University of Applied Sciences
Hochschule für Technik und Wirtschaft
Dresden
Fakultät Informatik/Mathematik

Masterarbeit

Konzipierung einer Sicherheitsarchitektur für drahtlose Sensornetze mit 6LoWPAN in der Hausautomatisierung

Angelos Drossos

Dezember 2013

im Masterstudiengang

Angewandte Informationstechnologien
Intelligente Informations- und Kommunikationstechnologien

Betreuer und Erstgutachter

Prof. Dr.-Ing. Jörg Vogt
University of Applied Sciences Dresden
Rechnernetze / Kommunikationssysteme

Zweitgutachter

Prof. Dr.-Ing. Robert Baumgartl
University of Applied Sciences Dresden
Betriebssysteme

Eidesstattliche Erklärung

Masterarbeit zum Thema	*Konzipierung einer Sicherheitsarchitektur für drahtlose Sensornetze mit 6LoWPAN in der Hausautomatisierung*
vorgelegt von	B. Sc. Angelos Drossos
zur Erlangung des Grades	„Master of Science (M. Sc.)“

Ich versichere, dass ich die Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Daher versichert die selbstständige und eigenhändige Anfertigung an Eides statt …

Dresden, 6. Dezember 2013, Angelos Drossos

Anmerkungen:

(1) Aus Gründen der Übersichtlichkeit und Lesbarkeit wurde in dieser Arbeit auf eine geschlechterspezifische Schreibweise verzichtet. Die gewählte männliche respektive weibliche Formulierung ist in diesem Sinne geschlechtsneutral zu verstehen.

(2) Ebenfalls aus Gründen der Übersichtlichkeit und Lesbarkeit wurde in dieser Arbeit keine Trennung zwischen Haus und Wohnung vollzogen. Wenn von Hausbewohnern gesprochen wird, sind folglich auch Bewohner von Wohnungen gemeint (und umgekehrt).

Danksagungen

Ich möchte mich bei Prof. Dr.-Ing. Jörg Vogt für die Betreuung der Masterarbeit bedanken, die aufgrund des vorangehenden Forschungsprojekt hervorgegangen ist. Das Themengebiet der Hausautomatisierung bietet sehr viele offene Fragen, die es zu beantworten gilt. Die Diskussionen zu diesen Fragen haben Spaß bereitet und veranlassen dazu, sich noch über die Masterarbeit hinaus mit diesem Themengebiet auseinander zu setzen.

Weiterhin möchte ich der Familie Siegel – insbesondere Madlen, Kerstin, Jörg und Marcel – für die liebevolle Unterstützung während der Erstellung der Masterarbeit danken. Sie war eine große Hilfe. Das betrifft nicht nur die Zeit während der Masterarbeit, sondern bezieht sich auf die komplette Studiumszeit.

Auch möchte ich meinen Eltern und meiner Verwandtschaft danken, die mich während des Studiums begleitet haben. Insbesondere aber gilt ein großer Dank meiner Mutter, Jeannette Drossos, die es ermöglicht hat, dass ich auf die Oberschule wechseln konnte, wo ich eine gute Schulausbildung bekam und meine Informatiklaufbahn begann.

Abstract

The wireless, internet-ready and open home automation is a expanding market segment in this day and age. Occupants needs to feel gemütlich in their home. They cannot agree to an illegal monitoring of their house. If the home automation system has no security, then the consumer acceptance is at risk.

Therefore, this masters thesis has the goal to develop a security architecture that can be used in a wireless, internet-ready and open home automation scenario. 6LoWPAN is used so that the home automation system becomes ready for the internet of things.

But before a security architecture can be developed, an analysis of home automation systems is needed. In this analysis, the behavior of communication is interesting. After the selection of one home automation system, secure communications in the internet of things is discussed.

With this discussion in mind, the security architecture needs to be designed. This is made up of three steps: First, threats are analysed. Second, these threats lead to security specifications for the home automation system. In the last step, security mechanisms and key management is discussed.

Kurzzusammenfassung

Die drahtlose, internetfähige und offene Hausautomatisierung ist ein zur Zeit expandierendes Marktsegment. Hausbewohner möchten sich in ihrem Wohnhaus oder ihrer Wohnung heimisch fühlen. Ein unerlaubtes Überwachen des Hauses kann folglich nicht vom Hausbewohner akzeptiert werden. Bietet die Hausautomatisierungsanlage auf diesem Gebiet keine Sicherheit, so ist die Akzeptanz dieser Anlage gefährdet.

Aus diesem Grund beschäftigt sich diese Masterarbeit mit der Konzipierung einer Sicherheitsarchitektur für eine drahtlose, internetfähige und offene Hausautomatisierung. Dabei wird die Internetfähigkeit durch ein drahtloses Sensornetz mit 6LoWPAN sichergestellt. Diese Internetfähigkeit ermöglicht die Verwendung von Anwendungen zum Internet der Dinge.

Damit eine Sicherheitsarchitektur entworfen werden kann, ist eine Analyse von Hausautomatisierungssystemen erforderlich. Diese betrachtet die Kommunikation in den zu analysierenden Systemen, um dadurch ein geeignetes System zu finden. Danach werden Aspekte zur sicheren Kommunikation im Internet der Dinge diskutiert, um diese bei der Konzepterstellung miteinzubeziehen.

Die Konzepterstellung gliedert sich dabei in drei Stufen: Zuerst wird eine Analyse möglicher Gefahren aufgestellt. Dann folgen daraus Sicherheitsanforderungen. Zuletzt werden sinnvolle Sicherheitsmechanismen betrachtet sowie ein mögliches Schlüsselmanagement in diesem Hausautomatisierungssystem vorgeschlagen. Ziel des Schlüsselmanagements ist es dabei, dass das System sich so gut wie möglich selbst konfigurieren kann. Dadurch wird der Endverbraucher so wenig wie möglich in den Prozess, der die Sicherheit der Daten gewährleistet, integriert.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

In the 21st century the technology revolution will move into the everyday, ***the small and the invisible****.*
—Mark Weiser (1952 – 1999)

1.1. Motivation für die Umsetzung einer drahtlosen Hausautomatisierung

Bislang haben hohe Produktpreise sowie der aufwändige Einbau in Wohnungen und Wohnhäusern dazu geführt, dass die Installation einer Hausautomatisierungsanlage nicht durchgeführt wurde. Die Alternative ist sie bereits beim Bau des Wohnhauses einzuplanen. Dadurch steigen allerdings die Kosten des Wohnprojekts, wodurch solche Vorhaben nur auf speziellem Wunsch der Bauherrn veranlasst werden. Daher konnten bislang nur vereinzelte Bewohner den Komfort genießen, den eine Hausautomatisierungsanlage[1] bieten kann.

Solche Anlagen sind derzeit meist drahtgebunden und müssen beim Einbau speziell an die Voraussetzungen des Wohnobjekts angepasst werden. Ein Umzug ist mit neuen Kosten verbunden und zudem höchst unflexibel — ein erheblicher Nachteil demzufolge.

In den letzten Jahren haben fallende Preise für drahtlose Sensornetze dafür gesorgt, dass diese auch in der Hausautomatisierung an Attraktivität gewonnen haben. Eine drahtlos kommunizierende Hausautomatisierungsanlage hat den Vorteil, dass der Aus- und Einbau vergleichsweise kostengünstig ist. Zudem können solche Anlagen auch auf einfache Weise in Mietswohnungen eingesetzt werden, ohne den Vermieter einbinden zu müssen. Daher möchte man seit der Entwicklung der drahtlosen Hausautomatisierung erreichen, dass diese für jeden Hausbewohner attraktiv wird und flexibel eingesetzt werden kann.

[1] Ein Haus oder eine Wohnung, die eine Hausautomatisierungsanlage besitzt, wird oft auch als „Smart Home“ oder „intelligentes Haus“ bezeichnet.

Die drahtlose Hausautomatisierung befindet sich allerdings noch in der Entwicklungsphase, obgleich bereits erste Produkte[2] auf dem Markt existieren. Denn die Hausautomatisierung wird immer stärker von der stetigen Entwicklung der Internettechnologien beeinflusst. Dadurch wird es in Hausautomatisierungssystemen möglich, dass verschiedenste Geräte über neuentwickelte Technologien miteinander kommunizieren.

Den Laufweg zum Lichtschalter zu ersparen und das Licht fernzusteuern, repräsentiert nicht den eigentlichen Komfort, der durch ein Hausautomatisierungssystem erreicht werden kann. Vielmehr möchte ein Hausbewohner möglichst wenig bzw. *unbewusst* mit dem Hausautomatisierungssystem in Kontakt treten. Erst dann ist der Einsatz dieses Systems gerechtfertigt. Dies hat bereits der Informatiker und Kommunikationswissenschaftler *Mark Weiser* im letzten Jahrhundert (1991) erkannt, als er behauptete, im 21. Jahrhundert bewirke die Revolution der Technologie, dass sie in den Alltag verschwinden werde, klein und unscheinbar.

Diese Vision entwickelt sich etwa 20 Jahre später und lässt ein aktuell expandierendes Marktsegment im Bereich der drahtlosen Hausautomatisierung entstehen.

1.2. Internetfähige Hausautomatisierung

Dieses expandierende Marktsegment fasziniert viele Forscher. Denn es gibt eine Reihe von komplizierten, teils kontroversen Problemen zu lösen. Insbesondere wurde früher im Bereich der drahtlosen Sensornetze für jedes Einsatzgebiet[3] eine eigenständige Lösung gesucht. Entwickler von Internettechnologien dagegen versuchen Standards zu schaffen, die offen und zukunftsorientiert sind. Forscher haben erkannt, dass dieses Denken auch in der drahtlosen Hausautomatisierung nötig ist, wenn diese internetfähig sein möchte. So ist die ZigBee Alliance ein gutes Beispiel für dieses Umdenken: Der ZigBee Standard galt lange als einer der besten Industriestandards für Funksysteme. Seit diesem Jahr wurde ZigBee IP verabschiedet, der nun auf Internettechnologien (wie IP, TCP und UDP) basiert. Das Marktsegment hat sich folglich weiterentwickelt und eine drahtlose, *internetfähige* Hausautomatisierung wird nun angestrebt.

Auch die Fakultät Informatik / Mathematik der University of Applied Sciences Dresden sowie ihre Studenten betreiben aktive Forschung zu diesen Themengebieten. Vor allem ist die Anwendung von Sensornetzen in der drahtlosen, internetfähigen und *offenen* Hausautomatisierung ein wichtiges Thema, wobei die offene Hausautomatisierung einen Schritt weiter geht und neben den offenen Internetstandards auch offene Systeme (Systeme ohne Lizenzgebühren) erfordert, um die Mehrkosten, die bei Anschaffung der Hausautomatisierungsanlage entstehen, zu reduzieren.

[2] Beispiele für Produkte in der drahtlosen Hausautomatisierung sind Produkte basierend auf FS20, HomeMatic oder EnOcean.

[3] Einsatzgebiete von drahtlosen Sensornetzen reichen von der Medizin, dem Katastrophenschutz, der Feuerwehr, bis hin zum Militär. Weitere Informationen folgen in Unterabschnitt 2.1.3.

1.3. Hintergrund und Zielsetzung

Aus einem vorangehenden Forschungsprojekt an der University of Applied Sciences Dresden ist diese Masterarbeit entstanden. Denn drahtlose Sensornetze müssen nicht nur einen geringen Energieverbrauch, sondern auch gewisse Sicherheitsaspekte aufweisen. Das folgende Beispiel soll diese Problematik erklären.

Bewegungsmelder werden heute bereits eingesetzt, um das Licht im Flur automatisch an- und auszuschalten. Da es oft nur zum Durchgehen gebraucht und daher kurz ein- und ausgeschaltet wird, ist der Gang zum Lichtschalter oft überflüssig. Dies gilt insbesondere für Wohnungen, in denen es nur ein Flurlichtschalter gibt.

Bewegungsmelder sind aber nicht nur zur Lichtsteuerung nützlich. Sie helfen auch bei der Steuerung der Heizung. Viele günstige Temperaturthermostate haben den Nachteil, dass lediglich eine zeitliche Steuerung zum Einsatz kommt und im Urlaub ein Urlaubsmodus eingestellt werden muss. Vergisst man diesen, wird unnötig Energie verbraucht. Ein Bewegungsmelder im Hausautomatisierungssystem erkennt die Bewegungen und kann bei längerer Abwesenheit in den Urlaubsmodus wechseln.

Ein weiteres interessantes Einsatzgebiet von Bewegungsmeldern ist bei der Detektion von Einbrüchen. Auch wenn diese nicht verhindert werden, so kann die genaue Uhrzeit der Polizei bei ihren Ermittlungen erheblich helfen.

Integriert man den Bewegungsmelder im selben Gerät wie die Steuerung und ist dieses Gerät nicht internetfähig, so bestehen keine Sicherheitsbedenken. Doch in der drahtlos kommunizierenden und internetfähigen Hausautomatisierung ist diese Voraussetzung nicht erfüllt.

Einbrecher können diese Bewegungsdaten missbrauchen, um zu erkennen, ob sich jemand in der Wohnung befindet oder nicht. Hierzu wird lediglich ein vom Einbrecher geschickt platziertes Gerät benötigt, das die (unverschlüsselte) Kommunikation belauscht. Dieses Gerät muss nicht viel teurer sein als jedes andere Gerät im Hausautomatisierungssystem. Werden diese Bewegungsdaten über längere Zeiträume unbefugt beobachtet, ergeben sich detaillierte Bewegungsmuster, welche einen Einbruch erleichtern können.

Weiterhin müssen spezielle gesetzliche Vorgaben erfüllt werden, wenn Bewegungsdaten Personen zugeordnet werden können. So ist auch der Schutz dieser Daten per Gesetz erforderlich, um die Privatsphäre der Hausbewohner zu schützen.

Aktuelle Vorfälle im Skandal der National Security Agency (NSA) der USA um Edward Snowden[4] haben gezeigt, dass *Vertrauen auf Nicht-Missbrauch* in der heutigen, internetorientierten Gesellschaft nicht ausreicht, um solche Daten zu schützen.

Aus diesen Gründen ist der „Schutz von Daten“ ein ernst zu nehmendes, nicht zu vernachlässigendes Thema. Auch die drahtlose, internetfähige Hausautomatisierung ist hiervon nicht befreit. Die Gewährleistung der Sicherheit in drahtlosen Sensornetzen

[4]Nachrichten zu diesem brisanten Thema finden sich beispielsweise auf Golem.de, unter der Internetadresse `http://www.golem.de/specials/nsa/`.

ist in der Forschung ein viel umstrittenes Problem, da Lösungen sich normalerweise negativ auf den Energieverbrauch auswirken. Ein drahtloses Sensornetz kämpft nicht nur mit typischen Sicherheitsproblemen, die seit Jahrzehnten in der Netzwerksicherheit bekannt sind und bekämpft werden. Die Beschaffenheit drahtloser Sensornetze erzeugen zusätzliche Probleme in der Netzwerksicherheit. Der Grund hierfür liegt in der Ressourcenarmheit der eingesetzten Geräte, die allerdings notwendig ist, um einen geringen Energieverbrauch zu erreichen. Diese Ressourcenarmheit erschwert das Finden einer optimalen Lösung.

Daher setzt sich diese Masterarbeit das Ziel, eine Sicherheitsarchitektur für drahtlose Sensornetze in der Hausautomatisierung zu finden, die einen ausreichenden Schutz der Daten bietet und dennoch den Anforderungen an eine drahtlose, internetfähige und offene Hausautomatisierung sowie den Anforderungen an das zugrunde liegende Sensornetz gerecht wird.

2 Voraussetzungen und Anforderungen an die Hausautomatisierung

*Next comes ubiquitous computing, or the **age of calm technology**, when technology recedes into the background of our lives.*

—Mark Weiser (1952 – 1999)

Zur Findung einer Sicherheitsarchitektur in der Hausautomatisierung werden in diesem Kapitel Voraussetzungen und Anforderungen erläutert. Es ist erforderlich, dass diese bekannt sind, wenn Hausautomatisierungssysteme in Kapitel 3 analysiert werden. Aber auch bei der Konzipierung der Sicherheitsarchitektur in Kapitel 5 müssen sie berücksichtigt werden.

2.1. Voraussetzungen

Nachdem oft der Begriff Hausautomatisierung gefallen ist, sollen zuerst die Aufgaben der Hausautomatisierung vorgestellt und von den Aufgaben der Gebäudeautomatisierung abgegrenzt werden. Danach werden allgemein drahtlose Sensornetze in der Hausautomatisierung und ihre Eigenschaften näher beleuchtet sowie ein Vergleich zwischen leistungsschwachen und -starken Netzknoten gezogen. Darauf aufbauend wird im nächsten Schritt das drahtlose Sensornetz mit 6LoWPAN behandelt, das die Internetfähigkeit der Hausautomatisierungsanlage herstellt. Abschließend wird die Wichtigkeit der Internetfähigkeit präzisiert.

2.1.1. Aufgaben der Hausautomatisierung

Die Unterstützung der Hausbewohner bei ihren täglichen Tätigkeiten im Haus ist eine Hauptaufgabe einer Hausautomatisierungsanlage. Denn durch Erhöhung des Komforts haben es die Hausbewohner leichter, beruflichen Stress abzubauen und sich von der anstrengenden Arbeit zu erholen.

Neben der Komforterhöhung ist auch die finanzielle Einsparung ein weiteres wichtiges Ziel, das ein Hausautomatisierungssystem zu kontrollieren vermag. Durch Regeln, die

den Automatisierungsprozess steuern, werden die an diesem Prozess beteiligten Geräte bei geschickter Konfiguration zum Energiesparen gezwungen.

Diese Regeln – egal, ob sie zur Komforterhöhung oder zur finanziellen Einsparung beitragen – werden von den Hausbewohnern einmalig in ein Regelwerk eingetragen. Alles weitere wird dann automatisch durch selbiges gesteuert. Hierbei nimmt das System Informationen von Sensoren (Sensorinformationen) entgegen und setzt diese anhand der im Regelwerk definierten Regeln in Steuerbefehle für Aktoren (Aktoransteuerungen) um.

