dem Knoten jedoch vorherzusagen, wann er überprüft wird. Somit kann er gezielt dann Falschinformationen in das Netzwerk leiten, wenn keine Überprüfung ansteht.

Eine weitere Möglichkeit der Synchronisation besteht in der Nutzung von Pseudozufallszahlen. Hierbei wird jeweils ein zufällig gewählter Wert bestimmt und jedem an der Aggregation beteiligten Knoten, ausser dem Aggregator selbst, mitgeteilt. Die Knoten nutzen diesen Wert dann als Seed zum Ziehen von Pseudozufallszahlen. Da mit dem gleichen Seed auch die gleichen Pseudozufallszahlen gezogen werden, treffen alle beteiligten Knoten immer die gleiche Entscheidung hinsichtlich einer Überprüfung des Aggregatorknotens. Ohne diesen Seed kann der Aggregatorknoten nicht herausfinden ob er im aktuellen Aggregationsvorgang überprüft wird oder nicht.

3.3 Nachrichten

Um die maximale Paketgröße im Sensornetzwerk möglichst gut auszunutzen, kann ein Knoten ein Datum an mehrere Empfänger auf einmal verschicken. Der erste Empfänger wertet den für ihn bestimmten Teil der Nachricht aus und sendet den Rest der Nachricht an den nächsten Empfänger weiter. Dies ist möglich, da ein Sensorknoten ausschliesslich Nachrichten an Knoten sendet, die auf seinem Aggregationspfad liegen und das Routing, entsprechend der Voraussetzungen in Abschnitt 2.1, entlang dieses Pfades erfolgt.

Um die Nutzdatengröße optimal auszunutzen, sendet ein Aggregator weiterhin nicht jede Nachricht sofort weiter, sondern sammelt die weiterzuleitenden Daten, bis er sämtliche Daten auf einmal weiterschicken kann. Dies ermöglicht bei gängigen Aggregationsfunktionen, wie etwa der Mittelwertbildung, eine deutliche Reduizierung des Kommunikationsaufwandes. Es muss allerdings beachtet werden, dass dies nicht generell so ist. Denkbar sind andere, komplexere Aggregationsfunktionen die von den im folgenden Beispiel angenommenen 32 bit pro Messwert abweichen.

Ein Beispiel im folgenden Abschnitt veranschaulicht das Vorgehen.

3.4 Ablauf

Sind p und k festgelegt, läuft ein Messvorgang wie folgt ab:

Jeder Blattknoten sendet sein Datum an seinen Elternknoten und an alle Zeugen des Elternknotens.

Jeder Aggregator a sammelt alle empfangenen Daten d_1, d_2, ...d_e seiner Kindknoten und berechnet sein Aggregat agg_a. Dieses leitet er an seinen eigenen Vaterknoten v weiter. Auch die Zeugen k_1, k_2, ... des Aggregators erhalten diese Daten und berechnen ebenfalls ein Aggregat agg_1, agg_2, ... Die Zeugen vergleichen anschliessend ihr berechnetes Aggregat mit jenem, dass sie von v erhalten. Schlägt ein Vergleich fehl, löst der jeweilige Zeuge Alarm aus und stellt seine Arbeit anschliessend ein.

Ist $p < 100\%$ dann wird, mit Hilfe des in Abschnitt 3.2.4 vorgestellten Mechanismus nur in $p\%$ aller Fälle eine Überprüfung durch Zeugen durchgeführt. In allen anderen Fällen wird eine normale Aggregation durchgeführt, die einen deutlich geringeren Kommunikationsaufwand erfordert.

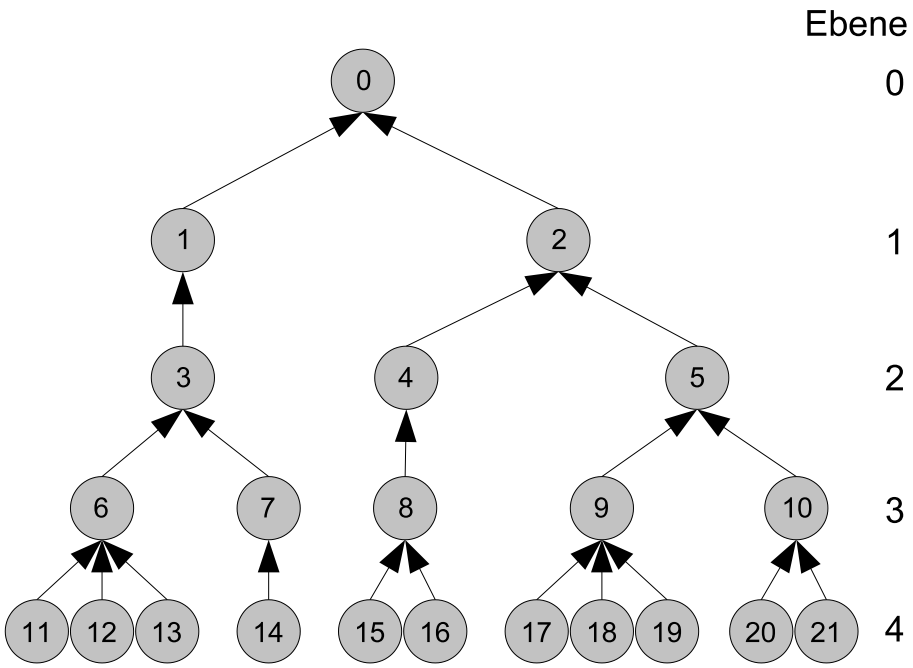


Abbildung 3.2: Beispielablauf einer sicheren Aggregation

von / an Knoten	Nachricht	Paketzahl
$11 \rightarrow 6$	$E_{11,6}(D_{11}, E_{11,3}(D_{11}), E_{11,1}(D_{11}))$	1
$12 \rightarrow 6$	$E_{12,6}(D_{12}, E_{12,3}(D_{12}), E_{12,1}(D_{12}))$	1
$13 \rightarrow 6$	$E_{13,6}(D_{13}, E_{13,3}(D_{13}), E_{13,1}(D_{13}))$	1
$14 \rightarrow 7$	$E_{14,7}(D_{14}, E_{14,3}(D_{14}), E_{14,1}(D_{14}))$	1
$15 \rightarrow 8$	$E_{15,8}(D_{15}, E_{15,4}(D_{15}), E_{15,2}(D_{15}))$	1
$16 \rightarrow 8$	$E_{16,8}(D_{16}, E_{16,4}(D_{16}), E_{16,2}(D_{16}))$	1
$17 \rightarrow 9$	$E_{17,9}(D_{17}, E_{17,5}(D_{17}), E_{17,2}(D_{17}))$	1
$18 \rightarrow 9$	$E_{18,9}(D_{18}, E_{18,5}(D_{18}), E_{18,2}(D_{18}))$	1
$19 \rightarrow 9$	$E_{19,9}(D_{19}, E_{19,5}(D_{19}), E_{19,2}(D_{19}))$	1
$20 \rightarrow 10$	$E_{20,10}(D_{20}, E_{20,5}(D_{20}), E_{20,2}(D_{21}))$	1
$21 \rightarrow 10$	$E_{21,10}(D_{21}, E_{21,5}(D_{21}), E_{21,2}(D_{21}))$	1

Tabelle 3.1: Nachrichten auf Ebene 4

von / an Knoten	Nachricht	Paketzahl
$6 \rightarrow 3$	$E_{6,3}(D_6, E_{6,1}(D_6), E_{6,0}(D_6), E_{11,3}(D_{11}), E_{11,1}(D_{11}),$ $E_{12,3}(D_{12}), E_{12,1}(D_{12}), E_{13,3}(D_{13}), E_{13,1}(D_{13}))$	3
$7 \rightarrow 3$	$E_{7,3}(D_7, E_{7,1}(D_7), E_{7,0}(D_7), E_{14,3}(D_{14}), E_{14,1}(D_{14}))$	2
$8 \rightarrow 4$	$E_{8,4}(D_8, E_{8,2}(D_8), E_{8,0}(D_8), E_{15,4}(D_{15}), E_{15,2}(D_{15}),$ $E_{16,4}(D_{16}), E_{16,2}(D_{16}))$	2
$9 \rightarrow 5$	$E_{9,5}(D_9, E_{9,2}(D_9), E_{9,0}(D_9), E_{17,5}(D_{17}), E_{17,2}(D_{17}),$ $E_{18,5}(D_{18}), E_{18,2}(D_{18}), E_{19,5}(D_{19}), E_{19,2}(D_{19}))$	3
$10 \rightarrow 5$	$E_{10,5}(D_{10}, E_{10,2}(D_{10}), E_{10,0}(D_{10}), E_{20,5}(D_{20}), E_{20,2}(D_2)$ $E_{21,5}(D_{21}), E_{21,2}(D_{21}))$	2