Ziel der Hausautomatisierung ist es, klein und unscheinbar den Hausbewohner zu unterstützen und damit „in den Hintergrund zurückzuweichen“[1]. Dies in Kombination mit der Regelverarbeitung unterscheidet ein Hausautomatisierungssystem von einer Fernbedienung bzw. einer Fernsteuerung.

2.1.2. Abgrenzung zur Gebäudeautomatisierung

Im Gegensatz zur Hausautomatisierung überwiegt bei der Gebäudeautomatisierung die finanzielle Einsparung. Denn die Gebäudeautomatisierung vermag nicht nur Energiekosten zu sparen, sondern auch Personaleinsparungen zu erreichen.

Durch die Automation werden teils komplexe Funktionsabläufe selbstständig durchgeführt. Diese automatischen Funktionsabläufe vereinfachen eine Überwachungsaufgabe oder eine Bedienungsaufgabe. Deshalb wird diese Art der Automatisierung insbesondere in Nicht-Wohngebäuden eingesetzt.

Die Erhöhung des Wohnkomforts gehört hierbei nicht zu den Zielen einer Gebäudeautomatisierung, da diese vor allem in öffentlichen Gebäuden oder Industriebauten Anwendung findet, nicht aber in privaten Wohnhäusern.

2.1.3. Drahtlose Sensornetze in der Hausautomatisierung

Das Hausautomatisierungssystem benötigt Sensorinformationen zur Erfüllung seiner Aufgaben. Da es üblicherweise nicht ausreichend ist, alle Sensoren in einem Gerät zu platzieren, kommt ein *verteiltes System* zum Einsatz. In diesem verteilten System werden verschiedene Netzknoten eingesetzt, wovon einige *Sensoren* und andere *Aktoren* integriert haben.

Diese Netzknoten haben die Aufgabe, Informationen (jeglicher Art) auszutauschen, und müssen folglich miteinander kommunizieren. Der Vorteil einer drahtlosen Hausautomation ist hierbei, dass die Verkabelung der Netzknoten entfällt. Es wird lediglich die selbe Funkschnittstelle zwischen zwei direkt miteinander kommunizierenden Netzknoten benötigt.

Ein primäres Ziel der Hausautomatisierung ist die finanzielle Einsparung. Demnach gehört auch die Reduzierung der Energiekosten zu diesem Ziel. Da das Hausautomatisierungssystem selbst auch Energie verbraucht, muss das System energieeffizient arbeiten, um dieses Ziel zu erreichen.

[1] An dieser Stelle wurde das Zitat von Mark Weiser übersetzt (siehe Seite 5).

Daher kommt zur Minimierung der Energiekosten des Hausautomatisierungssystems ein Sensornetz zur Anwendung: bei einer drahtlosen Kommunikation entsprechend ein drahtloses Sensornetz (engl. *wireless sensor network, WSN*). Da ein Hausautomatisierungssystem sowohl Sensoren wie auch Aktoren ansteuert, wird – um genauer zu sein – ein *drahtloses Sensor-Aktor-Netz* (engl. *wireless sensor actor network, WSAN*) eingesetzt.

Typische drahtlose Sensornetze

Typische drahtlose Sensornetze bestehen aus mehreren (hundert bis tausend) Sensorknoten und kommen in sehr unterschiedlichen Einsatzgebieten vor: beim Katastrophenschutz zur Vermeidung von Waldbränden, bei der Beobachtung von Tieren in ihrer natürlichen Umgebung, beim Militär zur Überwachung von Feindaktivitäten und bei der Küsten- und Grenzüberwachung. Allerdings sind die Voraussetzungen und Anforderungen der einzelnen Einsatzgebiete genauso unterschiedlich wie die Gebiete selbst.

Bei den letzten zwei Einsatzgebieten werden normalerweise baugleiche Sensorknoten eingesetzt, die als einzige Aufgabe haben, Aktivitäten zu detektieren und zu melden. Es werden folglich keine Aktoren benötigt. Im Gegensatz zum Sensor-Aktor-Netz bietet dieses Sensornetz folgende zwei Voraussetzungen, die die Kommunikation in diesem *homogenen Netz* vereinfachen: (1) Der Aufbau solcher homogenen Netze kann so geplant werden, dass der Ausfall weniger Sensorknoten tolerierbar ist. Da genug andere baugleiche Sensorknoten zur Verfügung stehen, können diese die ausfallenden Sensorknoten ersetzen. (2) Die Kommunikation in solch einem Sensornetz ist gerichtet, d. h., dass jeder Sensorknoten seine Informationen über die Umwelt an einen so genannten Aggregationsknoten sendet. Dieser sammelt die Informationen aller Sensorknoten und wertet sie aus. Hierbei ist es sinnvoll, dass dieser Knoten das Sensornetz auch verwaltet.

Drahtlose Sensor-Aktor-Netze

In der Hausautomatisierung können solche typischen Sensornetze nicht eingesetzt werden. Zum einen sind nicht alle Sensorknoten baugleich: Einige besitzen beispielsweise Temperatursensoren, andere Luftdrucksensoren. Zum anderen werden auch Netzknoten eingesetzt, die keine Sensoren, sondern ausschließlich Aktoren besitzen.

In der Hausautomatisierung werden folglich Sensor-Aktor-Netze eingesetzt, da die Netzknoten im Hausautomatisierungssystem gewöhnlich verschiedene Sensoren und / oder Aktoren integrieren. Das Ergebnis ist ein *heterogenes Netz*, in welchem die Knotenzahl von wenigen bis zu mehreren (hundert) Sensor-Aktor-Knoten[2] reichen kann.

[2] Netzknoten im Sensornetz werden auch allgemein Sensorknoten genannt; Netzknoten in einem Sensor-Aktor-Netz dagegen Sensor-Aktor-Knoten. Allerdings ist es nicht erforderlich, dass jeder Netzknoten sowohl Sensoren wie auch Aktoren integriert haben müssen. Zusätzlich ist der Begriff Sensor-Aktor-Netz in der Literatur nicht so geläufig und es wird meist allgemein vom Sensornetz gesprochen. Um Klarheit zu schaffen, wird der Begriff „Netzknoten“ verwendet, wenn es unwichtig ist, ob ein Netzknoten Sensoren oder Aktoren integriert. In **??** wird der Begriff Sensorknoten und Aktorknoten definiert.

2.1.4. Leistungsschwache und leistungsstarke Netzknoten

Die Untersuchung von Angriffen auf Sensornetze hat ergeben, dass einem Angreifer (Feind, engl. *adversary*) zwei Mittel zur Verfügung stehen, um ein Sensornetz anzugreifen:

1. leistungsschwache Netzknoten (engl. *mote-class nodes*) und
2. leistungsstarke Netzknoten (engl. *laptop-class nodes*).

Entsprechend haben sich die Begriffe „mote-class adversaries“ und „laptop-class adversaries“ in der englischsprachigen Literatur durchgesetzt.

Analog zu dieser Angreiferklassifikation lassen sich auch die Netzknoten eines Hausautomatisierungssystems nach ihrer Leistungsfähigkeit kategorisieren. In dieser Masterarbeit werden folglich die Begriffe leistungsschwach und „mote-class“ wie auch leistungsstark und „laptop-class“ bedeutungsgleich verwendet.

Allerdings ist es für Hausautomatisierungssysteme mit Sensor-Aktor-Netzen typisch, dass *nahezu alle* Netzknoten leistungsschwach sind. Das ist das effizienteste Mittel, um den Eigenverbrauch des Hausautomatisierungssystems sowie die Anschaffungskosten zu reduzieren.

Leistungsschwache Netzknoten (mote-class Netzknoten)

Leistungsschwache Netzknoten besitzen nur (sehr) wenig Speicher und eine geringe Rechenleistung, so dass nicht alle verfügbaren Algorithmen in angemessener Zeit ausgeführt werden können. Zudem ist i. A. bei allen leistungsschwachen, drahtlos kommunizierenden Netzknoten dieselbe Funkschnittstelle vorhanden.

Da die meisten Sensor-Aktor-Knoten (engl. *sensor actor nodes*) leistungsschwach sind und „unter Sensor-Aktor-Netzwerken […] [der] Zusammenschluss einer Vielzahl in die Umgebung eingebetteter und deshalb ggf. miniaturisierter Sensor-Aktor-Knoten [verstanden wird],“ [69] wird aufgrund der anvisierten Größe solcher Netze auch das Schlagwort „intelligenter Staub“ (engl. *smart dust*) verwendet. Da Staub aus sehr vielen Staubkörnern besteht, ist ein Sensor-Aktor-Knoten[3] demzufolge mit einem Staubkorn (engl. *mote*) gleichzusetzen. Daher hat sich auch der Begriff *mote-class* durchgesetzt, um Netzknoten als leistungsschwach zu kategorisieren.

Es gibt Hersteller, die diesen Begriff in ihre Produkte aufgenommen haben: die Tmote Sky Plattform von Moteiv und die Waspmote Plattform von Libelium.

Leistungsstarke Netzknoten (laptop-class Netzknoten)

Leistungsstarke Netzknoten besitzen im Gegensatz zu den leistungsschwachen über genügend Speicher und Rechenleistung, um die erforderlichen Algorithmen in angemessener Zeit auszuführen. Batteriebetriebene leistungsstarke Netzknoten besitzen zudem, verglichen mit batteriebetriebenen leistungsschwachen Netzknoten, höhere Energiekapazitäten.

[3] Hinweis: Das Schlagwort „intelligenter Staub“ gilt nicht nur für Sensor-Aktor-Netze, sondern auch für Sensornetze allgemein.

Oftmals haben solche leistungsstarken Netzknoten auch weitere Kommunikationsschnittstellen wie Ethernet/LAN, WLAN oder mobiles Internet integriert, um das Sensornetz mit anderen Netzen zu verbinden.

Da Sensornetze fast ausschließlich aus mote-class Netzknoten bestehen, ist ein Angreifer mit einem wesentlich leistungsstärkeren Netzteilnehmer im Vorteil. Da Angreifer in der Vergangenheit häufig einen Laptop benutzt haben, um Sensornetze mit Hilfe leistungsstarker Netzknoten anzugreifen, hat sich analog zum Begriff *mote-class* der Begriff *laptop-class* etabliert, um Netzknoten als leistungsstark zu kategorisieren. Der ausgeführte Angriff wird folglich analog zum englischen Sprachgebrauch auch „laptop-class Attacke“ genannt.

In der heutigen Zeit der mobilen Geräte haben sich neben dem Laptop auch andere im Vergleich zu Sensorknoten *leistungsfähigere* Geräte hervorgetan. Zu diesen gehören beispielsweise Netbooks, Smartphones und Tablets. Smartphones und Tablets besitzen zwar keinen auf Leistung ausgelegten Prozessor, sondern eine energiesparende Variante. Doch sind sie mit einem Mikroprozessor ausgestattet, der um $1\,\mathrm{GHz}$ Leistung besitzt und in welchem mittlerweile mehr als nur ein vollständiger Hauptprozessorkern verbaut ist. Im Vergleich zu einem Mikrocontroller mit $16\,\mathrm{MHz}$ bestehen folglich Welten.

2.1.5. Drahtlose Sensornetze mit 6LoWPAN und IEEE 802.15.4

Da im vorherigen Abschnitt bereits beschrieben wurde, warum drahtlose Sensornetze in der Hausautomatisierung einen entscheidenden Vorteil besitzen und was diese Sensornetze von anderen unterscheidet, gibt dieser Abschnitt einen Überblick zu den Kommunikationsprotokollen, die in dieser Masterarbeit wichtig sind: der IEEE 802.15.4 Standard sowie das 6LoWPAN Protokoll.

Es gibt zu drahtlosen Sensornetzen viele Protokolle. Aber nur eine Hand voll Standards haben sich durchgesetzt. Zu diesen zählt der IEEE 802.15.4 Standard, der 2003 in der ersten Version publiziert wurde. Darauf aufbauend hat sich seit 2008 aufgrund der fortwährenden Internettechnologieentwicklung das 6LoWPAN Protokoll durchgesetzt.

Energiesparende Kommunikation mit IEEE 802.15.4

Drahtlose Sensornetze benutzen eine energiesparende Funkkommunikation, um Nachrichten energieeffizient zwischen Netzknoten auszutauschen. Die Übertragungsgeschwindigkeit spielt dabei nur eine untergeordnete Rolle, weswegen auch oft Übertragungsverfahren für geringe Übertragungsraten eingesetzt werden.

Der Funkstandard, der sich in den letzten Jahren in diesem Bereich durchgesetzt hat, nennt sich *IEEE 802.15.4* und ist ein Übertragungsprotokoll für Low-power Wireless Personal Area Networks (LoWPANs), also WPANs mit geringen Übertragungsraten. Dieser bestimmt die Kommunikation auf den unteren beiden Schichten des OSI-Referenzmodells, der Bitübertragungs- und der MAC-Schicht (Teil der Sicherungsschicht). Hierbei wird der Zugriff auf das Medium (den Luftkanal) durch ein MAC-Protokoll geregelt. Organisiert wird das Netz durch einen Koordinator in jedem Personal Area Network (PAN), auch *PAN-Coordinator* genannt.

Sofern zwei Netzknoten miteinander kommunizieren wollen, die nicht in Funkreichweite sind, kommen Router zum Einsatz. Diese Router dürfen laut Standard kein Duty-Cycling verwenden und müssen das gesamte Funkprotokoll unterstützen. Im Standard werden diese Netzknoten *Full Function Devices (FFDs)* genannt. Allen anderen Netzknoten ist es gestattet, nur einen Teil des Funkprotokolls zu unterstützen. Sie werden als *Reduced Function Devices (RFDs)* bezeichnet.

Internet Protokoll für leistungsschwache Netzknoten mit 6LoWPAN

IEEE 802.15.4-basierende Netze besitzen keine Vermittlungsschicht (dritte Schicht im OSI-Referenzmodell). Zur Bereitstellung von netzwerkübergreifenden Adressen hat sich daher – analog zum IP-Standard leistungsstarker Netzknoten (IPv4 bzw. IPv6) – ein IP-Standard für leistungsschwache und drahtlos kommunizierende Netzknoten entwickelt, der IPv6-Pakete über IEEE 802.15.4-basierende Netze übermittelt. Dieses Kommunikationsprotokoll nennt sich IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) und ermöglicht es, LoWPANs mit nur geringem Aufwand in bestehende IPv6-Netze zu integrieren, indem so genannte *Border Router* eingesetzt werden. Mittels dieser Border Router können Nachrichten vom IPv6-Netz ins 6LoWPAN (und umgekehrt) vermittelt werden.

6LoWPAN sieht für jeden Netzknoten eine *Link-Local-Adresse* vor und komprimiert den IPv6- und UDP-Header, indem eine *Header Compression (HC)* angewendet wird. Zudem bietet 6LoWPAN eine Fragmentierung der IP-Pakete, damit sie über das IEEE 802.15.4-basierendes Netz versendet werden können.

6LoWPAN bietet zwei Formen des Routings an: *Mesh-under* und *Route-over* [1]. Es definiert aber kein Routing-Protokoll. Diese Funktionalität wird von einem separaten Routing-Protokoll gehandhabt.

2.1.6. Internet der Dinge

In der Einführung wird vielfach von einer drahtlosen, *internetfähigen* Hausautomatisierung gesprochen und darüber, dass sie viele Forscher fasziniert. Doch warum ist das so? Dies ist nicht nur darin begründet, drahtlose Sensornetze an das Internet anzuschließen. Vielmehr möchte man die nächste Stufe des Internets anstreben: das *Internet der Dinge*. Ein kurzer Abriss zur Geschichte des Internets hilft beim Verständnis.

In den 90er Jahren hat sich das Internet auf den Personal Computern (PCs) ausgebreitet. Beinahe zehn Jahre danach (um 2000) hat sich das Internet – mittlerweile in der vierten Generation – auch auf mobile Endgeräte erstreckt. Das mobile Internet ist geboren.

Diese beiden Internet-Generationen könnte man auch als *Internet der Computer* [2] bezeichnen. Denn der Computer – egal, ob in Form eines PCs daheim, in Form eines Laptops, Netbooks oder als Mobiltelefon, Smartphone, Tablet etc.– ist für den Menschen ein nützliches Werkzeug, um untereinander zu kommunizieren, Informationen jeglicher Art auszutauschen oder Dienste von Unternehmen in Anspruch zu nehmen und das

weltweit. Die Wichtigkeit des Computers sowie des Internets im Alltag sind in modernen Gesellschaften heutzutage nicht mehr wegzudenken [3].

Kombiniert man die Tatsache, dass Computer und Internet nicht mehr aus dem Alltag verbannt werden können, mit der Tatsache, dass der Computer immer kleiner wird, kommt man zu dem Schluss, dass beliebige Geräte sich vernetzen, um eine neuartige Kommunikationsschnittstelle für den Menschen zu schaffen. So existiert die Idee, dass die Waschmaschine sich mit einem Kommunikationsdienst verbindet, um dem Hausbewohner eine Nachricht zu schicken, wenn die Waschmaschine fertig ist oder sonstige Probleme auftreten. Analog kann der Kühlschrank oder die Tiefkühltruhe im Urlaub eine Nachricht versenden, um anzuzeigen, dass die Temperatur nicht mehr im Normalbereich liegt.

Verallgemeinert ausgedrückt, vernetzen sich Geräte untereinander, um mit dem Internet neuer Kommunikations- und Informationsmöglichkeiten zu erschaffen. Diese Geräte, die allgemein auch „Dinge“ genannt werden, bilden damit die dritte Generation des Internets: das *Internet der Dinge* (engl. *Internet of Things, IoT*).

Die Vision des Internets der Dinge ist bereits 1991 das erste Mal von Mark Weiser in einem Aufsatz [4] erwähnt worden. Das Internet der Dinge wird von Mattern und Floerkemeier in ihrem Artikel „Vom Internet der Computer zum Internet der Dinge“ geeignet beschrieben:

> Das Internet der Dinge steht für eine Vision, in der das Internet in die reale Welt hinein verlängert wird und viele Alltagsgegenstände ein Teil des Internets werden. Dinge können dadurch mit Information versehen werden oder als physische Zugangspunkte zu Internet-Services dienen, womit sich weitreichende und bis dato ungeahnte Möglichkeiten auftun.
>
> Eine zentrale Rolle kommt in dieser Vision den „smarten“ (bzw. „intelligenten“) Objekten zu: Ausgestattet mit Informations- und Kommunikationstechnik und angebunden an den Cyberspace mit seinen mächtigen Diensten erhalten alltägliche Gegenstände eine neue Qualität: Diese können über Sensoren ihren Kontext wahrnehmen, sich miteinander vernetzen, auf Internetservices zugreifen und mit dem Menschen interagieren. [2, zwei Auszüge]

Übertragen auf die Hausautomatisierung bedeutet dies, dass Alltagsgegenstände im Haus Teil des Internets werden können. Ein Konsortium von Firmen wurde gegründet, um die Zusammenarbeit von internetfähigen Geräten im Bereich Heimautomation, Unterhaltung und Produktivität zu verbessern [53]. Hierbei hilft das 6LoWPAN Protokoll, indem es auf einfache Weise ermöglicht, Informationen mit klassischen IPv6 Netzen auszutauschen, wodurch das drahtlose Sensornetz – wenn gewollt – auch Dienste im World Wide Web nutzen kann.

Anders herum können Netzknoten im drahtlosen Sensornetz ebenfalls Dienste für Geräte außerhalb des Sensornetzes anbieten. Ein Beispiel ist das Abrufen aktueller Sensorinformationen, indem ein Webservice angeboten wird.

2.2. Anforderungen

In der Einführung wurde das Beispiel gebracht, dass Bewegungsdaten von Bewegungsmeldern missbraucht werden können. Daher wird zuerst eine rechtliche Anforderung diskutiert, die den Schutz des allgemeinen Persönlichkeitsrechts behandelt. Im Anschluss wird mit einer betriebswirtschaftlichen Anforderung die Notwendigkeit der Offenheit eines Hausautomatisierungssystems beleuchtet, da Offenheit im Internet und damit auch in der internetfähigen Hausautomatisierung eine wichtige Rolle spielt. Abschließend folgen zwei betriebswirtschaftlich-technische Anforderungen, die es zu lösen gilt, um das Interesse der Hausbewohner zu wecken: die Skalierbarkeit des Systems sowie dessen Energieverbrauch.

Aber erst durch Erfüllen aller Anforderungen wird die Akzeptanz eines Hausautomatisierungssystems sichergestellt.

2.2.1. Schutz des allgemeinen Persönlichkeitsrechts

Aus rechtlicher Sicht ist der Schutz des allgemeinen Persönlichkeitsrechts die wichtigste Anforderung an ein Hausautomatisierungssystem. Dieses allgemeine Persönlichkeitsrecht[4] ist als Menschenrecht in allen modernen Demokratien verankert. Aus dem Schutz des Persönlichkeitsrechts wird der Schutz der Privatsphäre, der Intimsphäre und der Geheimsphäre abgeleitet:

Schutz der Privatsphäre Nach deutschem Recht bedeutet der Schutz der Privatsphäre, dass der Mensch die Möglichkeit hat, einen persönlichen Bereich zu besitzen, in welchem er sich *frei* bewegen und *ungezwungen* verhalten kann. In diesem Bereich darf der Mensch nicht die Befürchtung haben, dass er durch Dritte beobachtet oder abgehört werden könnte. Konkret wird der Schutz der Privatsphäre durch die *Unverletzlichkeit der Wohnung* und das *Post- und Fernmeldegeheimnis* sichergestellt.

Schutz der Intimsphäre Neben dem Schutz des häuslichen Bereichs ist auch die Intimsphäre der einzelnen Bürger geschützt. Zur Intimsphäre gehören beispielsweise die inneren Gedanken oder die Tagebuchaufzeichnungen (auch wenn sie elektronisch sind).

Schutz der Geheimsphäre Der Schutz der Geheimsphäre ist ebenfalls durch die allgemeinen Persönlichkeitsrechte abgedeckt. Aufnahme oder Veröffentlichung (Weitergabe) von Äußerungen eines/r Betroffenen, die ohne oder sogar gegen den Willen dieser Person getätigt wurden, sind im Sinne der Geheimsphäre zu unterlassen. Hierzu zählen persönliche Daten, die unter dem Schutz des Bundesdatenschutzgesetzes stehen.

[4] Grundlage für diesen Abschnitt bildet der Lexikonbeitrag im JuraForum.de [54]. Es sei darauf hingewiesen, dass es auch Ausnahmen beim Schutz des allgemeinen Persönlichkeitsrechts gibt, die in diesem Abschnitt nicht genannt werden, da an dieser Stelle ein Überblick reichen soll.

Die Vergangenheit hat gezeigt, dass Menschen entweder leichtfertig oder aus Unwissenheit mit ihren personenbezogenen Daten umgehen[5], obwohl Reportagen und Datenschützer Aufklärung betreiben.

Der Gedanke, den eigenen Computer oder das eigene Handy ausspioniert oder überwacht zu wissen, ist für viele Menschen weniger greifbar, wie der Gedanke, dass die eigene Wohnung überwacht wird. Daher ist gerade diese Anforderung eine wichtige in Bezug auf die Akzeptanz einer Hausautomatisierungsanlage.

Es ist daher um so wichtiger, dass Entwickler einer Hausautomatisierungsanlage oder einer ihrer Komponenten auf diese Anforderung Rücksicht nehmen. Ein kleines Beispiel soll schildern, was mit der Rücksichtnahme gemeint ist:

Ein Entwickler hat ein aktuelles Verschlüsselungsverfahren bereitstellt, um eine sichere Kommunikation und damit den Schutz der Daten zu gewährleisten. Allerdings liefert er seine Geräte mit dem gleichen Schlüssel aus. Wird dieser nicht manuell durch den Endverbraucher durch einen neuen Schlüssel ersetzt, so wird mit diesem Standard-Schlüssel eine verschlüsselte Kommunikation aufgebaut. Da davon auszugehen ist, dass ein Angreifer diesen bereits kennt, ist die Sicherheit der Daten nicht mehr gewährleistet. Der Entwickler hätte in diesem Fall die Gewährleistung der Sicherheit der Daten auf den Endverbraucher abgewälzt.

Es ist somit vom Entwickler gefordert, dass er ein Verfahren verwendet, das den Endverbraucher so wenig wie möglich in den Prozess, der die Sicherheit der Daten gewährleistet, integriert. Nur so kann der Schutz der Daten für alle Endverbraucher garantiert werden.

2.2.2. Interoperabilität in der Hausautomatisierung

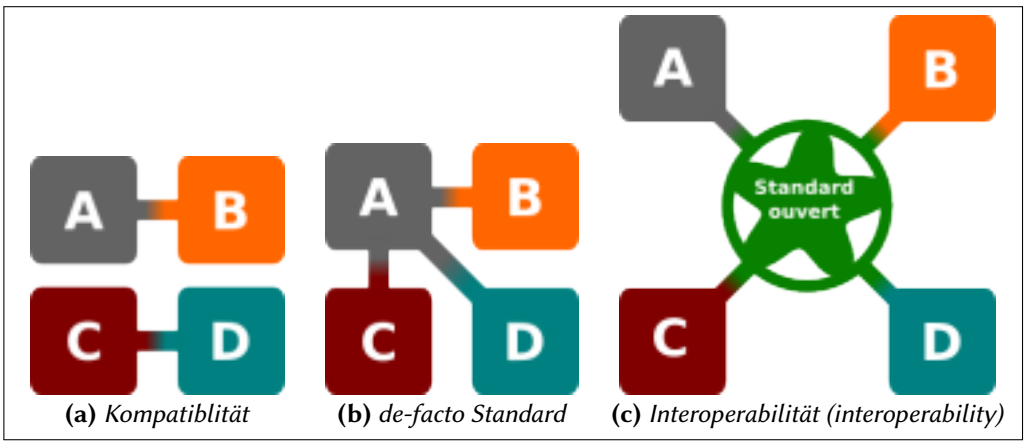
Die erfolgreiche Internetfähigkeit der Hausautomatisierung erfordert offene und zukunftsfähige Standards und damit die Gewährleistung der Interoperabilität, wobei die Interoperabiltät die höchste Stufe der Operabilität[6] darstellt.

Die Operabilität beschreibt die Fähigkeit eines Systems mit anderen Systemen zusammenzuarbeiten. Die AFUL Interoperability Working Group hat hierzu drei Stufen aufgestellt, die in Abbildung 2.1 bildhaft dargestellt sind.

Kompatiblität Die Kompatiblität weist auf ein Miteinanderfunktionieren von Systemen hin. Das bedeutet aber nicht, dass diese Systeme offen oder kompatibel mit zukünftigen Systemen sein müssen.

[5] Zwei Beispiele demonstrieren diese Situation: Facebook behält sich das Recht, Namen und/oder Profilbild zusammen mit den Inhalten oder Informationen des Facebook-Nutzern an Dritte weiterzugeben und zu veröffentlichen; auf Betriebssystemen für mobile Endgeräte werden kostenlose Spiele gegen die Einsicht in Kurznachrichten zur Verfügung gestellt.

[6] Der englische Begriff „operability“ wird in der deutschen Sprache mit „Interoperabilität“ übersetzt. Doch wird auch der englische Begriff „interoperability“ mit diesem Wort übersetzt. Das führt im folgenden Abschnitt zur Verwirrung, weshalb der Begriff *Operabilität* als Übersetzung zu „operability“ verwendet wird.



(a) *Kompatiblität* **(b)** *de-facto Standard* **(c)** *Interoperabilität (interoperability)*

Abbildung 2.1. *Grad der Interoperabilität (operability) [5].*
Die Buchstaben A bis D repräsentieren Systeme, die meist von Unternehmen angeboten werden.

De-facto Standard Der de-facto Standard ist ein meist von der Industrie geschaffener technischer Standard, welcher von keinem Normengremium verabschiedet wurde. Er wurde von einem oder mehreren Unternehmen definiert und von (genug) anderen Unternehmen akzeptiert.

Interoperabilität Die Interoperabilität wird von der AFUL Interoperability Working Group in angemessener Weise definiert:

> Interoperabilität ist die Fähigkeit eines Programms oder Systems (dessen Schnittstellen vollständig offengelegt sind) mit anderen gegenwärtigen oder zukünftigen Produkten oder Systemen ohne Einschränkungen hinsichtlich Zugriff oder Implementierung zusammenzuarbeiten bzw. zu interagieren. [5]

Da sich die Anforderung nach Interoperabilität in den weit verbreiteten Internettechnologien (IP, TCP, UDP, HTTP, u. v. m.) bewährt hat, ist diese Anforderung auch in der drahtlosen, internetfähigen Hausautomatisierung sehr wichtig. Dies bedeutet allerdings nicht, dass nur weit verbreitete Standards verwendet und keine energiesparsamen Varianten gefunden werden dürfen.

Diese Anforderung gilt nicht nur für die Kommunikation zwischen zwei Netzknoten, sondern auch für die geplanten Technologien bei der Konzipierung der Sicherheitsarchitektur. Dennoch ist es sinnvoll, zusätzlich zu den offenen Technologien auch Technologien zu betrachten, die (noch) nicht vollständig offengelegt sind oder auf denen Patente bestehen, sofern die hinter diesen Technologien basierenden Konzepte zur Verbesserung des Systems beitragen.

2.2.3. Offene Hausautomatisierung

Einführend wurde bereits erwähnt, dass die offene Hausautomatisierung nicht nur vollständig offengelegte Standards anstrebt, sondern auch freie Software und freie Hardware, um die Attraktivität aufgrund niedriger Anschaffungskosten zu steigern.

Zu diesem Thema wurden folgende Erkenntnisse zum Beginn des Forschungsprojekts der University of Applied Sciences Dresden (März 2012) erlangt, an dem der Autor mitgewirkt hat:

> Im Bereich der Hausautomatisierung gibt es bereits eine Vielzahl an sowohl offenen als auch proprietären Lösungsansätzen. Jedoch erfüllen sie die Bedingungen an eine freie und erweiterbare Hausautomatisierungslösung nur bedingt.
>
> Offene Varianten sind häufig „Ein-Mann"-Projekte, die sehr spezielle Probleme verfolgen oder nur sehr unzureichend dokumentiert sind. Teilweise stützen sie sich aber auch nur darauf, vorhandene proprietäre Systeme zu verwalten und miteinander zu verbinden.
>
> Diese proprietären Systeme sind für Fremdanbieter häufig aufgrund hoher Lizenzkosten unattraktiv. Des Weiteren setzen sie Netzwerkprotokolle ein, die energieeffizienter als standardisierte Protokolle wie HTTP, TCP und IPv6 arbeiten. Die Protokolle stehen aber durch ihre Geschlossenheit der Interoperabilität entgegen. [6]

In den letzten zwei Jahren hat sich allerdings viel geändert: Es entstehen immer mehr Projekte, die eine offene Hausautomatisierung anstreben. Unter diesen Projekten befindet sich beispielsweise das HexaBus System [55], [56].

Nach Meinung des Autors ist die offene Hausautomatisierung ein wesentlicher Schritt, um Interoperabilität zu fördern und Monopolstellungen zu vermeiden. Durch die Offenheit von Software und Hardware ist es wahrscheinlich, dass sich eine Vielzahl an Interessierten in einer Community gruppieren und die Weiterentwicklung unterstützen. Hierzu gibt es viele Beispiele, gerade im Linux Bereich. Zudem wäre ein Hardwarehersteller nicht mehr dazu verpflichtet, alle möglichen Teile eines Systems anzubieten (Server und verschiedene Automatisierungsfunktionalitäten), sondern er kann sich auf die Integration von Hausautomatisierungsfunktionalitäten für seine Produkte konzentrieren, ohne dass dieser mit hohen Lizenzkosten konfrontiert wird.

2.2.4. Skalierbarkeit

Drahtlose Sensornetze können eine statische oder auch dynamische Netzstruktur aufweisen. Klassische Netzwerke wie Wireless Local Area Networks (WLANs) sind üblicherweise statisch, da sie bei Erstellung eine zugrunde liegende Struktur aufweisen. In solchen Netzen existiert oft ein Router, der das Netzwerk mit dem Internet verbindet und so Pakete weiterleitet. Dieser Router ist allen Geräten bekannt.

Statische drahtlose Sensornetze bauen eine solche Struktur bei Ausbringung der Netzknoten selbstständig auf. Hierbei suchen die Netzknoten ihre Nachbarknoten und verbinden sich mit ihnen. Doch Änderungen dieser Struktur nach Ausbringung sind schwierig umsetzbar und bedürfen im Zweifelsfall eine Neuausbringung.

Dynamische drahtlose Sensornetze können sich dagegen neuen Umständen entsprechend anpassen und sich neu organisieren. Zur Ermittlung des Routingpfades von einem Netzknoten zum anderen sind folgende zwei Methoden von Bedeutung: (1) Der Routingpfad wird beim Versenden der Nachricht ermittelt, wodurch der Pfad aktuell ist, aber eine Verzögerung beim Versenden entsteht (On-Demand Routing). (2) Im Gegensatz dazu können Routingpfade in regelmäßigen Abständen überprüft werden, indem jeder Netzknoten seine aktuell zur Verfügung stehenden Nachbarn ermittelt und diese Informationen wiederum mit anderen teilt (Proactive Routing).

In einem drahtgebundenen Hausautomatisierungssystem ist eine statische Netzstruktur sinnvoll. Doch in einem drahtlos kommunizierenden System bietet gerade die dynamische Netzstruktur Vorteile. Das Hinzufügen neuer Netzknoten gestaltet sich für den Hausbewohner sehr einfach. Im Idealfall muss er diesen nur mit Strom versorgen und eine Regel im Hausautomatisierungssystem parat haben. Bei Netzknoten, die Informationen lediglich weiterleiten, entfällt sogar das Hinzufügen einer neuen Regel.

Folglich kann eine rasche Installation ohne große IT-Kenntnisse auch von einem Hausbewohner vorgenommen werden und benötigt keinen Fachmann. Gerade dieser Fakt wird aus betriebswirtschaftlicher Sicht einem Endverbraucher bei der Frage entscheidend sein, ob er sich ein Hausautomatisierungssystem zulegt oder nicht. Somit hängt die Akzeptanz von einem Hausautomatisierungssystem davon ab, wie flexibel neue Netzknoten eingebunden werden können.

Auf der anderen Seite darf der Ausfall (das Entfernen) von Netzknoten nicht dazu führen, dass das Hausautomatisierungssystem zum Erliegen kommt. Das wird um so wichtiger, je stärker das Hausautomatisierungssystem in den Hintergrund unseres Lebens tritt. In solchen Fällen sollte ein System den Hausbewohner entsprechend informieren.

2.2.5. Energieeffizienz und Lebensdauer

Wie bereits in Unterabschnitt 2.1.1 beschrieben wurde, ist finanzielle Einsparung eine der Hauptaufgaben der Hausautomatisierung und damit ein wesentliches Merkmal für die Akzeptanz des Hausautomatisierungssystems. Denn ohne Energieeffizienz ist es fraglich, ob das Hausautomatisierungssystem zur finanziellen Einsparung beiträgt, obwohl es theoretisch dazu in der Lage wäre.

Weiterhin sind viele leistungsschwache Netzknoten batteriebetrieben, um das Kabellegen zu einer Steckdose zu vermeiden. Daher ist es auch wichtig, dass diese Netzknoten eine hohe Lebensdauer besitzen, folglich mit einem Satz Batterien über mehrere Monate oder Jahre hinweg auskommen. Diese hohe Lebensdauer kann nur erreicht werden, wenn der Energieverbrauch extrem gering gehalten wird.