Tabelle 3.2: Nachrichten auf Ebene 3

von / an Knoten	Nachricht	Paketzahl
$3 \rightarrow 1$	$E_{3,1}(D_3, E_{3,0}(D_3), E_{6,1}(D_6), E_{6,0}(D_6), E_{7,1}(D_7),$ $E_{7,0}(D_7), E_{11,1}(D_{11}), E_{12,1}(D_{12}), E_{13,1}(D_{13}), E_{14,1}(D_{14}))$	3
$4 \rightarrow 2$	$E_{4,2}(D_4, E_{4,0}(D_4), E_{8,2}(D_8), E_{8,0}(D_8),$ $E_{15,2}(D_{15}), E_{16,2}(D_{16}))$	2
$5 \rightarrow 2$	$E_{5,2}(D_5, E_{5,0}(D_5), E_{9,2}(D_9), E_{9,0}(D_9),$ $E_{10,2}(D_{10}), E_{10,0}(D_{10}), E_{17,2}(D_{17}), E_{18,2}(D_{18}),$ $E_{19,2}(D_{19}), E_{20,2}(D_{21}), E_{21,2}(D_{21}))$	4

Tabelle 3.3: Nachrichten auf Ebene 2

von / an Knoten	Nachricht	Paketzahl
$1 \rightarrow 0$	$E_{1,0}(D_1, E_{3,0}(D_3), E_{6,0}(D_6), E_{7,0}(D_7))$	2
$2 \rightarrow 0$	$E_{2,0}(D_2, E_{4,0}(D_4), E_{5,0}(D_5), E_{8,0}(D_8), E_{9,0}(D_9), E_{10,0}(D_{10}))$	2

Tabelle 3.4: Nachrichten auf Ebene 1

Knoten	gesendete Pakete	Crypto-operationen	Energiekosten
0	0	10	13,0μAs
1	2	8	500,4μAs
2	2	13	506,9μAs
3	3	8	745,4μAs
4	2	5	496,5μAs
5	4	9	991,7μAs
6	3	6	742,8μAs
7	2	4	495,2μAs
8	2	5	496,5μAs
9	3	6	742,8μAs
10	2	5	496,5μAs
11-21	11	33	2737,9μAs
Summe	36	112	8965,6μAs

Tabelle 3.5: Energieaufwand für die Beispielmessung

Abbildung 3.2 zeigt beispielhaft einen Aggregationsbaum und veranschaulicht das Verfahren für $k = 2$ und $p = 100\%$. Die verschickten Nachrichten stellen die Tabellen 3.1 bis 3.4 dar. Dabei wurden einige Annahmen über Aufbau und Format der zu übertragenen Daten und der Datenpakete gemacht. Verwendet werden maximal 56 Byte grosse Pakete wie sie bei TinyOS [Berka] genutzt werden. Von den 56 Byte sind 29 Byte für Nutzdaten vorbehalten. Übertragen wird ein 32 Bit-Messwert. Jedes Datenpaket wird vor der Übertragung mit einem 64 Bit großen, symmetrischen Schlüssel codiert. Zur Berechnung der Gesamtkosten wurden Energieverbrauchsdaten der Mica II Plattform [Inco] verwendet. Der Versand eines kompletten Paketes kostet dort $245\mu As$, das Verschlüsseln von 64 Bit mit Hilfe von RC5 wird mit $1,3\mu As$ berücksichtigt [Berkb]. Nicht berücksichtigt sind die Kosten, die für die Aggregatberechnungen anfallen, da diese von der konkreten Aggregationsfunktion abhängen. Für typische Funktionen wie die Mittelwertberechnung fallen diese Kosten im Vergleich jedoch nicht ins Gewicht.

Tabelle 3.5 gibt Aufschluss über den notwendigen Energieaufwand für die Beispielmessung. Dabei wurde der Versand eines Paketes mit $245\mu As$ und das Chiffrieren und Dechiffrieren eines 64 Bit-Blocks mit RC5 mit jeweils $1{,}3\mu As$ veranschlagt. Diese Werte entsprechen denen der für Sensornetzwerke weit verbreiteten Hardwareplattform Mica2. Nicht beachtet werden die Kosten für die Durchführung der Aggregationen, da diese abhängig von der zum Einsatz kommenden Aggregationsfunktion sind. Insbesondere bei aufwendigen Aggregationsfunktionen können diese Kosten durchaus ins Gewicht fallen, da im Vergleich zu einer unsicheren Aggregation knapp das k-fache an Aggregationen berechnet werden muss.

3.5 Zusammenfassung

Das hier vorgestellte Verfahren ermöglicht durch die Nutzung von Zeugen eine sichere Aggregation mit den in Abschnitt 2.1 genannten Anforderungen, insbesondere der Unabhängigkeit von der Art der Aggregationsfunktion und das Erkennen von

Betrugsversuchen. Durch die Wahl der Systemparameter p und k kann das Verfahren individuell an den jeweiligen Einsatzzweck angepasst werden und somit ein optimaler Tradeoff zwischen Sicherheit und Energieeinsparung erreicht werden. Offen bleibt jedoch der Wunsch, bei einem erkannten Betrugsversuch den oder die korrumpierten Knoten zweifelsfrei zu identifizieren.

4. Evaluierung

Um das Potential von ESAWN bewerten zu können, wurden mit einer konkreten Implementierung von ESAWN Simulationsläufe durchgeführt und ausgewertet. In diesem Kapitel wird daher zuerst eine kurze Erläuterung der verwendeten Software und der getroffenen Randbedingungen gegeben. Anschließend werden die gewonnenen Daten präsentiert und ausgewertet.

4.1 GloMoSim als Simulationsumgebung

Die im Folgenden vorgestellten Werte und Daten sind Ergebnisse einer Implementierung des Verfahrens in GloMoSim [ZeBG98]. Dabei handelt es sich um eine skalierbare Simulationsumgebung für drahtlose und drahtgebundene Netzwerke die in Parsec, einer C-ähnlichen Programmiersprache, realisiert ist. Wie die meisten realen Netzwerke nutzt auch GloMoSim ein Schichtenmodell [Schu06] und trennt so z.B. Transport- von Netzwerkschicht.

Nach einigen Testläufen zeigte sich der enorme Rechnenaufwand der mit der detaillierten Simulation all dieser Schichten verbunden ist. Daher wurden durch einer Anpassung des GloMoSim-Quellcodes eine Beschränkung auf Anwendungs-, Transport- und Netzwerkschicht vorgenommen. Das so modifizierte Programm lief gut das zehnfache schneller als zuvor. Weiterhin war bei den neuen Simulationsergebnisse kein Unterschied zu den vorherigen feststellbar. Dies ist auch korrekt, da es bei ESAWN nur auf die Zahl der übertragenen Datenpakete, der Verschlüsselungsoperationen und Aggregationen für einen Messvorgang ankommt, da diese für die Energiekosten maßgeblich sind. Absolute Latenzzeiten oder Paketverlustraten sind nicht von Interesse.

Neben zahlreichen Simulationsläufen von ESAWN, dem hier vorgestellten Verfahren, wurden zur besseren Vergleichbarkeit auch Simulationen mit NOAG und AGGN durchgeführt.

NOAG ist ein Verfahren, dass keinerlei Aggregation verwendet, sondern jedes Datum verschlüsselt und einzeln zum Wurzelknoten leitet. Die Wurzel entschlüsselt alle Pakete und berechnet dann aus allen Werten ein Aggregat. Es ist offensichtlich, dass

ein Verfahren zur sicheren Aggregation nur dann sinnvoll ist, wenn die Energiekosten geringer sind, als die von NOAG. Ansonsten bringt die Aggregation keine Vorteile und es ist günstiger jedes Datum direkt zur Wurzel zu senden.

Bei AGGN handelt es sich um eine normale Aggregation die weder Zeugen noch eine Verschlüsselung verwendet. Dieses Verfahren bildet eine untere Schranke in Bezug auf den Kommunikationsaufwand. Kein Verfahren zur sicheren Aggregation kann weniger Energie kosten als diese Implementierung einer konventionellen unsicheren Aggregation.

Alle Simulationen wurden mehrfach in unterschiedlich aufgebauten Sensornetzen durchgeführt. Zu diesem Zweck ist ein in C geschriebenes Setup-Programm entstanden, dass nach Eingabe von Knotenzahl und dem Erwartungswert für den Verzweigungsgrad im Aggregationsbaum ein Sensornetzwerk mit den gegebenen Parametern in einem für GloMoSim lesbaren Format erzeugt.

Einige Simulationendaten sind aufgrund der langen Laufzeiten und der Vielzahl der Konfigurationsmöglichkeiten nicht aus Simulationsläufen der GloMoSim-Implementierung von ESAWN entstanden. Statt dessen wurde ein selbstgeschriebenes C-Programm verwendet, das in Verbindung mit dem oben erwähnten Setup-Programm zur Erstellung von Aggregationsbäumen Berechnungen an einer Vielzahl verschiedener Bäume durchführte. Um die Korrektheit dieser „Simulation“ sicherzustellen, wurden die Ergebnisse anschliessend mit der GloMoSim-Implementierung verglichen.