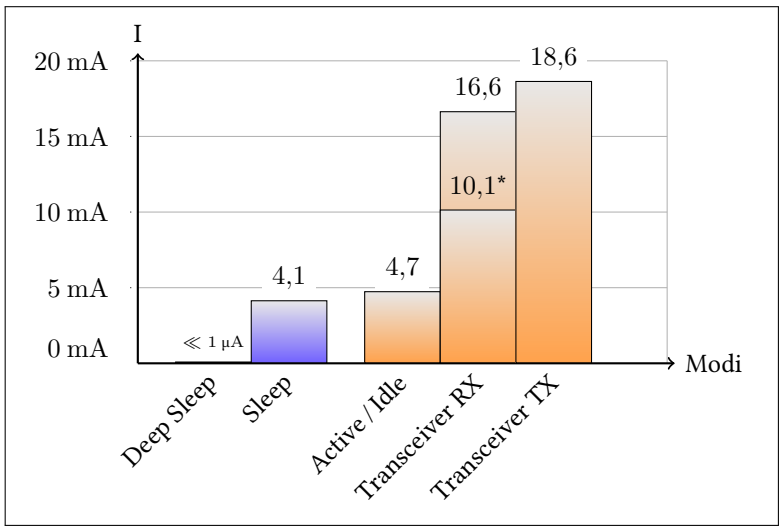


Abbildung 2.2. Beispielhafter Stromverbrauch in unterschiedlichen Betriebsmodi. *Abgebildet ist der Stromverbrauch eines leistungsschwachen Netzknotens, der einen AVR Microcontroller der Firma Atmel benutzt: ATmega128rfa1 bzw. sein Nachfolger ATmega128rfa2. * Der ATmega128rfa2 kann den Stromverbrauch im empfangsbereiten Betriebsmodus (Transceiver RX) um* $6{,}5\ \mathrm{mA}$ *senken, wenn der Reduce Power Consumption (RPC) Mode aktiviert ist.*

Bei einer Netzgröße von mehreren hundert Netzknoten führt dieser Vorteil zu einer erheblichen Energieeinsparung, wodurch drahtlose Sensornetze erst für die Hausautomatisierung attraktiv geworden sind.

Dieser Vorteil geht aber auch mit Nachteilen einher: Leistungsschwache Netzknoten können folglich nicht die gleiche Leistung erbringen wie leistungsstarke Netzknoten. Algorithmen müssen speziell für leistungsschwache Netzknoten konzipiert werden, damit sie effizient und zugleich energiesparsam arbeiten. Aber es müssen auch weitere Mechanismen zur Energieeinsparung geschaffen werden, um die Lebensdauer der Netzknoten sicherzustellen und weiter zu steigern.

Deep Sleep Modus Bei der Auswahl der Hardware für leistungsschwache Netzknoten können Mikrocontroller genommen werden, die einen so genannten Deep Sleep Modus beherrschen. Der Sinn dieses Modus ist es, nicht nur in einen normalen Schlafzustand zu wechseln, um Strom zu sparen, sondern dabei auch unbenötigte Komponenten abzuschalten.

Abbildung 2.2 zeigt einen beispielhaften Energieverbrauch in unterschiedlichen Betriebsmodi eines leistungsschwachen Netzknotens. Es kann erkannt werden, dass nur eine leichte Stromeinsparung zwischen dem Sleep Mode und Active Mode erreichbar ist. Doch die Stromeinsparung zwischen dem Sleep Mode sowie dem Deep Sleep Mode ist gravierend.

Duty-Cycling Da das Betriebssystem nicht ständig aktiv sein sollte, aber dennoch so oft wie möglich empfangsbereit sein muss, wurde das Duty-Cycling eingeführt.

Denn in Abbildung 2.2 ist auch erkennbar, dass es klug ist, den Transceiver so oft wie möglich auszuschalten und nur zum Senden in den TX Mode übergehen zu lassen. Beim Duty-Cycling folglich wechselt das Betriebssystem regelmäßig zwischen „aktiven“ und „schlafenden“ Betriebsmodi: Nach einer bestimmten Zeit oder mit Hilfe eines bestimmten Aufwachsignals nimmt der Netzknoten seine gewohnte Tätigkeit wieder auf und legt sich anschließend erneut schlafen. Der Duty-Cycle (Schlaf-Wach-Zyklus) gibt an, wie lange die Wachphase im Verhältnis zur Schlafphase ist:

$$\text{Duty-Cycle} = \frac{\text{Dauer der Wachphase}}{\text{Dauer der Schlafphase}} = \frac{T_w}{T_s}$$

Wenn ein Netzknoten durchschnittlich $5\ \text{min}$ wach ist, aber $100\ \text{min}$ schläft, hat er den gleichen Duty-Cycle wie ein zweiter Netzknoten, der durchschnittlich $100\ \text{ms}$ wach ist und $2\ \text{s}$ schläft:

$$\text{Duty-Cycle} = \frac{5\ \text{min}}{100\ \text{min}} = \frac{100\ \text{ms}}{2000\ \text{ms}} = 0{,}05 = 5\ \%$$

Energiesparende Netzknoten haben einen Duty-Cylce, der weit weniger als $5\ \%$ beträgt. Demzufolge sind sie nur selten wach.

Kommunikationsaufkommen Damit sich ein Netzknoten nach Versenden oder Empfangen einer Nachricht schnell wieder schlafen legen kann und so der durchschnittliche Duty-Cycle klein gehalten wird, ist es erforderlich, das Kommunikationsverhalten zu optimieren. Lange Nachrichten mit überflüssigen Inhalten sollten also vermieden werden. Dies gilt auch für den Nachrichtenoverhead, weshalb in 6LoWPAN eine Header Compression eingeführt wurde, mit dessen Hilfe der Nachrichtenoverhead verkleinert wird.

Anwendungen müssen diese Stromsparmechanismen berücksichtigen, wenn sie ihre Tätigkeiten ausführen. Aber auch das Kommunikationsverhalten im Sensornetz muss sich diesen Umständen hin anpassen, d. h., dass Netzknoten, die in der Schlafphase sind, in dieser Zeit keine Nachrichten über die Funkschnittstelle empfangen können.

2.3. Ansatz zur Konzipierung einer Sicherheitsarchitektur

Um Strom zu sparen und energieeffizient zu sein, ist es nicht zwingend notwendig, dass alle Netzknoten leistungsschwach sind. Leistungsstarke Netzknoten müssen lediglich sparsam und sinnvoll eingesetzt werden.

Hierbei ist es wichtig, die Struktur des Systems zu kennen, für das eine Sicherheitsarchitektur erstellt werden soll. Insbesondere ist dabei zu bedenken, dass ein Angreifer nicht den Anforderungen nach Energieeffizienz unterliegt.

Da allerdings die eigentliche Hausautomatisierungssteuerung nicht bekannt bzw. aufgrund der Anforderung nach Interoperabilität allgemein gehalten ist, gilt es einen Zwischenweg zu finden. Aus diesem Grund sieht der konkrete Ansatz zur Konzipierung einer Sicherheitsarchitektur folgende Schritte vor:

I. Im nächsten Kapitel wird eine Analyse von Hausautomatisierungssystemen durchgeführt. Aufgrund der Anforderung nach Interoperabilität wird keine genaue Anwendung betrachtet, sondern die Kommunikation im Hausautomatisierungssystem. Diese Analyse ist notwendig, damit die Sicherheitsarchitektur entworfen werden kann. (Kapitel 3)

II. Da die Hausautomatisierung ein Anwendungsgebiet des Internets der Dinge ist, gilt es, eine sichere Kommunikation im Internet der Dinge zu diskutieren. Nach einer Auffrischung ausgewählter Grundlagen, die für die Konzepterstellung wichtig sind, werden leichtgewichtige Sicherheitslösungen zur Kommunikation im Internet der Dinge vorgestellt. (Kapitel 4)

III. Nach der Analyse der Hausautomatisierungssysteme und der Auswahl eines davon sowie der allgemeinen Betrachtung der Kommunikation in Anwendungen zum Internet der Dinge, wird das Konzept der Sicherheitsarchitektur vorgestellt. Dieses gliedert sich in folgende vier Schritte: Erstellung einer Gefahrenanalyse, Definition von Sicherheitsanforderungen, Ausarbeitung von Sicherheitsmaßnahmen Ausarbeitung eines möglichen Schlüsselmanagements. (Kapitel 5)

IV. Zuletzt soll das erstellte Konzept der Sicherheitsarchitektur evaluiert werden. Da die Umsetzung des gesamten Konzepts den Umfang dieser Arbeit sprengt, wird ein Teil des Konzepts begutachtet. (Kapitel 6)

3 Analyse von Hausautomatisierungssystemen

> *Security is as important in smart object networks as it is in traditional computer networks, if not more so. By leveraging well-established security mechanisms and networking standards, and adapting them appropriately for resource-constrained environments, we can enhance the security of smart objects, their data and the networks in which they participate.*
>
> —IPSO Alliance (2013)

Die Konzipierung einer gut durchdachten Sicherheitsarchitektur verlangt Kenntnisse zum Hausautomatisierungssystem (bzw. zur Hausautomatisierungsanwendung), um Gefahren und Sicherheitsanforderungen angemessen bestimmen zu können. Da allerdings bisher kein spezielles Hausautomatisierungssystem vorausgesetzt wurde, ist es nötig, verschiedene Hausautomatisierungssysteme zu analysieren, zu vergleichen und dann eines davon auszuwählen. Zur Auswahl eines Hausautomatisierungssystems werden daher in diesem Kapitel folgende Schritte durchgeführt:

(1) Im ersten Schritt wird das Hausautomatisierungssystem in logische Ebenen eingeteilt. (2) Danach werden die Komponenten eines Hausautomatisierungssystems vorgestellt. Dabei werden sie den logischen Ebenen zugeordnet. (3) Eine Übersicht zu verschiedenen Systemen leitet die Analyse ein. Es wird gezeigt, nach welchem Kriterium die verschiedenen Hausautomatisierungssysteme unterschieden werden können. (4) Dann wird die System-Architektur jedes Systems kritisch analysiert. (5) Im nächsten Schritt werden die verschiedenen System-Architekturen in Hinblick auf die Voraussetzungen und Anforderungen, die in Kapitel 2 aufgestellt wurden, miteinander verglichen. (6) Eine Auswahl verwandter Hausautomatisierungssysteme zeigt aktuelle Forschungen und ergänzt den theoretischen Vergleich um praktische Beispiele. (7) Abschließend wird eine dieser analysierten System-Architekturen ausgewählt, das dann zur Konzipierung der Sicherheitsarchitektur verwendet wird.

Nach der Analyse von Hausautomatisierungssystemen, welche aufgrund des Vorliegens eines expandierenden Marktsegments erfolgt ist, schließt sich die Diskussion zur Sicherheitsarchitektur an. Diese Diskussion abschließend werden Möglichkeiten des so genannten Schlüsselmanagements gesondert besprochen, da dieses Thema in den letzten Jahren oft umgangen wurde, insbesondere im Gebiet der drahtlosen Hausautomatisierung.

3.1. Logische Ebenen

Eine Kommunikationsschnittstelle im Hausautomatisierungssystem sorgt dafür, dass der Hausbewohner dem Hausautomatisierungssystem mitteilen kann, welche Automatisierungsregeln der Hausbewohner sich wünscht. Folglich kann das Hausautomatisierungssystem flexibel auf eigene Wünsche und Bedürfnisse hin angepasst und somit konfiguriert werden. Nach dem Motto „Vertrauen ist gut, Kontrolle ist besser“ ist es für den Hausbewohner ebenfalls wichtig, zusätzlich zur Konfigurationsmöglichkeit auch Übersichten zu den Sensorinformationen einsehen zu können, folglich Automatisierungsvorgänge überwachen zu können. Ist diese Verwaltungsmöglichkeit (dieses **Management**) auch aus dem World Wide Web verfügbar, ist der Hausbewohner in der Lage, mehrere Wohnsitze zu überwachen und zu konfigurieren.

Sofern Automatisierungsvorgänge konfiguriert wurden, sorgt die **Automation** dafür, dass den Bewohnern der Umgang im Haus erleichtert wird, indem gewisse Tätigkeiten automatisiert ausgeführt werden, d. h. ohne Zutun des Hausbewohners. So wird ein erhöhter Wohnkomfort und ein höheres Sicherheitsempfinden beim Hausbewohner erreicht.

Demnach gliedert sich ein Hausautomatisierungssystem in zwei logische Ebenen: in die *Automationsebene* und in die *Managementebene*. Erst durch die Integration beider Ebenen ist ein Hausautomatisierungssystem vollständig.

3.2. Komponenten

Da jedes Hausautomatisierungssystem, unabhängig seiner System-Architektur, beide logische Ebenen vereint, werden die Komponenten diesen beiden Ebenen zugeorndet. Je nach Gestaltung der jeweiligen System-Architektur können jedoch mehrere Komponenten in einem einzigen Netzknoten eingebettet sein.

3.2.1. Komponenten der Managementebene

Die Managementebene ist die Kommunikationsschnittstelle zwischen dem Hausbewohner und dem Hausautomatisierungssystem. Da diese Schnittstelle in heutiger Zeit i. A. grafisch gestaltet ist, wird ein **Interface** benötigt, das diese grafische Schnittstelle umsetzt. Dieses Interface bietet also dem Hausbewohner die Möglichkeit, das System zu konfigurieren und zu überwachen. In welchem Netzknoten sich das Interface befindet, hängt jedoch von der Gestaltung der jeweiligen System-Architektur ab.

Die Automationsregeln, die der Hausbewohner am Interface konfiguriert, müssen an diejenigen Netzknoten weitergereicht werden, die am Automationsprozess beteiligt sind. Dieser Verteilung der Automationsregeln übernimmt ein **Hausautomationskoordinator**. Ein Beispiel soll den Einfluss der Automationsregeln auf den Kommunikationsfluss im Hausautomatisierungssystem verdeutlichen. Folgende Automationsregel soll konfiguriert werden:

Wenn die Raumtemperatur unter 20 °C *fällt, soll die Heizung angehen. Sobald die Raumtemperatur* 22°C *erreicht oder übersteigt, wird die Heizung abgestellt.*

Zur Umsetzung dieser Regel wird eine Raumtemperatur benötigt. Dieser Raumtemperatur wird durch Aggregation verschiedener Umgebungstemperaturen, die von Temperatursensoren gemessen werden, gebildet. Wenn im Raum beispielsweise vier verschiedene Temperatursensoren verteilt sind, dann wird die Raumtemperatur durch Mittelwertbildung dieser vier Sensorinformationen berechnet. Wie genau die Aggregationsfunktion aussieht, ist durch eine weitere Regel definiert. Weiterhin muss die Heizung an- oder abgestellt werden können. Die Ansteuerung eines Heizungsreglers (eines Aktors) ist somit erforderlich. Kurz und bündig müssen zur vollständigen Umsetzung der Regel folgende Schritte erledigt werden:

1. Messung aller Umgebungstemperaturen (im Raum)
2. Versand dieser Umgebungstemperaturen an einen Aggregationsknoten
3. Bildung der Raumtemperatur aus allen Umgebungstemperaturen
4. Versand der Raumtemperatur an den Regelwerkknoten (sofern es sich nicht um denselben Netzknoten handelt)
5. Ausführung der Automationsregel
6. Neueinstellung des Heizungsreglers

Der Hausautomationskoordinator legt also fest, wer mit wem kommunizieren muss, damit die Automationsregeln erfolgreich umgesetzt werden. Der Aufwand, der für diesen Schritt nötig ist, hängt von der System-Architektur ab.

3.2.2. Komponenten der Automationsebene

Die Automationsebene des Hausautomatisierungssystems ist dagegen komplexer und besteht aus verschiedenen Komponenten, die zusammen ein verteiltes System bilden und drahtlos miteinander vernetzt sind. Kennzeichnend für diese Ebene ist der Einsatz vieler leistungsschwacher Netzknoten, um Energie zu sparen. Einige dieser leistungsschwachen Netzknoten sind wiederum batteriebetrieben.

In der Automationsebene eines Hausautomatisierungssystem findet man folgende nach Funktionsweise getrennte Komponenten:

Sensorknoten sind Netzknoten, die Sensorinformationen an das Hausautomatisierungssystem weitergeben. Diese beziehen sich entweder auf den Netzknoten selbst (Batteriestatus, Temperatur im Gerät) oder auf die Umgebung (Umgebungstemperatur, Luftfeuchtigkeit).

Aktorknoten sind Netzknoten, die dem Hausautomatisierungssystem eine Schnittstelle zur Ansteuerung der integrierten Aktoren bereitstellen. Da mit Hilfe von Aktoren Aktionen ausgeführt werden können, sind sie essentielle Bestandteile des Systems.

Regelwerkknoten sind Netzknoten, die eine Regelverarbeitung (Regelwerk) besitzen. In einem Hausautomatisierungssystem ist mindestens ein Regelwerkknoten vorhanden. Das Regelwerk wird genutzt, um die Automatisierung des Hauses zu realisieren,

und gibt an, welche Bedingungen an bestimmte Aktoransteuerungen geknüpft sind. Hierzu werden insbesondere die Sensorinformationen genutzt, aber auch zeitliche Bedingungen sind gebräuchlich. Aggregationsknoten gehören ebenfalls zu den Regelwerkknoten, da sie ebenfalls Regeln zur Vereinfachung von Sensorinformationen ausführen.

Routingknoten sind Netzknoten, die lediglich die Nachrichten der anderen Netzknoten weiterleiten. Da sie weder Sensoren, Aktoren, noch eine Regelverarbeitung integriert haben, besitzen sie keine Hausautomatisierungsfunktionalität. Dennoch sind sie erforderlich, um zwei Netzknoten, die nicht direkt miteinander drahtlos kommunizieren können, zu verbinden.

Mischformen der zuvor genannten Netzknoten kommen ebenfalls häufig zum Einsatz. Die wohl beliebteste Mischform sind Sensor-Aktor-Knoten, die sowohl Sensoren wie auch Aktoren integriert haben. Aber auch nicht-batteriebetriebene Sensorknoten mit Routingfunktionalitäten sind nicht selten.

3.3. Analyse verschiedener System-Architekturen

Nachdem die Komponenten eines Hausautomatisierungssystems vorgestellt wurden, werden nun in diesem Abschnitt verschiedene System-Architekturen für Hausautomatisierungssysteme auf ihre Netzwerkkommunikation hin analysiert. Die Analyse der System-Architekturen betrachtet hierbei nur die Kommunikation auf Anwendungsschicht. Transportschicht und darunterliegende Schichten werden zugunsten der Übersichtlichkeit vernachlässigt.

Der Regelwerkknoten beinhaltet die Regelverarbeitung und steuert somit den Automationsprozess. Da mindestens ein Regelwerkknoten in einem Hausautomatisierungssystem vorkommt, können System-Architekturen dadurch unterschieden werden, wie die Regelverarbeitung umgesetzt ist.

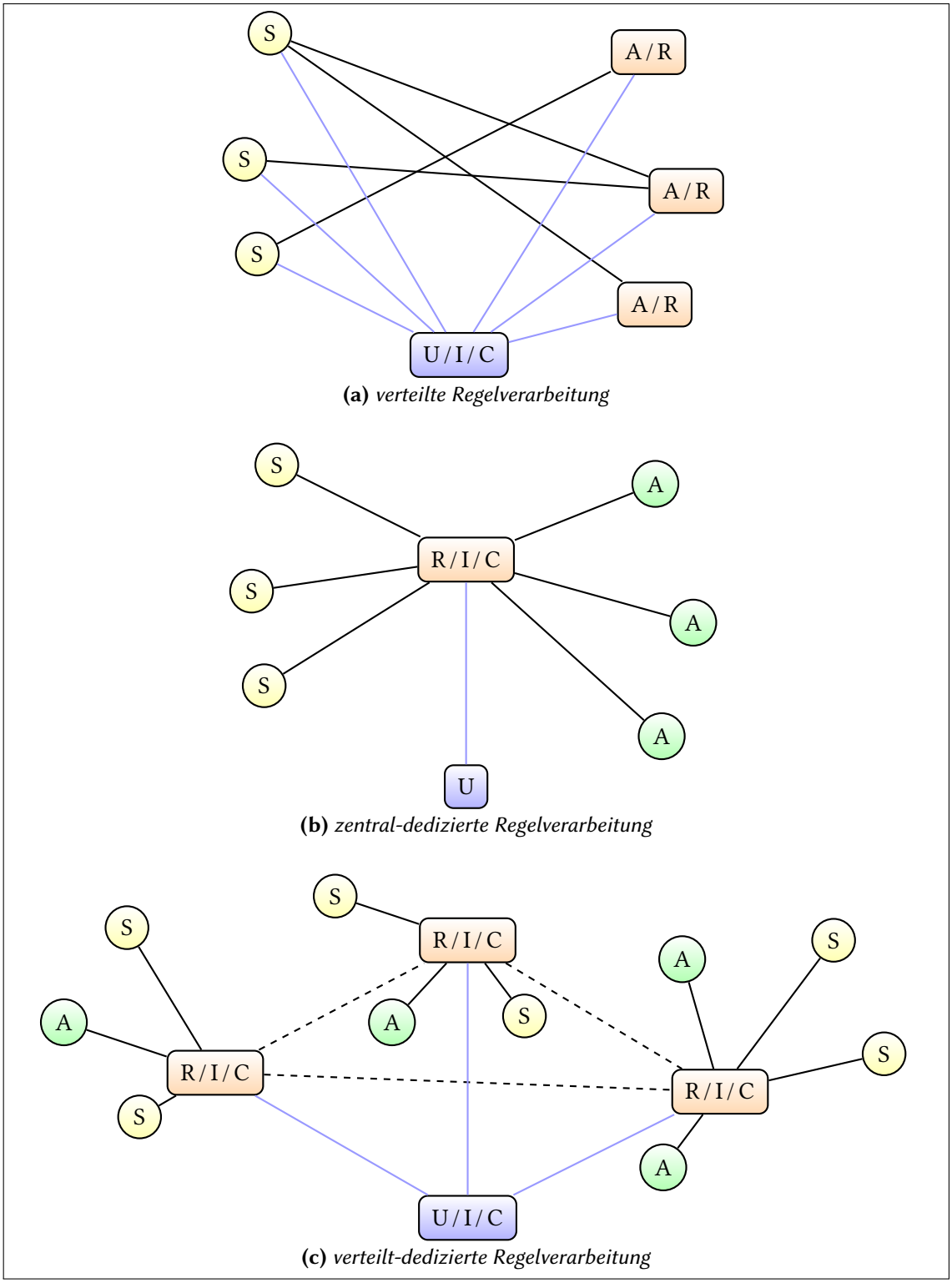
Durch die Art und Weise der Regelverarbeitung ergeben sich drei grundsätzlich verschiedene Ansätze zu dessen Gestaltung:

1. System-Architektur mit verteilter Regelverarbeitung,
2. System-Architektur mit zentral-dedizierter Regelverarbeitung und
3. System-Architektur mit verteilt-dedizierter Regelverarbeitung.

In den folgenden Abschnitten werden diese drei System-Architekturen beleuchtet und anhand eines Beispiels erklärt. Alle drei Beispiele sind in Abbildung 3.1 (auf Seite 25) zu sehen und haben eines gemeinsam: Der Hausbewohner nutzt einen Computer oder ein mobiles Endgerät, um mit dem Hausautomatisierungssystem zu interagieren. Dieser wird im Folgenden User-Netzknoten (U) genannt.

3.3.1. Verteilte Regelverarbeitung

Bei der verteilten Regelverarbeitung werden alle Regeln auf verschiedene Netzknoten verteilt. Diese Netzknoten besitzen neben der Regelwerkkomponente oft auch andere



(a) *verteilte Regelverarbeitung*

(b) *zentral-dedizierte Regelverarbeitung*

(c) *verteilt-dedizierte Regelverarbeitung*

Abbildung 3.1. *System-Architekturen für Hausautomatisierungssysteme.Abgebildet ist nur die Kommunikation zwischen den Endknoten des Hausautomatisierungssystems.*
U: User, S: Sensorknoten, A: Aktorknoten, R: Regelwerkknoten, I: Interface, C: Hausautomationskoordinator.

Komponenten.

Sensorknoten besitzen üblicherweise keine Regelverarbeitung, da sie meist batteriebetrieben sind und somit ein Duty-Cycling verwenden, um Energie zu sparen. Durch den Einsatz von Duty-Cycling in regelverarbeitenden Netzknoten ist dieser nur für eine geringe Zeit bereit, Automationsprozesse zu steuern, wodurch das Hausautomatisierungssystem ineffizient werden kann.

Beispiel

In Abbildung 3.1a (auf Seite 25) ist ein Beispiel eines Hausautomatisierungssystems aufgeführt, das eine System-Architektur mit verteilter Regelverarbeitung umsetzt.

In diesem Beispiel ist zu erkennen, dass die Komponenten der Managementebene im User-Netzknoten integriert sind. Dabei ist es dem Hausbewohner jederzeit möglich, das Interface zu benutzen. So kann er auf vorhandene Daten zugreifen oder Regeln hinzufügen, ändern oder löschen, ohne eine Netzwerkverbindung aufbauen zu müssen.

Bei Einsatz eines mobilen User-Netzknotens besteht allerdings das Problem, dass dieser nicht ständig eine Verbindung zum Hausautomatisierungsnetzwerk aufbauen kann (offline ist). Dieser Nachteil ist aber tolerierbar, da das Hausautomatisierungssystem eher selten mit dem Hausbewohner in Kontakt tritt.

Weiterhin ist es in solch einer System-Architektur sinnvoll, Regelwerk- und Aktorknoten in einem Netzknoten zu vereinen. Dann speichert jeder Aktorknoten nur diejenigen Regeln, die ihn betreffen. Der Nutzer-Netzknoten sorgt bei Neukonfigurationen für die nötige Organisation der Kommunikation, so dass die nötigen Sensorinformationen direkt an die Aktorknoten gesendet werden.

3.3.2. Zentral-dedizierte Regelverarbeitung

Bei der zentral-dedizierten Regelverarbeitung werden alle Regeln auf einem einzigen, zentralen Netzknoten untergebracht. Durch eine bevorzugte Abarbeitung des Automationsprozesses wird eine schnelle Regelverarbeitung garantiert.

Die dedizierte Eigenschaft besagt, dass dieser Netzknoten bis auf das Regelwerk keine weiteren Komponenten der Automationsebene integrieren muss. Weiterhin verläuft aufgrund der zentralen Steuerung die gesamte Kommunikation im System (auf Anwendungsschicht) über ihn.

Beispiel

In Abbildung 3.1b (auf Seite 25) wird ein Beispiel eines Hausautomatisierungssystems gezeigt, das eine System-Architektur mit zentral-dedizierter Regelverarbeitung umsetzt.

Da der Regelwerkknoten eine zentrale Instanz im System darstellt, ist es sinnvoll, die Komponenten der Managementebene mit ihm zu vereinen, so dass der Hausbewohner mit Hilfe eines *beliebigen* (mobilen) Endgerät mit dem Interface des Hausautomatisierungssystems in Kontakt treten kann. Diese Netzstruktur bietet zwei wichtige Vorteile: (1) Dem

User-Netzknoten müssen nicht alle Netzknoten des Hausautomatisierungssystems bekannt sein. Es reicht aus, dass er mit dem Regelwerkknoten kommunizieren und auf das Interface zugreifen kann. (2) Alle Daten, die das Interface zur Verfügung stellt, sind aktuell, da es sich auf einem Netzknoten befindet, der zentral am Automatisierungsprozess beteiligt ist.

Auch die Integration des Hausautomationskoordinators im Regelwerkknoten bietet den Vorteil, dass dieser nicht mit anderen Netzknoten kommunizieren muss, um Regeln im System zu verteilen oder zu verwalten.

3.3.3. Verteilt-dedizierte Regelverarbeitung

Bei der verteilt-dedizierten Regelverarbeitung kommen mehrere dedizierte Netzknoten zum Einsatz, die mit der Regelverarbeitung beauftragt sind. Dabei wird das Hausautomatisierungssystem in mehrere Subsysteme eingeteilt und jeder dieser Regelwerkknoten verwaltet sich um die Automationsprozesse in einem dieser Subsysteme. Jedes Subsystem für sich betrachtet bildet eine zentral-dedizierte Regelverarbeitung.

Betrachtet man nun das Gesamt-System, dann müssen nicht mehr sämtliche Sensorinformationen zu einem einzigen Regelwerkknoten hingeführt werden. Auch die Kommunikation mit den Aktoren ist nicht mehr von diesem abhängig. Durch die Einteilung des Systems in Subsysteme wird folglich das Risiko eines Gesamtausfalls auf die Subsysteme verteilt und damit minimiert. Dennoch bleibt der Vorteil der dedizierten Regelverarbeitung erhalten.

Beispiel

In Abbildung 3.1c (auf Seite 25) ist ein Beispiel eines Hausautomatisierungssystems aufgeführt, das eine System-Architektur mit verteilt-dedizierter Regelverarbeitung umsetzt.

Wie in dieser Abbildung zu sehen ist, existieren im System mehrere Regelwerkknoten, aber auch mehrere Interfaces und Hausautomatisierungskoordinatoren. Die letzten beiden, integriert im User-Netzknoten, organisieren die Subsysteme und stellen dem Hausbewohner eine Kommunikationsschnittstelle für das Gesamtsystem zur Verfügung. Während einer Regeländerung wird der Kontakt zu den Regelwerkknoten, die die Subsysteme organisieren, gesucht, so dass diese Änderungen auch entsprechend umgesetzt werden können. Wie im Fall der verteilten Regelverarbeitung bewirkt der Einsatz eines mobilen Endgeräts, dass dieser offline sein kann. Aber auch hier ist dieses Problem aus den gleichen Gründen vernachlässigbar.

Weiterhin kann es wichtig sein, dass jedes Subsystem getrennt von den anderen konfigurierbar ist. Dies kann beispielsweise in einer größeren Wohngemeinschaft der Fall sein: Jeder WG-Bewohner überwacht und verwaltet sein eigenes Subsystem. Gemeinsame Funktionen wie die Überwachung der Gemeinschaftsräume können wiederum von allen eingesehen werden.

Ferner kann es in anderen Fällen nötig sein, dass sich die einzelnen Subsysteme untereinander verständigen. So kann eventuell eine ausgefallene Verbindung zu einem Sensor-

oder Aktorknoten mit Hilfe eines anderen Subsystems wiederhergestellt werden. Dabei ist es ausreichend, wenn die Regelwerkknoten miteinander kommunizieren können.

Zudem ist der Einsatz von Subsystem-übergreifenden Regeln möglich. Ein Beispiel demonstriert eine solche Regel: In einem Subsystem hat der zuständige Regelwerkknoten anhand der Sensorinformationen der im Subsystem vorhandenen Sensorknoten schlussfolgern können, dass ein gravierender Fehler aufgetreten ist. Der gravierende Fehler wird zunächst als stiller Alarm an einen Aktorknoten gemeldet. Wenn nun solch ein gravierender Fehler in mehrere Subsystemen auftritt, greift folgende Subsystem-übergreifende Regel:

> *Wenn in mindestens drei Subsystemen ein gravierender Fehler ausgelöst wurde, so wird der normale Betrieb des Systems vorübergehend eingestellt und ein nicht-stiller Alarm gemeldet. Das System soll dann warten, bis dieser Fehlerzustand behoben worden ist.*

Da mit Erfüllung dieser Regel mehrere gravierende Fehler vorliegen, ist im System ein größeres Problem vorhanden. Daher werden sicherheitshalber alle Automationsprozesse eingestellt, bis die aufgetretenen Fehler behoben sind.

Spätestens an dieser Stelle ist es ersichtlich, dass eine verteilt-dedizierte Regelverarbeitung nur in komplexen Hausautomatisierungsszenarien sinnvoll Anwendung findet und die Kompetenzen der meisten Hausbewohner übersteigen wird. Daher wird diese System-Architektur auch vorwiegend in der Gebäudeautomatisierung eingesetzt.

3.4. Vergleich der System-Architekturen

Zusammengefasst sind System-Architekturen mit verteilter Regelverarbeitung und mit zentral-dedizierter Regelverarbeitung für kleinere Hausautomatisierungsszenarien gedacht. Bei größeren bzw. komplexen Hausautomatisierungsszenarien kommen dagegen eher System-Architekturen mit verteilt-dedizierter Regelverarbeitung zum Einsatz, weshalb sie – trotz ihres Vorteils, verteilt und doch dediziert Regeln zu verarbeiten – in den wengisten Wohnhäusern Anwendung finden.

Daher beschränkt sich diese Arbeit auf den Vergleich von System-Architekturen mit verteilter und zentral-dedizierter Regelverarbeitung. Letztere wird nachfolgend nur noch *dedizierte Regelverarbeitung* genannt, da nicht mehr zwischen „zentral“ und „verteilt“ unterschieden werden muss.

Vorweg sei bereits erwähnt, dass beide System-Architekturen sowohl Vor- wie auch Nachteile besitzen und dass jede von ihnen in bestimmten Anwendungsszenarien ihre Anwendung findet. Da die Sicherheitsarchitektur in der Hausautomatisierung für die Akzeptanz des Systems ein wichtiger Faktor darstellt (siehe Unterabschnitt 2.2.1), ist es beim Vergleichen angebracht, Aspekte, die die Sicherheitsarchitektur betreffen, in die Auswahl einer System-Architektur miteinzubeziehen.

3.4.1. Ausfall Regel-verarbeitender Netzknoten

Hausautomatisierungssysteme, die eine System-Architektur mit verteilter Regelverarbeitung besitzen, haben gegenüber solchen mit dedizierter Regelverarbeitung den großen Vorteil, dass bei Ausfall von Regel-verarbeitenden Netzknoten nicht sofort das gesamte System ausfällt, sondern nur diejenigen Automationsprozesse, die sich auf den ausfallenden Netzknoten beziehen. Denn es wird keine zentralisierte Netzinfrastruktur eingesetzt, sondern eine vermaschte

Dieser Vorteil geht allerdings mit der Notwendigkeit einher, dass das Kommunikationsverhalten der Netzknoten bei Änderungen am Regelwerk aufgrund der vermaschten Netzstruktur (siehe Abbildung 3.1a) neu organisiert werden muss. Durch Verwendung von Broadcast- bzw. Multicast-Nachrichten ließe sich dieser Nachteil kompensieren.

3.4.2. Leistungsfähigkeit Regel-verarbeitender Netzknoten

Im Sensornetz gilt der Grundsatz, dass alle Netzknoten energieeffizient arbeiten. Aus diesem Grund werden so gut wie alle Netzknoten leistungsschwach sein, um eine gewisse Energieeffizienz zu erzwingen.

Wenn System-Architekturen mit verteilter Regelverarbeitung zum Einsatz kommen und wenn mehrere Aktorknoten die Regelverarbeitung übernehmen (siehe Abbildung 3.1a), ist es sinnvoll, diese auch leistungsschwach zu gestalten, sofern möglich. Einige dieser mote-class Aktorknoten werden sogar batteriebetrieben sein (z. B. Heizungsregler).

In System-Architekturen mit dedizierter Regelverarbeitung existiert lediglich ein Regelwerkknoten. Folglich ist der Einsatz eines laptop-class Regelwerkknoten denkbar, da (1) einerseits alle anderen Netzknoten voraussichtlich leistungsschwach sind und (2) andererseits dieser genügend Ressourcen zur Verfügung hat, um das gesamte Hausautomatisierungsnetz zu verwalten und zu steuern. Der Einsatz von erzwungenen Energiesparmechanismen wie Duty-Cycling könnte den Automatisierungsprozess behindern.

3.4.3. Kommunikationsverhalten im Netz

Eine weitere zu untersuchende Eigenschaft der System-Architekturen ist die Analyse des Kommunikationsverhalten.

In einer System-Architektur mit verteilter Regelverarbeitung kommunizieren nur diejenigen Sensor- und Aktorknoten miteinander, die Informationen aufgrund vorhandener Regeln austauschen wollen. Allerdings kann es passieren, dass durch die vermaschte Netzinfrastruktur und dem Absenz einer zentralen Instanz der Ausfall einzelner Netzknoten unbemerkt bleibt. Das Hinzufügen neuer Sensor- oder Aktorknoten erfolgt dabei in zwei Schritten: Zuerst verbinden sie sich mit dem User-Netzknoten, da nur er die Informationen besitzt, mit wem sie kommunizieren müssen, damit der Automatisierungsprozess reibungslos abgearbeitet werden kann. Danach müssen sie kontrollieren, ob sie eine Verbindung zu den zugeordneten Aktor- bzw. Sensorknoten aufbauen können.