4.2 Energiekosten

Zuerst werden die Energiekosten betrachtet, die durch eine Verwendung von ESAWN im Vergleich zu den Referenzmodellen NOAG und AGGN entstehen.

4.2.1 Voraussetzungen

Für die Simulationsergebnisse wurden dieselben Annahmen über Größe und Format der Pakete sowie den Energieverbrauch getroffen, wie sie bereits in Abschnitt 3.4 Anwendung fanden.

4.2.2 Absolute Energiekosten

Die Diagramme in Abbildung 4.1 stellen die absoluten Energiekosten für einen vollständigen Messvorgang in Sensornetzwerken mit einem Erwartungswert von 2 und 6 bei dem Verzweigungsgrad im Aggregationsbaum dar. Unter einem vollständigen Messvorgang ist dabei die Übertragung und gleichzeitige Aggregation aller erfassten Messwerte in jedem Sensor bis hin zum Wurzelknoten des Aggregationsbaums zu verstehen.

Hier lässt sich zum einen erkennen, dass die Energiekosten von AGGN (dicke, untere Kurve) unabhängig von diesem Erwartungswert ist, denn hier schickt jeder Knoten genau ein Paket an seinen Vaterknoten im Aggregationsbaum. Wie der genau aufgebaut ist, spielt dafür keine Rolle. Zum anderen lassen sich die Mehrkosten ablesen, die der Einsatz zusätzlicher Zeugen mit sich bringt. Da jeder Knoten alle weiterzuleitenden Daten auf einmal verschickt und nicht für jedes Datum ein Paket versendet fällt der Mehraufwand jedoch geringer aus, als erwartet. Deshalb bringt eine Verdopplung der Zeugen keine Verdopplung der Kosten mit sich.

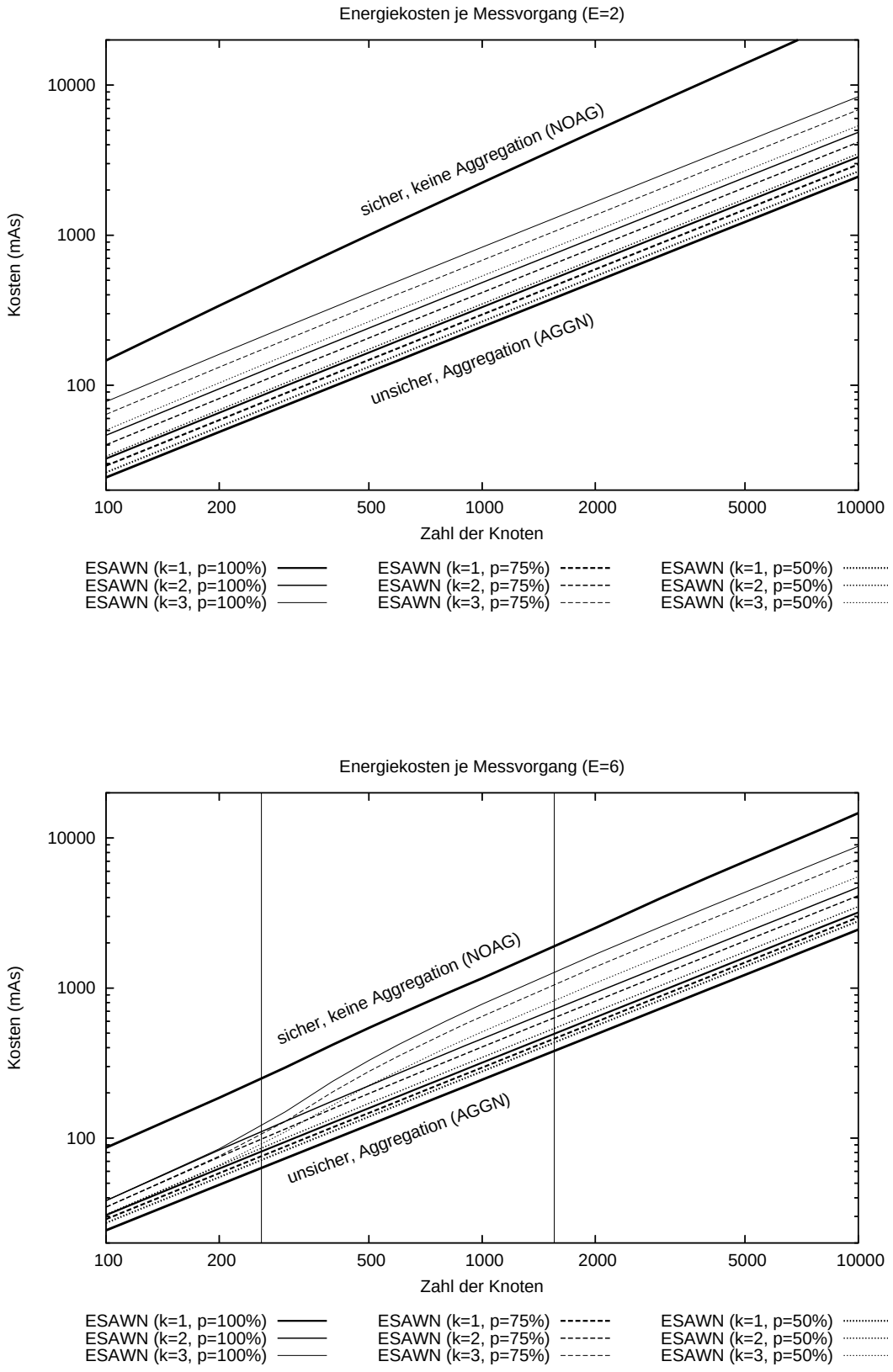


Abbildung 4.1: Energieverbrauch je vollständigem Messvorgang bei $E = 2$ und $E = 6$

Betrachtet man die obere dicke Kurve, die das Verfahren NOAG repräsentiert, fällt auf, dass in einem breiteren Aggregationsbaum bei gleicher Knotenzahl weniger Datenpakete versendet und somit geringere Energiekosten anfallen, als in einem schmalen. Das ist naheliegend, da bei einem breiten Baum die Kommunikationspfade kürzer sind. Stellt man die Ergebnisse nicht mehr absolut, sondern relativ zu NOAG dar, um die Einsparung gegenüber diesem Verfahren zu visualisieren, muss man dies berücksichtigen. Zwar ist die Einsparung relativ zu NOAG bei einem breiten Aggregationsbaum geringer als bei einem schmalen, trotzdem ist der breitere Aggregationsbaum absolut gesehen günstiger.

Der Grund für den überproportionalen Anstieg der Kurven für $k = 3$ im unteren Diagramm ergibt sich aus der Implementierung von ESAWN. Da einem Knoten nur Zeugen zugewiesen werden, die auf seinem Aggregationspfad Richtung Wurzel liegen, kann es bei kurzen Aggregationspfaden passieren, dass sich weniger Knoten auf dem Aggregationspfad befinden, als eigentlich Zeugen benötigt würden. In diesem Fall werden entsprechend weniger Zeugen zugewiesen.

Ein Beispiel macht dies deutlich. Betrachtet man einen Aggregationsbaum mit einem festen Verzweigungsgrad von $E = 2$, so können bis zu einer Knotenzahl von $n_2 = \sum_{i=0}^{h=3} 2^i = 15$ keinem Aggregatorknoten drei Zeugen zugewiesen werden, da alle Aggregationspfade kürzer sind. Ab einer Knotenzahl von $m_2 = \sum_{i=0}^{h=4} 2^i = 31$ können dagegen zumindest jedem Aggregator direkt unterhalb der Blattknotenebene drei unterschiedliche Zeugen zugewiesen werden. Aufgrund der Beschränkung auf Knotenzahlen von 100 bis 10000 ist dies jedoch nicht im Diagramm ersichtlich.

Anders dagegen bei einem Verzweigungsgrad von $E = 6$. Hier ist $n_6 = \sum_{i=0}^{h=3} 6^i = 259$ und $m_6 = \sum_{i=0}^{h=4} 6^i = 1555$. Diese Schwellen sind im unteren Diagramm klar erkennbar durch vertikale Linien markiert. Ab einer Knotenzahl von n_6 beginnt sich das jeweilige Verfahren mit $k = 3$ von dem mit $k = 2$ zu unterscheiden. Ab einer Zahl von m_6 verhalten sich die Kurven wieder proportional zur unteren Schranke. Die leichten Abweichungen von diesem Werten im Diagramm ergeben sich dadurch, dass die Aggregationsbäume in der Simulation zufällig erzeugt wurden und keine ausbalancierten Bäume darstellen.