In einer System-Architektur mit dedizierter Regelverarbeitung dagegen kommunizieren alle Netzknoten mit dem Regelwerkknoten. Dieser bemerkt somit sofort, wenn einzelne

Netzknoten nicht antworten. Auch müssen Sensor- und Aktorknoten voneinander keine Kenntnis haben, damit der Automatisierungsprozess funktioniert. Das Hinzufügen neuer Sensor- oder Aktorknoten gestaltet sich im Gegensatz zu einer verteilten Regelverarbeitung einfacher: Die hinzuzufügenden Netzknoten bauen lediglich eine Verbindung zum Regelwerkknoten auf. Mehr ist nicht erforderlich.

Betrachtet man das Kommunikationsverhalten der Routingknoten, die Informationen der anderen Netzknoten lediglich weiterleiten, dann fällt auf, dass es in einer dedizierten Regelverarbeitung in der Nähe des Regelwerkknotens zu Engpässen kommen kann, da diese i. A. leistungsschwach sind. Abhilfe können intelligente Routingmechanismen ermöglichen, indem ausgelastete Routingknoten nicht in den Routingpfad einbezogen werden, sofern möglich. Die Alternative ist die Verwendung zusätzlicher Routingknoten oder eine bessere örtliche Verteilung der vorhandenen. Auch System-Architekturen mit verteilter Regelverarbeitung sind nicht vor solchen Problemen geschützt.

3.4.4. Resultierende Anforderungen an die Sicherheitsarchitektur

Die Akzeptanz des Hausautomatisierungssystems hängt ebenfalls von der Art und Weise ab, wie die Sicherheitsarchitektur gestaltet wird. Daher werden zuletzt – den Vergleich abschließend – Anforderungen an die Sicherheitsarchitektur diskutiert, die aus der System-Architektur resultieren. Dabei werden die aus dem bisherigen Vergleich ermittelten Erkenntnisse einbezogen.

Es sei dem nächsten Kapitel bereits vorweggenommen, dass der Aufbau neuer Verbindungen „angriffssicher“ erfolgen muss. Dieses ist beim Hinzufügen von Sensor-, Aktor- und Routingknoten erforderlich, aber auch dann, wenn eine solche Verbindung zwischen bereits im Netz befindlichen Netzknoten noch nicht existiert.

In einer System-Architektur mit dedizierter Regelverarbeitung tritt nur der erste Fall auf, weil beim Hinzufügen lediglich eine angriffssichere Verbindung zum Regelwerkknoten benötigt wird. Dagegen wird bei Verwendung einer verteilten Regelverarbeitung üblicherweise beides benötigt: Neue Netzknoten (ausgenommen Routingknoten) müssen sowohl zum User-Netzknoten eine angriffssichere Verbindung aufbauen, aber auch zu den ihnen zugeordneten Sensor- und Aktorknoten. Gravierenden Änderungen am Regelwerk können auch eine Neuorganisation der Kommunikation zur Folge haben, so dass wiederum neue angriffssichere Verbindungen aufgebaut werden müssten. Diese vermaschte Netzstruktur besitzt folglich einen entscheidenen Nachteil gegenüber einer zentralisierten.

Allerdings bietet die Zentralisierte für den Angreifer interessante Angriffsziele. Denn in der Nähe des Regelwerkknotens verläuft die höhste Informationsmenge im gesamten Hausautomatisierungssystem. Auf der anderen Seite wird bei der verteilten Regelverarbeitung das Versenden von „angriffssicheren“ Broadcast-/Multicastnachrichten ebenfalls weitere Probleme bereiten, sofern solche Nachrichten zur Organisation des Netzes erforderlich sind.

3.5. Verwandte Hausautomatisierungsprojekte

Da der theoretische Vergleich der verschiedenen System-Architekturen abgeschlossen ist, folgt in diesem Abschnitt eine Vorstellung ausgewählter Hausautomatisierungsprojekte. Sie sollen den theoretischen Vergleich um praktische Beispiele zur aktuellen Forschung ergänzen.

3.5.1. HexaBus Home Automation System

Das HexaBys Home Automation System [55], [56] ist eine offene Hausautomatisierungslösung[1]. Sowohl Hardware wie auch Software sind Open-Source, so dass keine Lizenzkosten entstehen. Das kommt der gewünschten Interoperabilität in der offenen Hausautomatisierung sehr entgegen. Dieses Open-Source-Projekt ist im Rahmen des BMBF-Projekts „mySmartGrid“ [57] des Frauenhofer ITWM entstanden.

Alle Geräte im HexaBus System nutzen zur Umsetzung einer internetfähigen Hausautomatisierung IPv6 over Low power Wireless Personal Area Networks (6LoWPAN). Derzeitige Geräte unterstützen den Funkbereich um 868 MHz. Zur Realisierung einer energiesparenden Kommunikation wird als Transportprotokoll UDP (ggf. inkl. Header Compression) eingesetzt. Auf UDP aufbauend kommt ein eigenes Anwendungsprotokoll zur Anwendung, das sich *HexaBus Protocol* nennt. Dieses kümmert sich um die Kommunikation zwischen Sensor- und Aktorknoten. Ob und welche Regeln neu hinzugefügte Netzknoten verarbeiten, ist vorher festgelegt worden.

Es spielt aber keine Rolle, von welchen Netzknoten Regeln verarbeitet werden, da jede Hausautomatisierungsnachricht vorzugsweise per Broadcast (Multicast) an das gesamte Sensornetz versendet wird. Damit entfällt einerseits die Organisation des Netzwerkes durch den Hausautomationskoordinator, da jeder jede Nachricht erhält. Andererseits hat dieses Kommunikationsverhalten ein so genanntes *Overhearing* zur Folge, wodurch mote-class Netzknoten in ihrem Schlafrhythmus gestört werden, weil sie Nachrichten verarbeiten, die sie nicht interessieren. Andererseits können mehrere Empfänger mit nur einer Nachricht erreicht werden.

Da im HexaBus System keine zentrale Instanz zur Abarbeitung des Automatisierungsprozesses vorgesehen ist, liegt eine System-Architektur mit **verteiler Regelverarbeitung** vor. Im Gegensatz zum Beispiel aus Abbildung 3.1a (siehe Seite 25) ist es in diesem System möglich, Regeln sowohl in Sensor-, Aktor- wie auch in Routingknoten abzuarbeiten.

3.5.2. Fhem

Fhem beschreibt sich selbst als „ein in perl geschriebener, GPL lizensierter Server für die [offene] Heimautomatisierung“ [70]. Im Gegensatz zum HexaBus System, das ein eigenes

[1] Da nur wenig veröffentlichte, wissenschaftliche Dokumentation gefunden wurde, dient der veröffentlichte Source Code wie auch das dazugehörige Wiki in dieser Arbeit als Diskussionsgrundlage.

Tabelle 3.1. *Eigenschaften der verteilten im Vergleich zur dedizierten Regelverarbeitung.*

Eigenschaft	Regelverarbeitung	
	verteilt	dediziert
Infrastruktur	vermascht	zentralisiert
Ausfall Regel-verarbeitender Netzknoten	tolerierbar	Gesamtausfall
Leistungsfähigkeit Regel-verarbeitenden Netzknoten	Mote-class	Laptop-class
Kommunikation Sensor – Aktor	Unicast (direkt)	Unicast (indirekt über den Regelwerkknoten)
Kommunikation bei Neuorganisation	Unicast oder Multicast	nicht nötig

Anwendungsprotokoll auf den Sensor- und Aktorknoten einsetzt, gilt bei Fhem eine andere Zielstellung: Das Ziel ist es, verschiedene bereits vorhandene Hausautomatisierungslösungen unter der Kontrolle eines zentralen Servers zu vereinen. Das bedeutet, dass beispielsweise sowohl Sensorknoten mit dem HomeMatic-System wie auch Aktorknoten mit dem FS20-System in einem Hausautomatisierungsnetzwerk vereint werden können. Hierzu werden zur Integration der einzelnen Hausautomatisierungssysteme so genannte *Module* entwickelt. Damit kümmert sich dieses Projekt nicht um die Hardware und deren Firmware, sondern nur um die Erstellung eines **Interfaces** und die Art und Weise der Regelverarbeitung.

Vernachlässigt man den Fakt, dass unterschiedliche Hausautomatisierungssysteme vereint werden, bildet dieses Netzwerk eine System-Architektur mit **dedizierter Regelverarbeitung**, in welchem der Fhem Server als Regelwerkknoten zum Einsatz kommt. Wenn dieses Projekt Angriffssicherheit bieten möchte, ist es erforderlich, nur solche Hausautomatisierungslösungen einzusetzen, die eine Angriffssicherheit bieten.

3.5.3. smarthomatic

Smarthomatic (kurz SHC) ist ein Hausautomatisierungssystem, das auf Open-Source basiert, Sicherheit bieten möchte und Erweiterbarkeit zum Ziel hat [71]. Dieses Projekt war früher unter dem Namen Open Home Control bekannt [72].

Das Projekt konzentriert sich auf das Entwickeln von **Hardware** und deren **Firmware**, aber nicht auf die Erstellung von zugehörigen Interfaces oder die Art und Weise der Regelverarbeitung. Dies wird anderen Projekten wie dem Fhem überlassen. Zur Zeit ist der Einsatz des Verschlüsselungsverfahrens AES-256 in Verbindung mit CRC32 geplant. Da allerdings das zu verwendende Anwendungsprotokoll (Communication Protocol) sich noch im Aufbau befindet, wird dieses Projekt nicht weiter untersucht.

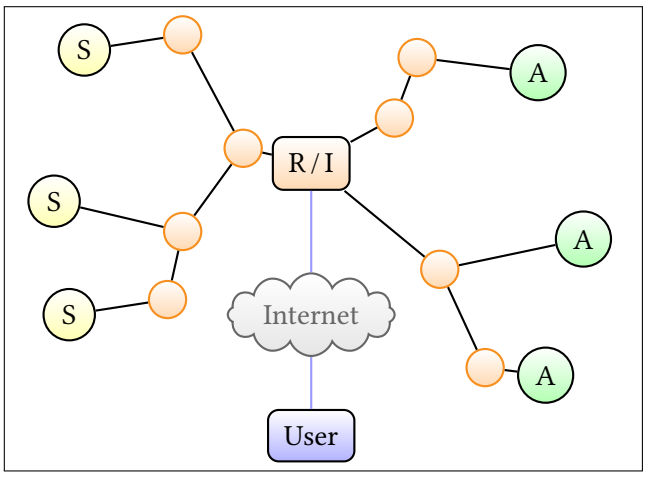


Abbildung 3.2. *Das Hausautomatisierungssystem.*

Sensor- (S) und Aktorknoten (A) kommunizieren ausschließlich mit dem Regelwerkknoten (R / I), um Informationen zur Steuerung des Automatisierungsprozesses auszutauschen. Es kommt gelegentlich vor, dass der Nutzer (User) den Automatisierungsprozess überwachen möchte und greift daher auf das Interface (R / I) zu.

3.6. Das ausgewählte Hausautomatisierungssystem

Dieses Kapitel abschließend, werden die Eigenschaften der verteilten Regelverarbeitung im Vergleich zur dedizierten Regelverarbeitung in Tabelle 3.1 zusammengefasst. Der Vergleich hat ergeben, dass beide Regelverarbeitungen abhängig vom jeweiligen Hausautomatisierungsszenario angewendet werden können, ohne die geforderten Voraussetzungen und Anforderungen zu verletzen. Auch verwandte Projekte haben gezeigt, dass diese beiden Regelverarbeitungen bereits zum Einsatz kommen.

Da kein genaues Anwendungsprotokoll für die Hausautomatisierungsanwendung vorausgesetzt war, wird die Sicherheitsarchitektur für eines dieser System-Architekturen entwickelt. Dabei wird die Art und Weise der Umsetzung der technischen Anforderungen an die Hausautomatisierung (Energieeffizienz und Skalierbarkeit) am stärksten berücksichtigt.

Das Verwenden von Multicast-Nachrichten zur Verteilung der Hausautomatisierungsnachrichten – wie es im HexaBus System zum Einsatz kommt – verursacht ein Overhearing bei den meisten Netzknoten, so dass die Anforderung nach Energieeffizienz nicht hinreichend erfüllt ist. Zudem bereitet die Gewährleistung der Vertraulichkeit von Broadcast- / Multicast-Nachrichten in Sensornetzen größere Probleme wie die von Unicast-Nachrichten.

Bei Verwendung von Unicast-Nachrichten wird allerdings ein Hausautomatisierungskoordinator benötigt, der die zur Umsetzung des Automatisierungsprozesses nötigen Kommunikationen organisiert. In einer System-Architektur mit verteilter Regelverarbeitung ist diese Organisation aufgrund der vermaschten Netzstruktur mit Kommunikationsaufwand verbunden. In der zentralisierten Netzstruktur entfällt dieses.

Auch das Hinzufügen neuer Netzknoten gestaltet sich in der Zentralisierten einfacher, da diese lediglich mit dem Regelwerkknoten kommunizieren. Allerdings bleibt das Problem, dass das Gesamtsystem bei Ausfall des Regelwerkknotens zum Erliegen kommt. Diese Problematik ist vernachlässigbar, da der Regelwerkknoten leistungsstark gestaltet wird und somit nur selten (maximal einmal alle drei Jahre) ausfallen sollte.

Aus diesen Gründen wird die Sicherheitsarchitektur für ein Hausautomatisierungssystem mit einer System-Architektur, die eine dedizierte Regelverarbeitung besitzt, konzipiert. Abbildung 3.2 zeigt dieses ausgewählte Hausautomatisierungssystem einschließlich Routingknoten. Damit kann die in dieser Arbeit angestrebte, *angriffssichere* Hausautomatisierungslösung beispielsweise in das Projekt **Fhem** integriert werden. Ein weiterer Vorteil ist es, dass die entwickelte Sicherheitsarchitektur auch auf ein System mit verteilt-dedizierter Regelverarbeitung übertragen werden kann.

4 Sichere Kommunikation im Internet der Dinge

To support the large number of emerging applications for smart objects, the underlying networking technology must be inherently scalable, interoperable, and have a solid standardization base to support future innovation as the application space grows.

—IPSO Alliance (2013)

Eine Kommunikation im Internet der Dinge vor dem Belauschen oder der Manipulation jeglicher Art zu schützen, stellt eine Herausforderung dar. Im Speziellen gilt dies für Netzknoten mit eingeschränkten Ressourcen — wie sie in drahtlosen Sensornetzen zum Einsatz kommen.

Die Anforderung nach einer sicheren und zugleich energiesparenden Kommunikation ist für gewöhnlich widersprüchlich, weshalb in den letzten 10 Jahren sehr viel zum Gebiet der *drahtlosen Sensornetze* geforscht wurde. Zu Anfang wurden sehr viele neue Kommunikationprotokolle entwickelt und mit ihnen neue Sicherheitsmaßnahmen. Diese sind allerdings auf spezielle Anwendungsgebiete und ihren Umgebungen beschränkt, so dass sie nicht auf triviale Weise in anderen Anwendungsgebieten eingesetzt werden können. [35]

Mit dem Entwickeln des Internets der Dinge werden immer mehr leichtgewichtige, aber allgemeingültige Kommunikations- und Sicherheitslösungen gesucht. Ein vielversprechender Ansatz ist es, etablierte und standardisierte Lösungen klassischer Netze auf die veränderten Anforderungen des Internets der Dinge hin anzupassen.

4.1. Einführung in das Umfeld der IT

Es gibt verschiedene Begriffe zur Sicherheit und dessen Teilbereiche im Umfeld der Informationstechnik (IT). Daher wird an dieser Stelle ein kleiner Überblick zu wichtigen Defintionen gegeben.

Diese und weiterführende Grundlagen können beispielsweise dem Leitfaden für Informationssicherheit vom Bundesamt für Sicherheit in der Informationstechnik (BSI) [7]

entnommen werden. Weiterhin können Begriffe aus dem Umfeld Internet und Sicherheit auch im Internet Security Glossar (RFC 4949 [14]) nachgeschlagen werden.

4.1.1. Sicherheit

Der Sicherheitsbegriff in der IT ist nicht eindeutig definiert. Zwei der verbreitesten Defintionen sind Folgende:

„Sicherheit beschreibt einen Zustand, der frei von unvertretbaren Risiken der Beeinträchtigung ist oder als gefahrenfrei angesehen wird." Diese Defintion ist beispielsweise in der technischen Definition der DIN 65108 verankert.

Weiterhin gilt eine zweite Defintion zum Begriff Sicherheit: „Allgemein wird Sicherheit als relativer Zustand der Gefahrenfreiheit angesehen, der stets nur für einen bestimmten Zeitraum, eine bestimmte Umgebung oder unter bestimmten Bedingungen gegeben ist." Diese Defintion trifft auch auf das Internet (der Dinge) zu: Heute kann ein System Sicherheit bieten. Doch mit fortschreitendem Stand der Technik nimmt diese Sicherheit ab, bis sie unsicher wird.

Der Begriff Sicherheit wird unterschieden in Betriebssicherheit und Angriffssicherheit. Manchmal kommt auch der Begriff Computersicherheit vor.

4.1.2. Betriebssicherheit (Funktionssicherheit)

Die Betriebssicherheit (auch Funktionssicherheit, engl. *safety*) ist das Erkennen und Beseitigen von Störungen, welche eine korrekte Funktionalität beeinträchtigen. Diese Störungen treten i. A. zufällig auf.

Die Betriebssicherheit beschäftigt sich mit der Frage nach der *Zuverlässigkeit* eines Systems. Sie garantiert die richtige Lauffähigkeit bestimmter vorher definierter Funktionalitäten und gibt an, wie im Fehlerfall vorgegangen wird, sobald ein Fehler nicht mehr toleriert werden kann.