4.2.3 Einsparungspotential der sicheren Aggregation

Ein wichtiges Kriterium für ein Aggregationsverfahren ist, wieviel Kommunikation durch die Aggregation im Vergleich zu keiner Aggregation (NOAG) erzielt wird. In den Diagrammen in den Abbildung 4.2 bis 4.6 sind die Paketzahlen deshalb in Relation zu NOAG dargestellt.

Betrachtet werden Netzwerke zwischen 100 und 10000 Knoten. Als Zahl der Zeugen wurden Simulationen mit $k = 1$, $k = 2$ und $k = 3$ durchgeführt und die Wahrscheinlichkeit einer Überprüfung mit $p = 100\%$, $p = 75\%$ und $p = 50\%$ angesetzt. Der Erwartungswert des Verzweigungsgrads im Aggregationsbaum wurde mit $E = 2$ bis $E = 6$ festgelegt.

Werte über 100% bedeuten, dass der entsprechende Durchlauf in Bezug auf die Kommunikationskosten im Mittel teurer ist als der vergleichbare Durchlauf ohne jede Aggregation. Werte unter 100% zeigen an, dass der Durchlauf im Mittel günstiger ist. Die jeweils unterste Kurve gibt die Kosten für AGGN, also normale Aggregation ohne zusätzliche Sicherheit, wieder. Je näher sich eine Kurve an dieser befindet,

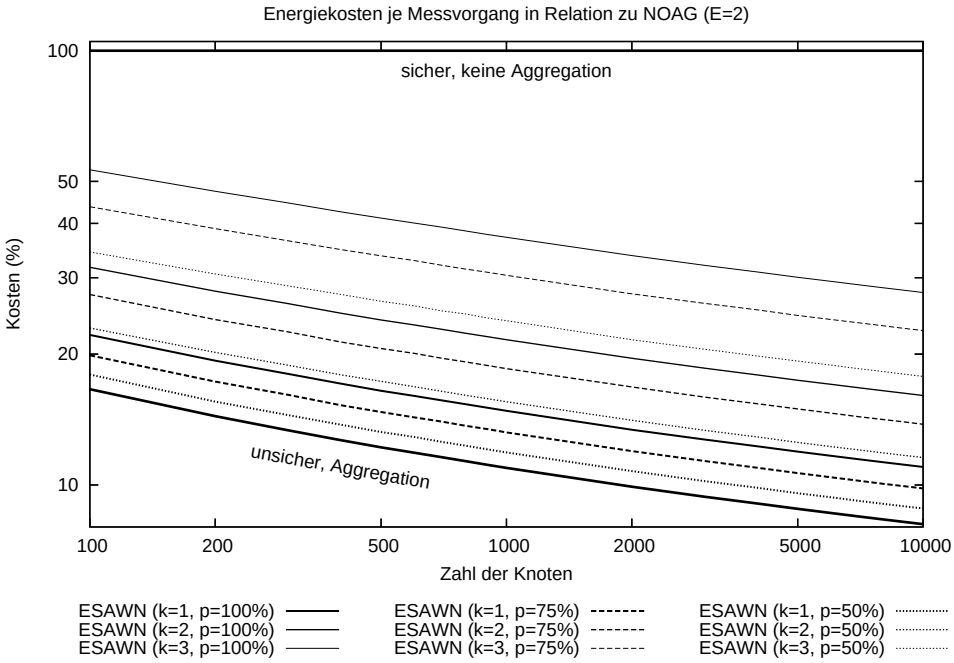


Abbildung 4.2: Energieverbrauch je vollständigem Messvorgang bei E=2

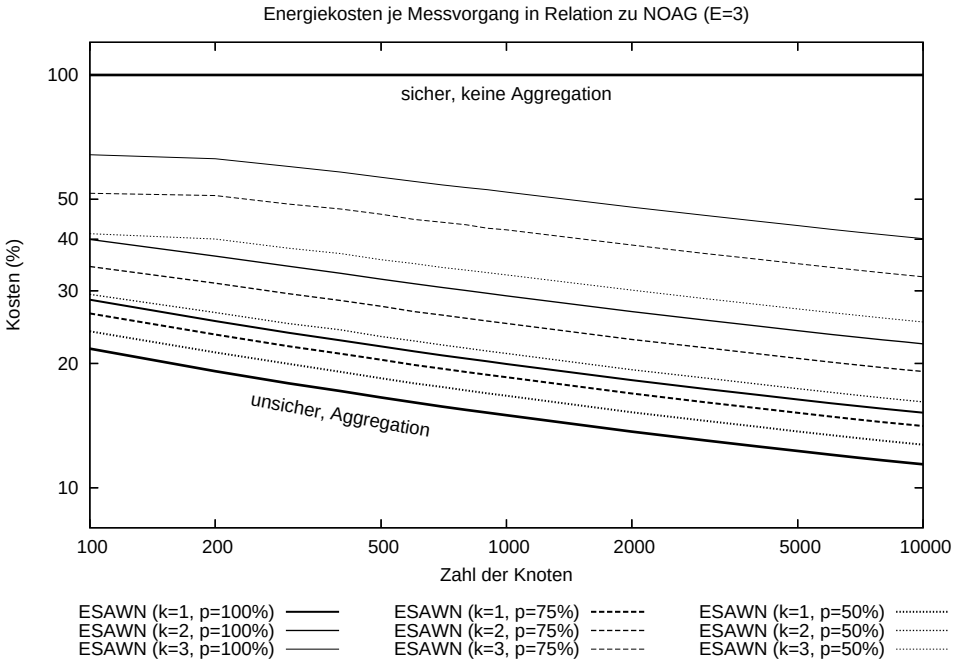


Abbildung 4.3: Energieverbrauch je vollständigem Messvorgang bei E=3

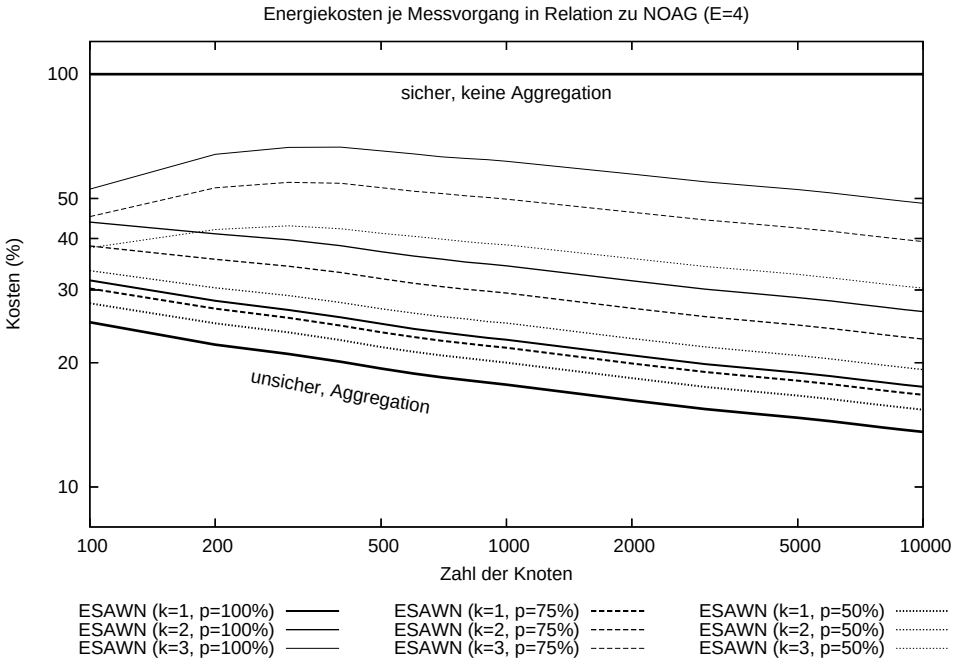


Abbildung 4.4: Energieverbrauch je vollständigem Messvorgang bei E=4

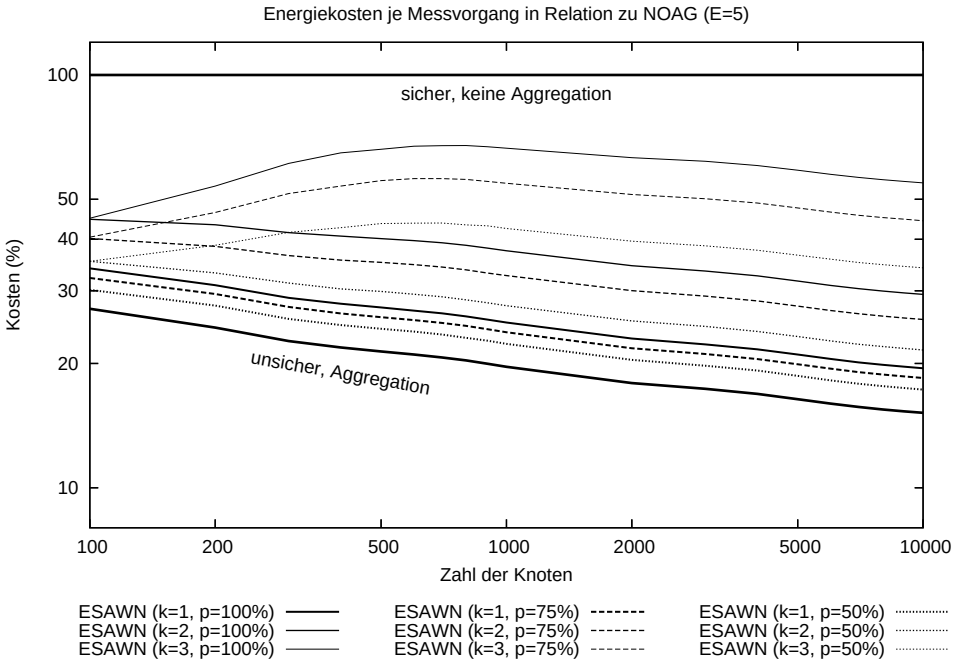


Abbildung 4.5: Energieverbrauch je vollständigem Messvorgang bei E=5

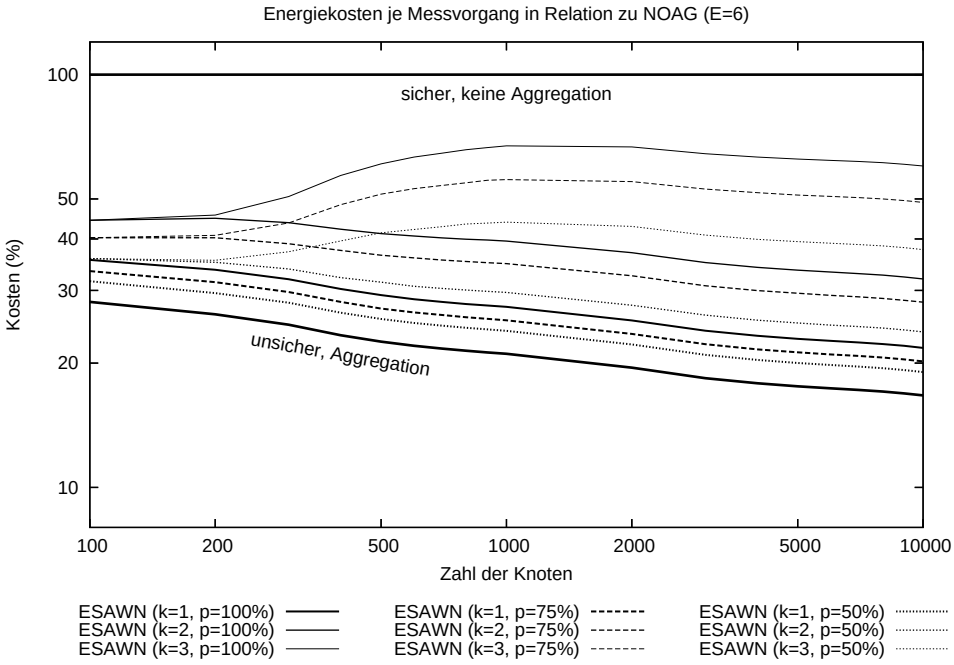


Abbildung 4.6: Energieverbrauch je vollständigem Messvorgang bei E=6

desto geringer ist der Overhead der durch die zusätzlichen Sicherheitsanforderungen entsteht.

Unabhängig von den Parametern E, k und p lässt sich erkennen dass alle Kurven streng monoton fallend sind, dass also die Ersparnis mit steigenden Netzwerkgrösse zunimmt. Eine Ausnahme bildet lediglich der mit großem E deutlich werdende Anstieg einiger Kurven im Bereich kleiner Netzwerkgrößen. Dieser Anstieg ist auf die konkrete Implementierung von ESAWN zurückzuführen, der bereits in Abschnitt 4.2.2 betrachtet wurde.

Die mit großer Netzwerkgröße zunehmende Ersparnis lässt sich mit der größer werdenden Höhe des Aggregationsbaum begründen. Ohne Aggregation müssen aufgrund der längeren Pfade zwischen Blattknoten und Wurzel deutlich mehr Pakete übertragen werden. Mit Aggregation – und dabei ist es prinzipiell egal ob sicher oder nicht – ist dies nicht der Fall. Je größer also das Netzwerk und je kleiner der Verzweigungsgrad des Aggregationsbaumes, desto größer ist die prozentuale Ersparnis durch Aggregation.

In den Diagrammen lassen sich ausserdem der nötige Mehraufwand für eine größere Zahl Zeugen ablesen (mit steigendem k) und auch das Einsparpotential, das eine Verminderung der Überprüfungswahrscheinlichkeit p mit sich bringt. Dabei muss beachtet werden, dass solch eine Verminderung die Authentizität des Ergebnisses beeinflussen kann. Näheres dazu in Abschnitt 4.3.

4.3 Wahrscheinlichkeit sicherer Aggregation

Bis jetzt offen gelassen wurde die Frage, mit welcher Wahrscheinlichkeit ein Aggregat in Gegenwart von b bösen Knoten einwandfrei und ohne erfolgreiche Manipulationen ist. Darüber sollen die Diagramme der Abbildungen 4.7 und 4.8 Aufschluß geben.

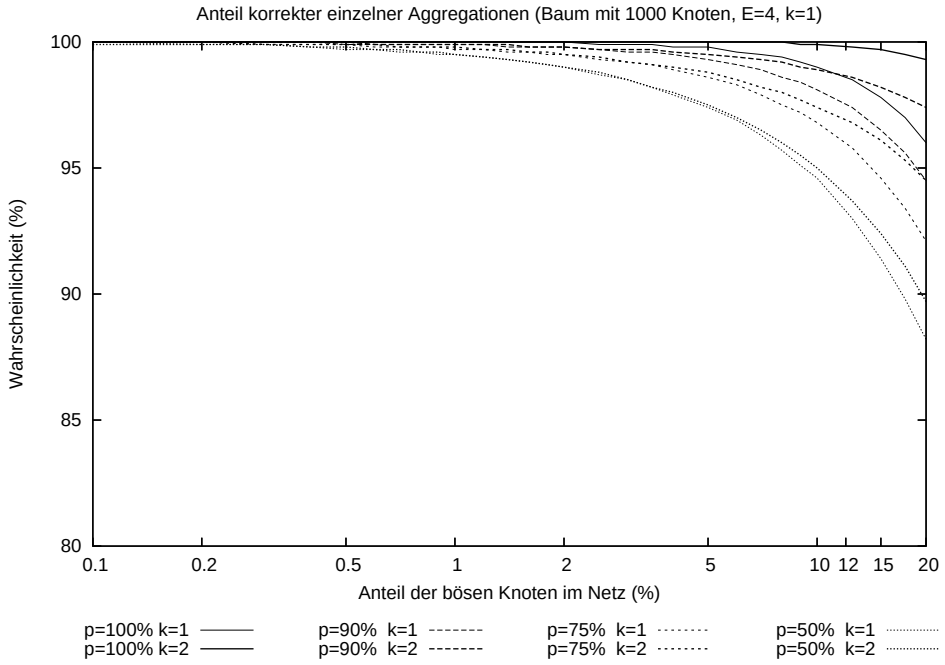


Abbildung 4.7: Wahrscheinlichkeit einzelner korrekter Aggregation

In dem Diagrammen in Abbildung 4.7 ist die Wahrscheinlichkeit dargestellt, mit der eine beliebige Aggregation innerhalb des Aggregationsbaums im Mittel authentisch ist. Es ist gut zu erkennen, dass die Wahrscheinlichkeit für eine einzelne korrekte Aggregation selbst bei eine größeren Anzahl böser Knoten sehr hoch ist.