4.1.3. Angriffssicherheit (Informationssicherheit)

Angriffssicherheit (auch Informationssicherheit, IT-Sicherheit, engl. *security*) ist das Erkennen und Abwehren von gezielten Angriffen, wobei von intelligenten Angreifern ausgegangen werden muss.

Sie kümmert sich um die *Vertraulichkeit*, *Integrität* sowie um die *Verfügbarkeit* von Informationen jeglicher Art in einem System. Dabei werden Maßnahmen beschrieben, um Risiken auf ein Minimum zu reduzieren.

4.1.4. Computersicherheit

Die Bedeutung des Begriffs Computersicherheit hat sich im Gegensatz zur Betriebs- oder Angriffssicherheit in den letzten Jahrzehnten gewandelt. Am Anfang des Computerzeitalters gab es noch keine großflächige Vernetzung. Daher wurde mit Computersicherheit üblicherweise die Betriebssicherheit gemeint.

Durch die weltweite Vernetzung und der immer komplexer werdenen Technik ist das Risiko deutlich gestiegen, einem Angriff ausgesetzt zu sein. In der heutigen Zeit wird daher unter Computersicherheit eher die Angriffssicherheit verstanden.

4.1.5. Sicherheitsarchitektur

Die Sicherheitsarchitektur definiert Schutzmaßnahmen und Schutzmechanismen, die aus den Sicherheitsanforderungen an einem System entstanden sind.

Aufgrund zunehmender Vernetzung und der daraus resultierenden Möglichkeiten für Angriffe auf Computersysteme (z. B. Botnetze) gewinnt die Angriffssicherheit immer mehr an Bedeutung. Deswegen wird unter Sicherheitsarchitektur i. A. eine IT-Sicherheitsarchitektur (engl. *security architecture*) verstanden.

4.1.6. Authenticated Encryption with Associated Data (AEAD)

Verschlüsselung ohne Authentifizierung beinhaltet die Gefahr, dass die Verschlüsselung umgangen und ein anderer Nachrichteninhalt eingesetzt werden kann, so dass es sinnvoll ist, Verschlüsselung in Kombination mit Authentifikation unter Verwendung eines Schlüssels in einem so genannten Authenticated Encryption (AE) Schema zu vereinen [8].

Werden unterschiedliche Schlüssel für Verschlüsselung und Authentifizierung eingesetzt, besteht das Problem, dass zum einen der Speicheraufwand durch die Verwendung unterschiedlicher Schlüssel erhöht ist und dass zum anderen die Performance durch das mehrmalige Durchlaufen aufgrund der getrennten Verschlüsselung sowie Authentifizierung nicht optimal ist. Andererseits gilt das Benutzen desselben Schlüssels sowohl für die Verschlüsselung wie auch für die Generierung des Message Authentication Code als gefährlich und muss hinreichend untersucht werden.

Eine häufige Anforderung bei sicheren Netzwerkprotokollen ist es, dass nicht nur der Nachrichtenpayload authentisch verschlüsselt werden soll, sondern dass auch der Header authentisch sein soll. Allerdings kann er i. A. nicht verschlüsselt werden, da Router diesen zwecks Weiterleitung verstehen müssen. Daher werden alle Algorithmen, die eine Nachricht inklusive assoziierter Daten authentifizieren, aber nur den Nachrichtenpayload verschlüsseln, als *Authenticated Encryption with Associated Data (AEAD)* Schema bezeichnet.

Im Allgemeinen ist ein AEAD-Schema nach RFC 5116 [15] ein Paar (E, D) von deterministischen Algorithmen, wobei E eine Verschlüsselungsfunktion und D eine Entschlüsselungsfunktion ist:

$$E : \text{Key} \times \text{Nonce} \times \text{Header} \times \text{Message} \to \text{Ciphertext} \times \text{Auth-Tag}$$
$$D : \text{Key} \times \text{Nonce} \times \text{Header} \times \text{Ciphertext} \times \text{Auth-Tag} \to \text{Message} \cup (\text{Fail})$$

Der Schlüssel ist ein Geheimnis zweier Kommunikationsteilnehmer. Die *Nonce* ist ein Initialisierungsvektor (IV), der nicht zufällig gewählt sein muss — ein Counter reicht

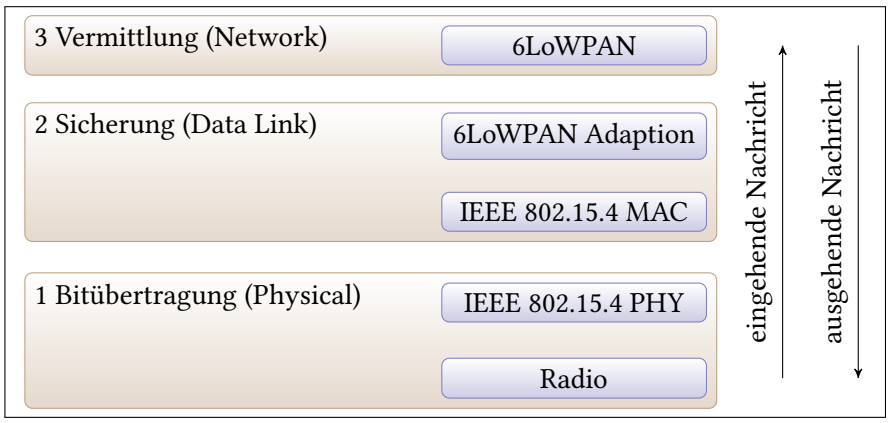


Abbildung 4.1. 6LoWPAN Netzwerk-Stack. *Abgebildet sind die im 6LoWPAN Standard [19] definierten Netzwerk-Schichten (nach dem OSI-Referenzmodel).*

oft. Der *Header* bildet die zusätzlichen assoziierten Daten, auch *Additional Associated Data (AAD)* genannt. Diese werden authentisch, aber unverschlüsselt übertragen. Der *Auth-Tag* entspricht dem Message Authentication Code.

Bekannte AEAD Block Cipher Modes of Operation sind Counter with Cipher Block Chaining-Message Authentication Code (CCM) [16] und Galois Counter Mode (GCM) [17]. Beide gelten als patentfrei. GCM ist zudem noch in Software wie auch in Hardware parallelisierbar. Sie können zusammen mit dem Advanced Encryption Standard (AES) [18] eingesetzt werden.

4.2. Netzwerktechnologien

Dieser Abschnitt behandelt wichtige Grundlagen, die zum Verständnis der Konzepterstellung beitragen. Grundlagen zu IPv6 sind im Unterabschnitt B.1.1 zu finden, so dass mit dem IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) Protokoll begonnen wird. Dann wird in Kürze das Routing-Protokoll vorgestellt, welches ausführlicher im Unterabschnitt B.1.2 beschrieben ist. Danach folgt eine Einführung in die Verschlüsselungsprotokolle IPsec sowie TLS und DTLS. Mit diesen Grundlagen können dann leichtgewichtige Sicherheitslösungen zur Kommunikation im Internet der Dinge besprochen werden.

4.2.1. IPv6 for Low-power Wireless Personal Area Networks (6LoWPAN)

Das IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) Protocol hat die Zielstellung, Funktechnologien zu nutzen, die eine kurze Reichweite sowie eine niedrige Übertragungsrate besitzen, wenig Strom verbrauchen und folglich geringe Anschaffungskosten aufweisen, um über diese Technologien IPv6 Packete versenden und empfangen zu können.

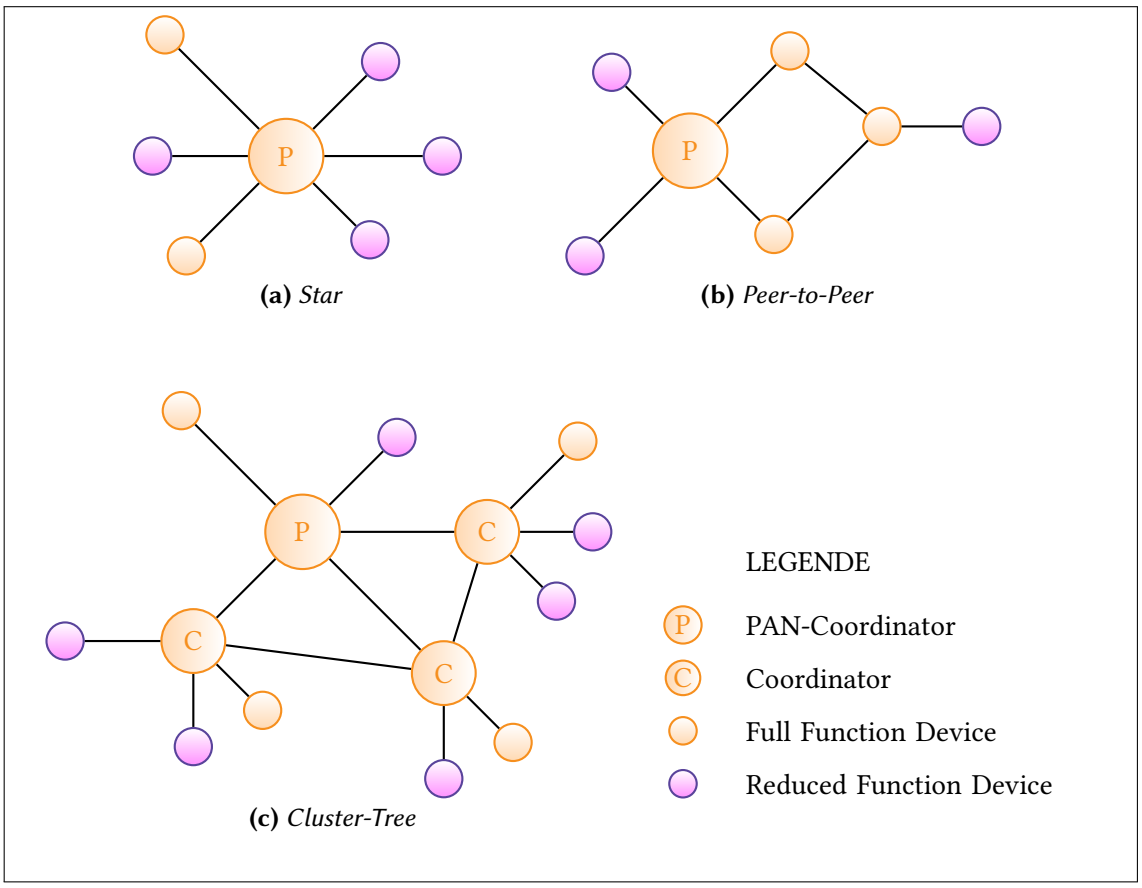


Abbildung 4.2. *Drei verschiedene Netztopologien im IEEE 802.15.4 Netzwerk.*

Zu diesen Funktechnologien zählt insbesondere der IEEE 802.15.4-2003 Standard [20]. Abbildung 4.1 zeigt den Aufbau des Netzwerk-Stacks unter Nutzung von IEEE 802.15.4-2003 Funkhardware (Radios). Hierbei wird das Problem gelöst, dass IPv6 Pakete in mehreren 802.15.4 Frames transportiert werden müssen. Denn IPv6 Pakete sind per Standard mindestens 1280 B[1] lang, aber 802.15.4 Frames dürfen lediglich eine maximale Länge von 127 B aufweisen. So stellt die 6LoWPAN Adaption Schicht eine transparente Fragmentierung bereit, um eine virtuelle MTU von 1280 B zu schaffen.

6LoWPAN-Netzwerke unterstützen verschiedene Netzwerk-Topologien, unter denen sich auch eine Stern- und eine Mesh-Topologie befindet. Abbildung 4.2 zeigt die Netzwerk-Topologien, die in 802.15.4-Netzwerken vorkommen. Betrachtet man die Kommunikation in Abbildung 3.2 (siehe Seite 33), dann erkennt man eine Peer-to-Peer-Kommunikation. Solche Netzwerk-Topologien implizieren ein Multi-Hop-Routing.

In solchen Netzwerk-Topologien kann das Routing Protocol for Low-power and Lossy Networks (RPL) eingesetzt werden. Im Gegensatz zu anderen Protokollen wurde RPL standardisiert und wird zur Zeit als Routingprotokoll für das Internet der Dinge favori-

[1] In dieser Arbeit wird Byte und Octet gleichbedeutend verwendet: $1\,\mathrm{B} = 8\,\mathrm{bit} = 1$ octet. Es sei angemerkt, dass es Computersysteme gibt, bei denen $1\ \mathrm{B} \neq 8$ bit ist.

siert, weshalb es in dieser Arbeit nicht unbeachtet bleiben darf. Dennoch wird bei der Konzipierung der Sicherheitsarchitektur versucht, nur allgemein auf Routingprobleme einzugehen, falls in Zukunft ein anderes Routingprotokoll eingesetzt werden soll.

4.2.2. Routing Protocol for Low-power and Lossy Networks (RPL)

Das Routing Protocol for Low-power and Lossy Networks (RPL) (RFC 6550 [21]) ist ein dynamisches Routing-Protokoll und ist speziell für LLNs, zu denen auch drahtlose Sensornetze mit 6LoWPAN zählen, entwickelt worden. Es ist ein Route-Over Routing-Protokoll, d. h., es routet auf der Netzwerkschicht des OSI-Referenzmodells. Jeder Knoten besitzt hierbei eine eigene IPv6-Adresse, die i. A. zustandslos (stateless) aus der MAC-Adresse des Netzwerkadapters gebildet wird.

Weiterführende Grundlagen zu RPL können im Unterabschnitt B.1.2 nachgeschlagen werden. Dort ist beschrieben, welche Vorteile RPL gegenüber anderen Routing-Protokollen bietet. An dieser Stelle sei angemerkt, dass RPL auch eigene Sicherheitsmechanismen mitbringt, die alternativ zu denen der Sicherungsschicht eingesetzt werden können. Allerdings sind die RPL-Sicherheitsmechanismen noch unausgereift, so wie es in „RPL in a nutshell: A survey“ wie folgt beschrieben wird:

> Security in RPL is immature. It is indeed important to define security mechanisms such as key management and authentication techniques to secure the join of an instance in the network. The process by which the key is received by a node as well as the key establishment and maintenance processes are not specified. In addition, the configuration of security mechanisms for the processing of the incoming packets is still an open issue. [36, Section 7]

Wenn sie ausgereift sind, sollten sie in Bezug auf die in dieser Arbeit entwickelte Sicherheitsarchitektur überprüft werden. Daher wird auf eine detailliertere Betrachtung der RPL-Sicherheitsmechanismen verzichtet und eine alternative Lösung gesucht.

4.2.3. Internet Protocol Security (IPsec)

Internet Protokoll Security (IPsec) – definiert in RFC 4301 [22] – ist ein Verschlüsselungsprotokoll auf Vermittlungsschicht. Es stellt keine eigenständige Schicht im OSI-Referenzmodel bereit, sondern erweitert das Internet Protokoll um Verschlüsselungs- wie auch Authentisierungsmechanismen. Es wurde als integraler Bestandteil für IPv6 entwickelt und dann auch in IPv4 einsatzfähig gemacht.

Mit der Internet Protokoll Security ist es möglich, Datenpakete auf Vermittlungsschicht, unabhängig ihrer Anwendungsprozesse, in transparenter Weise zu verschlüsseln oder zu authentisieren. Andererseits können dadurch Datenpakete unterschiedlicher Prozesse nicht unterschieden werden, wodurch Sicherheitsanforderungen unterschiedlicher Prozesse nicht berücksichtigt werden können.

Allgemeine Funktionsweise

Ziel von IPsec ist es, eine Sicherheitsbeziehung (engl. *security association*) zwischen zwei kommunizierenden Netzteilnehmern über ungesicherte Netze zu ermöglichen. Diese Sicherheitsbeziehung muss nicht zwischen Sender und Empfänger hergestellt werden. Oft reicht es aus, wenn zwischen zwei Routern solch eine Beziehung aufgebaut wird, da man davon ausgehen kann, dass im lokalen Netz die Verbindung vertrauenswürdig ist.

IPsec verwendet drei Protokolle, um dies zu erreichen:

- Encapsulating Security Payload (ESP) – zum Schutz der Vertraulichkeit und der Datenintegrität
- Authentificaation Header (AH) – zur Authentisierung des Paketursprungs und zum Schutz der Datenintegrität
- Internet Key Exchange (IKE) – zum Schlüsselaustausch

Die ersten beiden Protokolle (ESP und AH) können sowohl einzeln wie auch kombiniert eingesetzt werden. Desweiteren existieren zwei Betriebsmodi: der Tunnelmodus und der Transportmodus:

Transportmodus Dieser Modus kommt zum Einsatz, wenn A mit B vertraulich kommunizieren möchte und A niemandem auf diesem Wege traut (selbst nicht im eigenen Netzwerk). Dann müssen jedoch A und B die Sicherungsdaten erzeugen können. Zudem wird der IP-Header ungeschützt versendet.

Tunnelmodus Dieser Modus kommt dann zum Einsatz, wenn zwar A mit B kommunizieren möchte, aber es ausreicht, wenn das IP-Paket nur über das unsichere Netz geschützt ist. In solch einem Fall kommen IP-Gateways zur Anwendung: eins (GW_A), das das Paket von A verschlüsselt, und eins (GW_B), das das geschützt Paket entschlüsselt und an B weiterleitet. Im Tunnel (zwischen GW_A und GW_B) ist das IP-Paket von A inklusive IP-Header geschützt. Auf diese Weise wird eine Verkehrsflussanalyse erschwert. Auch ermöglicht dieser Modus, dass Firewalls diese Pakete einsehen und ggf. filtern können.

Das Internet Key Exchange (IKE) Protokoll, das in Version 2 vorliegt und seit 2010 in RFC 5996 [23] definiert ist (und damit RFC 4306 [24] außer Gebrauch setzt), ist für die Ausführung des Schlüsselaustauschs zwischen zwei Kommunikationspartner zuständig. Es kommen X.509 Zertifikate zum Einsatz, die entweder bereits bekannt sind (pre-shared) oder mittels DNS verteilt werden. Ein Diffie-Hellman-Schlüsselaustausch lässt beide Kommunikationspartner ein gemeinsames Geheimnis generieren, aus welchem die Schlüssel zur sicheren Kommunikation abgeleitet werden. Das IKE-Protokoll selbst ist nicht Teil der Vermittlungsschicht, sondern nutzt UDP als Transportprotokoll.