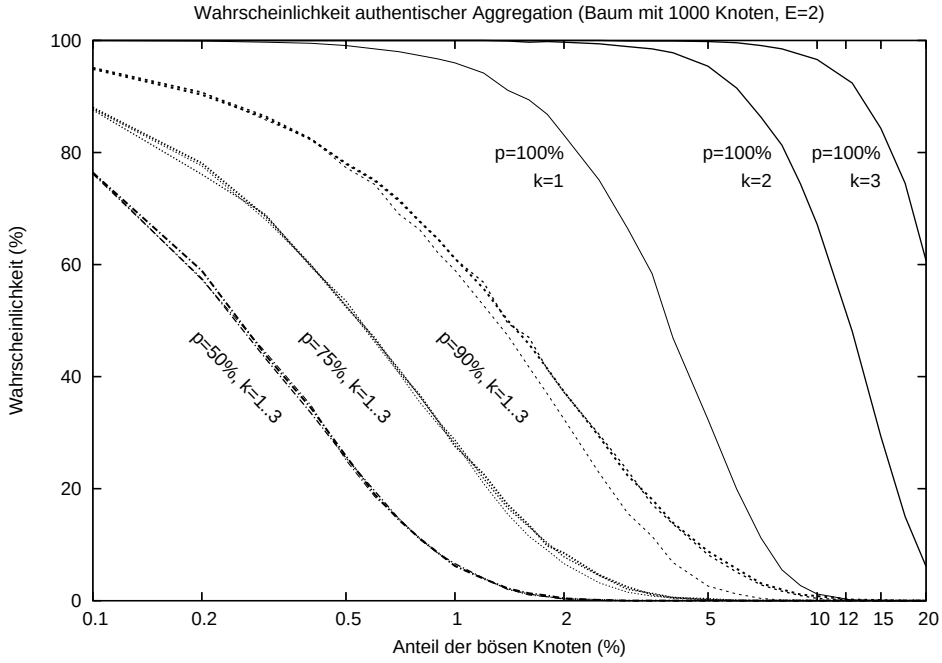
In dem Diagramm in Abbildung 4.8 ist die Wahrscheinlichkeit dargestellt, mit der der gesamte Aggregationsvorgang, das heisst sämtliche Aggregationen innerhalb des Aggregationsbaums, authentisch ist. Dies überrascht auf den ersten Blick, da die Korrektheitswahrscheinlichkeit einer einzelnen Aggregation doch sehr hoch ist, wie Abbildung 4.7 gezeigt hat. Es zeigt sich hier jedoch, dass durch die Vielzahl durchzuführender Aggregationen die Gesamtwahrscheinlichkeit deutlich niedriger ausfällt.

Anhand dieser Diagramme lässt sich sehr gut erkennen, welchen Einfluss die Reduzierung der Prüfwahrscheinlichkeit p auf die Gesamtauthentizität des Aggregationsvorgangs besitzt. Bereits bei der Reduzierung von $p = 100\%$ auf $p = 95\%$ ist ein deutlicher Abschlag erkennbar.

Ausserdem liegen die Kurven bei kleinem p für unterschiedliche k quasi aufeinander. Bei einer kleinen Prüfwahrscheinlichkeit hat also die Zahl der Zeugen praktisch keinen Einfluss mehr auf die Wahrscheinlichkeit korrekter Aggregation, denn die Tatsache, dass nur ein Aggregat nur selten geprüft wird, hat auf die Wahrscheinlichkeit korrekter Aggregation einen wesentlich größeren Einfluss als die Hinzunahme von weiteren Zeugen. Dies sollte bei der Wahl der Parameter berücksichtigt werden, da ein höheres k auch einen höheren Kommunikationsaufwand mit sich bringt.

4.4 Wahl der Parameter p und k

Während bisher die Parameter p und k immer vorgegeben waren und für verschiedene Kombinationen Kommunikationsaufwand und Wahrscheinlichkeiten korrekter Aggregation berechnet wurden, erfolgt jetzt die Betrachtung von der anderen Seite.



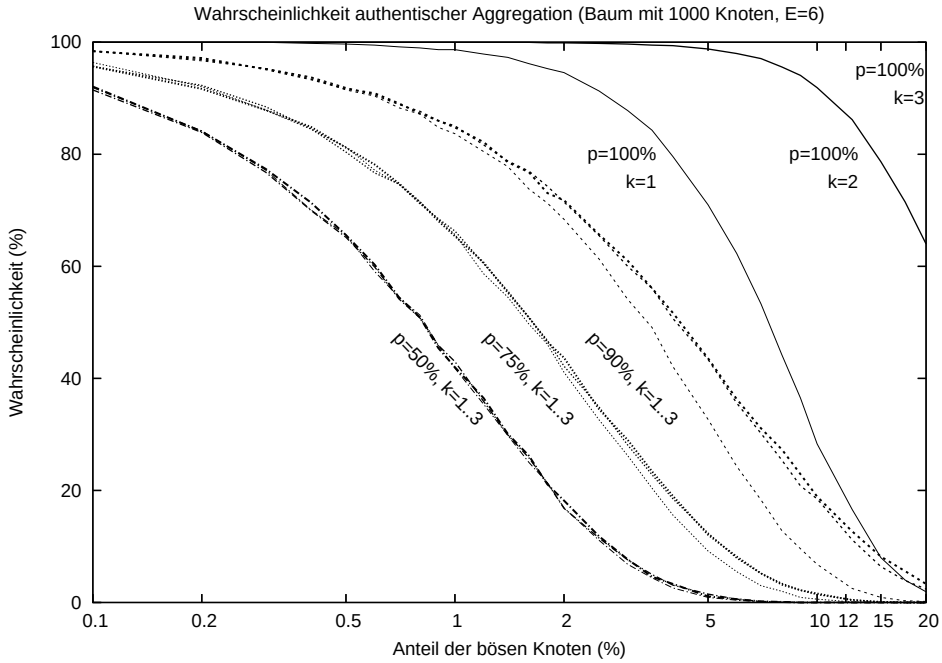


Abbildung 4.8: Wahrscheinlichkeit korrekter Aggregation mit n=1000 und E=2 bzw. E=6

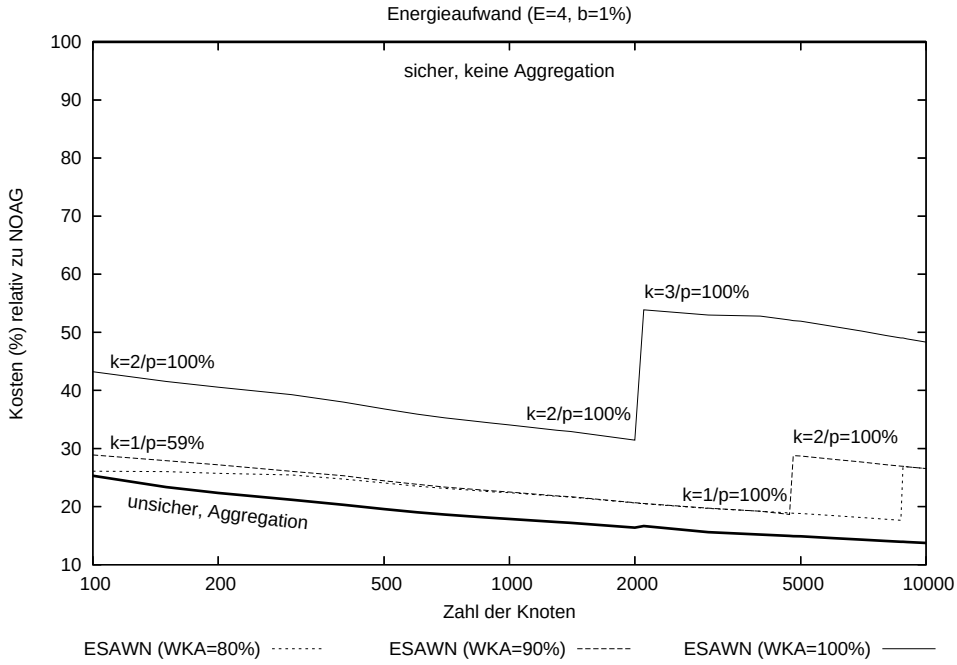


Abbildung 4.9: Energieaufwand für verschiedene Mindestwahrscheinlichkeiten bei E=4 und maximal 1% bösen Knoten

WKA	50%		60%		70%		80%		90%		100%	
Knoten	k	p	k	p	k	p	k	p	k	p	k	p
100	1	21	1	25	1	27	1	30	1	59	2	100
200	1	24	1	30	1	30	1	60	1	80	2	100
300	1	27	1	31	1	52	1	73	1	85	2	100
400	1	30	1	50	1	64	1	78	1	91	2	100
500	1	44	1	58	1	71	1	83	1	92	2	100
600	1	56	1	66	1	77	1	86	1	95	2	100
700	1	61	1	70	1	82	1	88	1	95	2	100
800	1	67	1	75	1	83	1	90	1	96	2	100
900	1	70	1	78	1	84	1	90	1	97	2	100
1000	1	74	1	78	1	87	1	93	1	97	2	100
2000	1	88	1	91	1	94	1	97	1	99	2	100
3000	1	92	1	95	1	97	1	98	1	100	3	100
4000	1	94	1	96	1	98	1	100	1	100	3	100
5000	1	96	1	97	1	98	1	100	2	100	3	100
6000	1	97	1	98	1	99	1	100	2	100	3	100
7000	1	97	1	98	1	100	1	100	2	100	3	100
8000	1	98	1	99	1	100	1	100	2	100	3	100
9000	1	99	1	99	1	100	2	100	2	100	3	100
10000	1	99	1	100	1	100	2	100	2	100	3	100

Tabelle 4.1: Parameterwahl zu den Simulationsreihen aus Abbildung 4.9

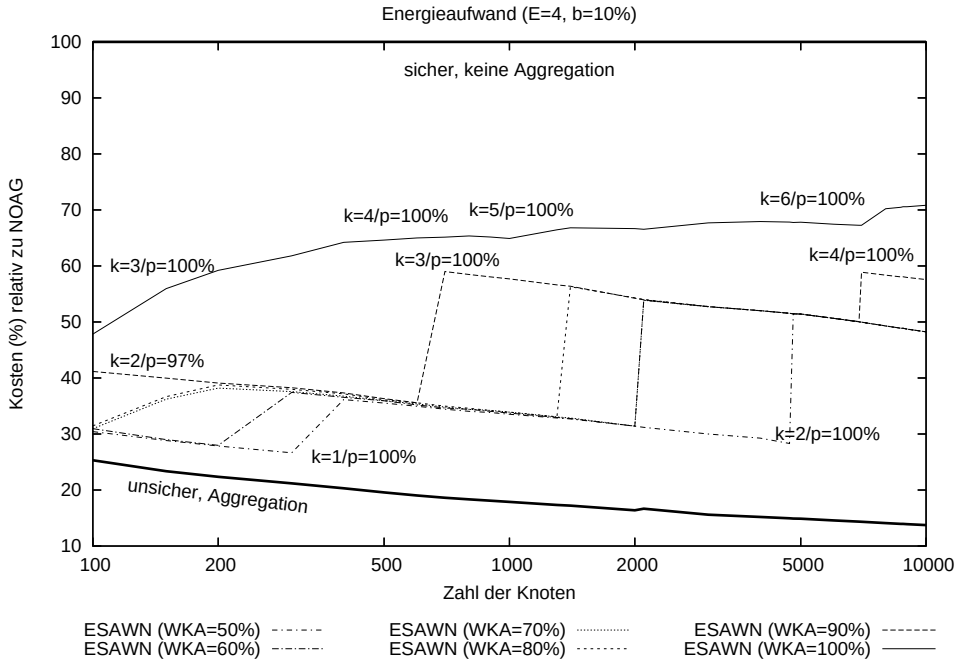


Abbildung 4.10: Energieaufwand für verschiedene Mindestwahrscheinlichkeiten bei E=4 und maximal 10% bösen Knoten

WKA	50%		60%		70%		80%		90%		100%	
Knoten	k	p	k	p	k	p	k	p	k	p	k	p
100	1	78	1	86	1	89	1	97	2	97	3	100
200	1	93	1	96	2	94	2	97	2	99	3	100
300	1	99	2	95	2	96	2	98	2	100	3	100
400	2	94	2	96	2	97	2	99	2	100	4	100
500	2	96	2	98	2	98	2	99	2	100	4	100
600	2	97	2	98	2	99	2	100	2	100	4	100
700	2	97	2	98	2	99	2	100	3	100	4	100
800	2	97	2	99	2	99	2	100	3	100	4	100
900	2	98	2	99	2	100	2	100	3	100	4	100
1000	2	98	2	99	2	100	2	100	3	100	4	100
2000	2	100	2	100	2	100	3	100	3	100	5	100
3000	2	100	3	100	3	100	3	100	3	100	5	100
4000	2	100	3	100	3	100	3	100	3	100	5	100
5000	3	100	3	100	3	100	3	100	3	100	5	100
6000	3	100	3	100	3	100	3	100	3	100	5	100
7000	3	100	3	100	3	100	3	100	4	100	5	100
8000	3	100	3	100	3	100	3	100	4	100	6	100
9000	3	100	3	100	3	100	3	100	4	100	6	100
10000	3	100	3	100	3	100	3	100	4	100	6	100

Tabelle 4.2: Parameterwahl zu den Simulationsreihen aus Abbildung 4.10

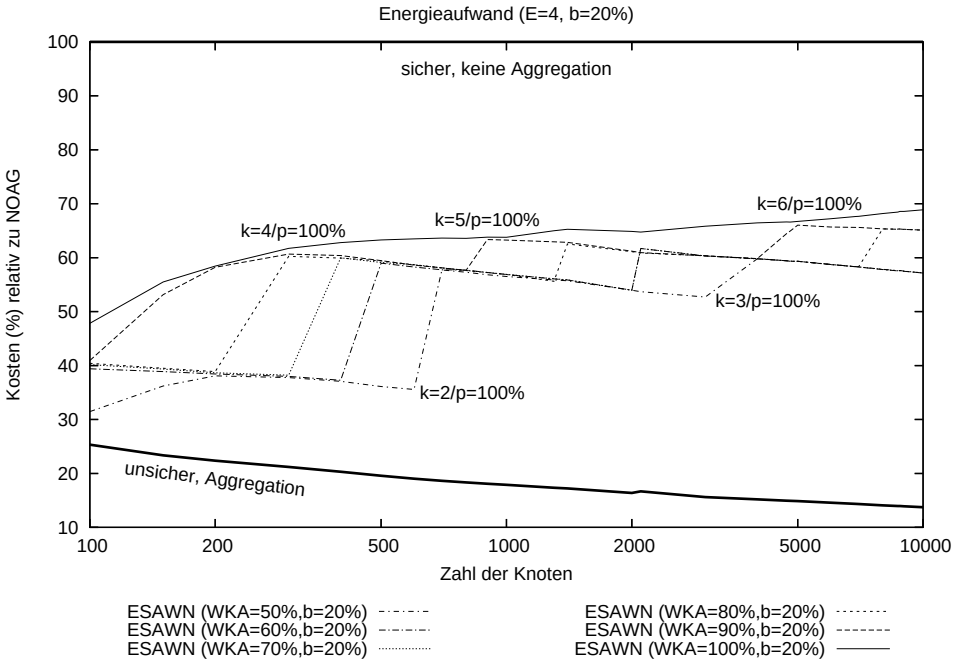


Abbildung 4.11: Energieaufwand für verschiedene Mindestwahrscheinlichkeiten bei E=4 und maximal 20% bösen Knoten

WKA	50%		60%		70%		80%		90%		100%	
Knoten	k	p	k	p	k	p	k	p	k	p	k	p
100	1	95	2	90	2	94	2	96	2	99	3	100
200	2	95	2	97	2	98	2	100	3	99	3	100
300	2	98	2	99	2	100	3	99	3	100	4	100
400	2	99	2	100	3	99	3	99	3	100	4	100
500	2	99	3	99	3	99	3	100	3	100	4	100
600	2	100	3	99	3	100	3	100	3	100	4	100
700	3	99	3	99	3	100	3	100	3	100	4	100
800	3	99	3	100	3	100	3	100	3	100	4	100
900	3	99	3	100	3	100	3	100	4	100	5	100
1000	3	99	3	100	3	100	3	100	4	100	5	100
2000	3	100	3	100	3	100	4	100	4	100	5	100
3000	3	100	4	100	4	100	4	100	4	100	5	100
4000	4	100	4	100	4	100	4	100	4	100	6	100
5000	4	100	4	100	4	100	4	100	5	100	6	100
6000	4	100	4	100	4	100	4	100	5	100	6	100
7000	4	100	4	100	4	100	4	100	5	100	6	100
8000	4	100	4	100	4	100	5	100	5	100	6	100
9000	4	100	4	100	4	100	5	100	5	100	6	100
10000	4	100	4	100	4	100	5	100	5	100	6	100

Tabelle 4.3: Parameterwahl zu den Simulationsreihen aus Abbildung 4.11

Abhängig vom konkreten Anwendungsfall wird ein Sensornetzwerk mit sicherer Aggregation benötigt, das mit einer bestimmten Mindestwahrscheinlichkeit ein korrektes Aggregat liefert. Eine korrekte Aggregation ist dabei wie folgt definiert: Kein Betrug konnte unbemerkt durchgeführt werden. Das resultierende Gesamtaggregat ist also korrekt und authentisch.

Im Folgenden soll diese Wahrscheinlichkeit korrekter Aggregation (WKA) vorgegeben werden. Ausserdem muss die Zahl der Angreifer spezifiziert werden, mit der höchstens zu rechnen ist. Aus diesen Angaben können die notwendigen Parameter p und k ermittelt werden. Dabei muss beachtet werden, dass es im Allgemeinen mehrere Möglichkeiten der Parameterwahl gibt, die die gleiche WKA garantieren. Erhöht man zum Beispiel k um eins, kann im Gegenzug dazu die Prüfwahrscheinlichkeit p gesenkt werden ohne dass die gewünschte WKA unterschritten wird. Von allen Möglichkeiten interessiert jedoch die, mit dem geringsten Energieverbrauch. Vor einer Erhöhung von k, also der Hinzunahme von eines weiteren Zeugen, sollte deswegen immer geprüft werden, ob der gleiche Effekt nicht durch eine Erhöhung von p erreicht werden kann. Das konkrete Vorgehen verdeutlicht folgender Pseudocode:

```
% Eingabe: Knotenzahl n,
%          Anteil böser Knoten b,
%          Wahrscheinlichkeit korrekter Aggregation WKA

k=1
p=1

do {
  if ( p < 100 )
    p++;

  else {
    p=1,
    k++
  }
  WKA' = secusim(n,b,k,p);
} while (WKA' < WKA)
```

Das Diagramm aus Abbildung 4.9 zeigt den Energieaufwand mit dem zu rechnen ist, um verschiedene Mindestwahrscheinlichkeiten für die Korrektheit von Aggregaten zu garantieren wenn mit höchstens einem Prozent böser Knoten zu rechnen ist. Deutlich erkennbar sind die deutlich höheren Kosten die eine Aggregatkorrektheit von 100% mit sich bringt. Die Kosten für niedrigere WKAs liegen dagegen verhältnismäßig dicht beisammen.

Auffällig ist darüber hinaus die Tatsache, dass die Kosten für Wahrscheinlichkeiten zwischen 50% und 90% im Bereich um 2000 Knoten kurzfristig die selben sind sowie die auffallend großen Sprünge der Kostenkurve an verschiedenen Stellen, etwa für WKA=100% bei 2000 Knoten. Diese lassen sich mit den Daten aus Tabelle 4.1 erklären. In dieser Tabelle sind die Parameter erfasst, die für eine bestimmte Netzkonfiguration gewählt werden müssen, damit die Wahrscheinlichkeit korrekter Aggregation mindestens WKA% beträgt.

Die Diagramme aus Abbildung 4.10 und 4.11 sowie die dazugehörigen Tabellen 4.2 und 4.3 zeigen die Ergebnisse bei höheren Anteilen böser Knoten.

4.5 Zusammenfassung

Die Ergebnisse zeigen, dass das hier vorgestellte Verfahren ESAWN zu nennenswerten Energieeinsparungen in Sensornetzen führt, ohne Sicherheitsaspekte zu vernachlässigen. Durch einen verminderten Authentizitätsbedarf können diese Einsparungen nochmals deutlich erhöht werden.

5. Zusammenfassung und Ausblick

Ziel dieser Studienarbeit war ein Verfahren zur sicheren Datenaggregation. Die Kommunikation sollte gegen passives Abhören und auch gegen aktive Manipulationsversuche geschützt werden und Betrugsversuche erkannt werden. Weiterhin sollte es keine Einschränkungen bezüglich der Aggregationsfunktionen geben.

Bereits bei der Analyse der Problemstellung zeigte sich die Gegensätzlichkeit dieser Anforderungen. Zum einen soll Datenaggregation die Kosten senken, zum anderen sollen jedoch Sicherheitsanforderungen erfüllt werden. Dies alles in einem Umfeld, das potentiellen Manipulationen durch physikalischen Zugriff offen steht und Angriffsmodelle zulässt, wie sie in konventionellen Netzwerken nicht betrachtet werden.

Es lag daher nahe, einen Tradeoff zwischen benötigter Sicherheit und erreichter Kosteneinsparung zu finden. ESAWN realisiert solch einen Tradeoff und erfüllt dabei die gestellten Anforderungen: Es bietet Flexibilität in Bezug auf Aggregationsfunktionen und realisierter Sicherheit. Betrugsversuche werden mit einstellbarer Wahrscheinlichkeit erkannt. Höhere Sicherheitsanforderungen führen dabei zu höheren Kosten.

Offen bleibt das Problem der Identifizierung korrumpierter Knoten. Schlägt ESAWN Alarm, lässt sich der oder die korrumpierten Knoten nicht zweifelsfrei identifizieren. Es ist sogar unklar, ob überhaupt eine Manipulation der Daten stattgefunden hat, oder ob es sich um einen Störversuch eines Angreifers handelte.

Klar ist, dass eine Änderung des Protokolls, die eine solche Identifizierung ermöglicht, mit zusätzlichem Kommunikationsaufwand und damit deutlich höheren Energiekosten verbunden ist. So bleibt auch hier die Frage nach einem möglichst effizienten Verfahren, das diesen Zusatzaufwand möglichst klein ausfallen lässt.

Literatur

[ABCL+98] Ross Anderson, Francesco Bergadano, Bruno Crispo, Jong-Hyeon Lee, Charalampos Manifavas und Roger Needham. A new family of authentication protocols. *SIGOPS Operating Systems Review*, 32(4), 1998, S. 9–20.

[AcGW05] Mithun Acharya, Joao Girão und Dirk Westhoff. Secure Comparison of Encrypted Data in Wireless Sensor Networks. In *3rd International Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks (WiOpt 2005), Trentino, Italy*. IEEE Computer Society, April 2005, S. 47–53.

[Berka] UC Berkeley. TinyOS, an open-source operating system designed for wireless embedded sensor networks. http://www.tinyos.net/.

[Berkb] UC Berkeley. TinySec: Link Layer Encryption for Tiny Devices. http://www.cs.berkeley.edu/~nks/tinysec/.

[BlZi06] Erik-Oliver Blaß und Martina Zitterbart. An Efficient Key Establishment Scheme for Secure Aggregating Sensor Networks. In *ACM Symposium on Information, Computer and Communications Security*, März 2006.

[CaMT05] Claude Castelluccia, Einar Mykletun und Gene Tsudik. Efficient Aggregation of Encrypted Data in Wireless Sensor Networks. In *Proceedings of the ACM/IEEE Mobiquitous Conference*, Juli 2005.

[DiFo04] Tassos Dimitriou und Dimitris Foteinakis. Secure In-Network Processing in Sensor Networks. In *IEEE BASENETS, San Francisco*, 2004.

[EGHK99] Deborah Estrin, Ramesh Govindan, John S. Heidemann und Satish Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, August 1999, S. 263–270.

[HuEv03] Lingxuan Hu und David Evans. Secure Aggregation for Wireless Networks. In *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, Washington, DC, USA, 2003. IEEE Computer Society, S. 384.

[IEGH01] C. Intanagonwiwat, D. Estrin, R. Govindan und J. Heidemann. Impact of network density on data aggregation in wireless sensor networks.

In *Proceedings of International Conference and Distributed Computing Systems*, November 2001.

[Inco] Crossbow Technology Incorporated. MICA, MICA2 Motes und Sensoren. http://www.xbow.com/de/Funksensor-Netzwerk.htm.

[KaKP00] Joseph M. Kahn, Randy Howard Katz und Kristofer S. J. Pister. Emerging Challenges: Mobile Networking for „Smart Dust“. *J. Comm. Networks*, September 2000, S. 188–196.

[oEAS] Division of Engineering und Harvard University Applied Sciences. CodeBlue: Wireless Sensor Networks for Medical Care. http://www.eecs.harvard.edu/~mdw/proj/codeblue/.

[PeSW04] Adrian Perrig, John Stakovic und David Wagner. Security in Wireless Sensor Networks. *Communications of the ACM*, 47(6), Juni 2004, S. 53–57.

[PrSP03] Bartosz Przydatek, Dawn Song und Adrian Perrig. SIA: secure information aggregation in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, NY, USA, 2003. ACM Press, S. 255–265.

[PSWC+01] Adrian Perrig, Robert Szewczyk, Victor Wen, David E. Culler und J. D. Tygar. SPINS: security protocols for sensor networks. In *Mobile Computing and Networking*, 2001, S. 189–199.

[Qadi] Ala Qadi. Object Tracking Using Sensor Networks (Report). http://cse.unl.edu/~sdas/WSNL/report.pdf.

[RSPS02] Vijay Raghunathan, Curt Schurgers, Sung Park und Mani B. Srivastava. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine*, 19(2), März 2002, S. 40–50.

[Saus05] Rosario C. Sausa. Real-time, laser-based sensors for military and civilian applications. U.S. Army Research Laboratory, 2005.

[Schu06] Christian Schulze. IT-System-Elektroniker Handbuch – Hilfe bei der IT-Ausbildung: Das ISO-OSI-Referenzmodell. http://www.itse-guide.de/artikel/iso-osi-referenzmodell, 2006.

[WeGS06] Dirk Westhoff, Joao Girao und Markus Schneider. Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks. In *Proceedings of the 2006 International Conference on IEEE Transactions on Mobile Computing*, 2006.

[WLLP01] Brett Warneke, Matt Last, Brian Liebowitz und Kristofer S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *Computer*, 34(1), 2001, S. 44–51.

[ZeBG98] Xiang Zeng, Rajive Bagrodia und Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *Workshop on Parallel and Distributed Simulation*, 1998, S. 154–161.

[ZhSJ03] Sencun Zhu, Sanjeev Setia und Sushil Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. In *ACM Conference on Computer and Communications Security (CCS '03)*, Oktober 2003.

[ZSJN04] Sencun Zhu, Sanjeev Setia, Sushil Jajodia und Peng Ning. An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks. In *Proceedings of IEEE Symposium on Security and Privacy, Oakland, California*, Mai 2004.