Auf eine detailliertere Beschreibung wird an dieser Stelle verzichtet, da nur ein Einblick in die Funktionsweise gegeben werden soll. Weiterführende Erläuterungen können beispielsweise in [9] und [10] nachgeschlagen werden.

4.2.4. Transport Layer Security (TLS)

Transport Layer Security (TLS) – definiert in RFC 5246 [25], ehemals Secure Sockets Layer (SSL) – ist ein Verschlüsselungsprotokoll, basierend auf dem Client-Server-Prinzip. Prozesse (wie HTTPS) können mittels TLS ihre Daten auf Transportschicht[2] auf transparente Weise verschlüsseln oder authentisieren. Dabei soll die Interoperabilität und die Erweiterbarkeit nicht eingeschränkt werden.

Interoperabilität im Zusammenhang mit TLS bedeutet, dass Anwendungen, die Transport Layer Security (TLS) verwenden, in der Lage sind, kryptograpische Parameter auf sichere Weise zu verhandeln. Dabei ist es unwichtig, ob die Programmierer dieser Anwendungen den Quellcode der anderen kennen. Auch die Erweiterbarkeit soll in TLS nicht vernachlässigt werden, da sich jede Technologie weiterentwickelt. Hiermit soll verhindert werden, dass neue Methoden in der Kryptographie zu einem neuen Protokoll oder sogar zu einer komplett neuen Sicherheitsbibliothek führen, wodurch ältere Anwendungen ggf. unbrauchbar würden.

Weiterhin ist die relative Effizienz in TLS wichtig: Kryptographische Operationen sind oft mit hohen CPU-Kosten verbunden. Durch Verwendung eines optionales Session-Cachings kann eine bereits aufgebaute Verbindung wiederverwendet werden, sofern der TLS-Server dies unterstützt. So ist es möglich, die Anzahl der Verbindungsaufbauten und damit die Kommunikation im Netzwerk zu senken.

Zielstellung ist es, eine Ende-zu-Ende-Verschlüsselung auf Transportschicht zu ermögliche.

An dieser Stelle wird auf eine Unterscheidung zwischen den Versionen verzichtet und nur auf die aktuelle Version 1.2 eingegangen, da ab dieser die Authenticated Encryption with Associated Data (AEAD) Verfahren unterstützt werden.

Allgemeine Funktionsweise

TLS ist ein hybrides Verschlüsselungsprotokoll, d. h., es verwendet ein asymmetrisches kryptographisches Verfahren, damit sich Server und Client auf einen symmetrischen Schlüssel einigen können. Alternativ kann auch ein Pre-Shared-Key (PSK) zum Einsatz kommen, der bereits auf anderem Wege ausgetauscht wurde.

Mit diesen (ausgetauschten) Informationen wird ein Pre-Master-Secret zusammengesetzt. Aus diesem wird danach, abhängig von der Cipher Suite spezifizierten Pseudozufallsfunktion (PRF), das Master-Secret generiert, woraus dann die Schlüssel zur Sicherung der Vertraulichkeit und Datenintegrität abgeleitet werden. Typischerweise entstehen zwei unterschiedliche Schlüssel: einer für das Versenden und einer für das Empfangen.

Damit sich Server und Client auf eine so genannte Cipher Suite – also ein Schlüsselaustauschs- und ein Verschlüsselungsverfahren – einigen können, wird ein Handshake zu Beginn jeder Kommunikation durchgeführt. Nach dem Handshake können dann

[2] Es sei darauf hingewiesen, dass in mancher Literatur TLS auf OSI-Schicht 5 (Sitzungsschicht) eingeordnet wird, da es sich zwischen dem Transportprotokoll und dem Anwendungsprotokoll befindet.

Tabelle 4.1. *Ausgewählte TLS Cipher Suites [26].*

Verschlüsselung		Schlüsselaustausch		IANA-Index	DTLS-OK
Vertraulichkeit	Integrität	Methode	Signatur		
AES-128-GCM-SHA256[1]		RSA	RSA	00 9c	Y
		DHE	RSA	00 9e	Y
		DH	RSA	00 a0	Y
		DHE	DSS	00 a2	Y
		DH	DSS	00 a4	Y
		DH-anon	—	00 a6	Y
		PSK	—	00 a8	Y
		DHE-PSK	—	00 aa	Y
		RSA-PSK	—	00 ac	Y
AES-128-CCM-8[2]		RSA	RSA	c0 a0	Y
		DHE	RSA	c0 a2	Y
		PSK	—	c0 a8	Y
		DHE-PSK	—	c0 aa	Y

[a]Die gleichen Schlüsselaustauschalgorithmen gelten auch für AES-256-GCM-384.
[b]Die gleichen Schlüsselaustauschalgorithmen gelten auch für AES-128-CCM, AES-256-CCM(-8).

die Nutzdaten vertraulich mit Hilfe von symmetrischen kryptographischen Verfahren versendet werden.

Daher besteht das TLS Protokoll aus zwei Schichten, die auf einer *zuverlässigen* Transport-Schicht (bspw. Transmission Control Protocol (TCP)) aufsetzen: Direkt auf der Transport-Schicht setzt die Record-Schicht auf und beinhaltet neben der Versionsnummer, um welche Art von TLS-Nachricht gerade versendet wird. Auf dieser Schicht setzt eine von vier Schichten auf:

- Handshake – zum Austausch von Handshake-Nachrichten
- Change-Chipher-Spec – zum Mitteilen, dass sich die Cipher Suite geändert hat
- Alert – zum Melden von Fehlern, insbesondere während des Handshakes
- Application Data – zum Versenden von vertraulichen Nutzdaten

In Tabelle 4.1 sind die Cipher Suites für die Verschlüsselungsverfahren AES-CCM und AES-GCM dargestellt, da diese beiden Verfahren standardmäßig im Internet der Dinge aufgrund ihrer AEAD-Eigenschaften zum Einsatz kommen werden. Diese Cipher Suites werden üblicherweise mit folgender Syntax beschrieben:

$$\underbrace{\text{TLS}}_{\text{TLS Präfix}}\_\underbrace{\text{DHE\_PSK}}_{\text{Schlüsselaustausch}}\_\text{WITH}\_\underbrace{\text{AES\_128\_GCM\_SHA256}}_{\text{Verschlüsselung}}$$

Funktionsweise des Handshakes

Der erstmalige Handshake wird immer von einem TLS Client initiiert. Dieser sendet eine **Client Hello** Nachricht an den TLS Server. In dieser Nachricht gibt der Client folgendes

bekannt:

- die höchste TLS Version, die er unterstützt,
- eine zufällige Zeichenfolge, auch Random genannt,
- eine Session ID, wenn eine bereits aufgebaute Verbindung wieder aufgenommen werden soll,
- eine Liste der unterstützten Cipher Suites,
- eine Liste unterstützter Kompressionsmethoden und
- eine optionale Liste von Extensions, die der Client unterstützt.

Der Server antwortet auf diese mit einer **Server Hello** Nachricht, in der er insbesondere die Cipher Suite und die Kompressionsmethode festlegt. Danach folgen weitere von der Cipher Suite abhängige oder optionale Informationen vom Server, wie ein Zertifikate, das Bekanntgeben des Server-Schlüssels oder eine Zertifikatsanforderung. Abgeschlossen werden diese Informationen mit einer **Server Hello Done** Nachricht.

Nun hat der Client auf diese Informationen zu antworten und gibt seine Informationen bekannt. Zu diesen zählt ein optionales Zertifikat, das Bekanntgeben des Client-Schlüssels (**Client Key Exchange**) und eine optionale Zertifikatsüberprüfung. Abgeschlossen wird dies mit einer zusammengesetzten Nachricht, bestehend aus einer **Change Chiper Spec** Nachricht gefolgt von einer mit diesen neuen Sicherheitsparametern verschlüsselten **Finished** Nachricht.

Sollte der Server diese letzte Nachricht korrekt entschlüsseln können, so antwortet er ebenfalls mit solch einer zusammengesetzten Nachricht, die der Client entschlüsseln muss, um zu prüfen, ob die Sicherheitsparameter korrekt ausgehandelt wurden.

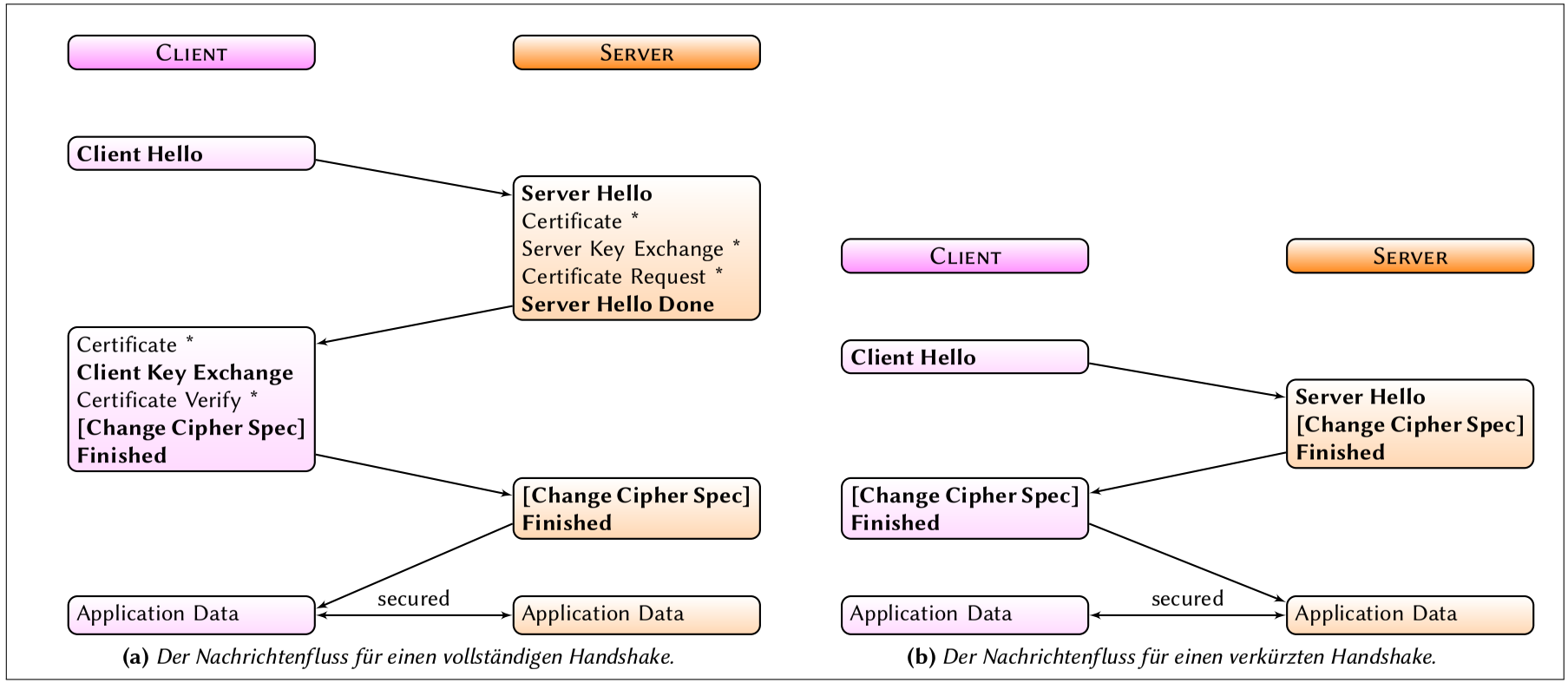
In Abbildung 4.5 (auf Seite 45) ist dieser Handshake abgebildet. Wird bei der Client Hello Nachricht bereits eine Session ID mitgesendet und der Server unterstützt dies, so kommt der verkürzte Handshake zum Einsatz.

In dieser Abbildung ist nicht gezeigt, dass der TLS Server auch einen neuen Handshake mittels einer **Hello Request** Nachricht einleiten kann. Dies ist erforderlich, wenn der symmetrische Schlüssel an Sicherheit verliert, weil beispielsweise der TLS Server angegriffen wird oder der Schlüssel zu oft eingesetzt wurde.

4.2.5. Datagram Transport Layer Security (DTLS)

Datagram Transport Layer Security (DTLS) – definiert in RFC 6347 [27] – ist ein Verschlüsselungsprotokoll und basiert auf TLS. Das Problem bei TLS ist, dass es nur mit zuverlässigen Transportprotokollen arbeitet. Einige Anwendungen machen es aber erforderlich, einen nicht-zuverlässigen Transport zu verwenden, da es beim zuverlässigen Transport zu Verzögerungen kommen kann. Damit auch solche auf nicht-zuverlässigen Transport beruhende Anwendungen vertrauliche Nachrichten verwenden können, wurde DTLS entwickelt.

DTLS möchte so gut wie möglich zu TLS kompatibel sein, so dass nur diejenigen Änderungen umgesetzt werden, die zum Betrieb mit einem nicht-zuverlässigen Transport (bspw. UDP) benötigt werden. Die DTLS Version 1.2 ist demnach kompatibel mit TLS



(a) *Der Nachrichtenfluss für einen vollständigen Handshake.*

(b) *Der Nachrichtenfluss für einen verkürzten Handshake.*

Abbildung 4.5. *Das TLS/DTLS Handshake Protocol [25]. Es wird vom Client ausgehend eine Verbindung mit einem Server aufgebaut. Die mit * markierten Nachrichten werden optional oder nur unter bestimmten Umständen versendet. Bei einer [Change Cipher Spec] werden die ausgehandelten Sicherheitsparameter aktiviert. Der DTLS Handshake verwendet üblicherweise einen (optionalen) Cookie Exchange, bestehend aus einem Hello Verify Request und einer zweiten Client Hello Nachricht. Dieser findet zwischen der (ersten) Client Hello und der Server Hello Nachricht statt.*

Version 1.2.[3]

Allgemeine Funktionsweis

In DTLS kann jede Cipher Suite verwendet werden, die in [26] mit „DTLS-OK“ gekennzeichnet ist. In Tabelle 4.1 wurde dies bereits ergänzt. Somit können AES-CCM und AES-GCM auch in DTLS mit den angegebenen Schlüsselaustauschverfahren verwendet werden.

Um den Nachteil eines unzuverlässigen Transports zu kompensieren, wird in der Record-Schicht eine Epoche sowie eine Sequenznummer eingesetzt. Die Epoche wird hierbei nur um eins erhöht, wenn eine Change Cipher Spec Nachricht versendet wurde, ansonsten bleibt sie gleich. Die Sequenznummer dagegen wird mit jeder neuen Nachricht erhöht und lässt den Empfänger erkennen, ob er die Nachrichten in der richtigen Reihenfolge empfangen hat.

Auch in der Handshake-Schicht kommen neue Felder vor: Eine Message Sequenz Number kennzeichnet die aktuelle Handshake Nachricht. Zudem können zu große Handshake Nachrichten mit Hilfe zwei weiterer Felder in Fragmente zerlegt werden, wobei das erste den Fragment Offset angibt und das zweite die Fragment Länge.

Funktionsweise des Handshakes

Der TLS Handshake wurde zur Verwendung mit DTLS um einen optionalen, aber empfohlenen Cookie-Austausch ergänzt, um eventuellen Denial-of-Service Angriffen (wie Resource Consumption Attacken) vorzubeugen.

Dabei wird die erste **Client Hello** Nachricht – wie von TLS gewohnt – vom DTLS Client an den DTLS Server gesendet. Sie enthält noch keinen Cookie. Der Server antwortet nun mit einer **Hello Verify Request** Nachricht, die einen Cookie enthält. Der Client muss nun seine Client Hello Nachricht wiederholt senden, allerdings diesmal mit dem vom Server empfangenen Cookie. Diesen überprüft der Server und fährt bei Korrektheit mit dem weiteren Handshake wie in TLS fort (siehe Abbildung 4.5).

4.3. Leichtgewichtige Sicherheitslösungen

Eine interessante und aktuelle Dissertation stammt von Shahid Raza am SICS Swedish ICT[4], um leichtgewichtige Sicherheitslösungen für das Internet der Dinge [37] zu diskutieren.

Bereits zuvor hatte er in Zusammenarbeit mit Kollegen Vorschläge gemacht, um anerkannte Sicherheitsstandards wie IPsec und DTLS in leichtgewichtige Varianten zu

[3] Allerdings ist die DTLS Version 1.0 nicht kompatibel mit der gleichnamigen TLS Version, sondern mit der TLS Version 1.1. Eine DTLS Version 1.1 gibt es nicht.

[4] Allgemein ist das schwedische Institute für Informatik (engl. *Swedisch Institute of Computer Science, SICS*) „a leading research institute for applied information and communication technology in Sweden“, auch bei der Entwicklung des Internets der Dinge. Siehe `https://www.sics.se/`

überführen, indem eine 6LoWPAN-HC für diese Protokolle definiert wurde. Mit Constrained Application Protocol (CoAP) ist ein zu HTTP kompatibles, aber energiesparendes Webprotokoll in Entwicklung, wozu es auch bereits erste Vorschläge zur Integration von Sicherheitsdiensten gibt. [37]–[41]

Daneben gibt es auch andere Forschungsarbeiten, die Kommunikations- und Sicherheitslösungen zum Umfeld Internet der Dinge suchen [42].

4.3.1. Anforderungen an Kommunikationsprotokolle

Kommunikationsprotokolle, die im Internet der Dinge eingesetzt werden, müssen – wie auch S. Raza in [37] erklärt – Sicherheitsmaßnahmen bereithalten, die …

1. leichtgewichtig und damit energiesparend sind,
2. speziell für den Einsatz in Umgebungen mit eingeschränkten Ressourcen entwickelt worden sind,
3. den Anforderungen des Internet der Dinge gerecht werden (z. B. die Unterstützung von IPv6),
4. eine sichere Kommunikation zwischen dem *Internet der Dinge* und dem *Internet der Computer* erlauben.

Mit 6LoWPAN ist ein energiesparendes Kommunikationsprotokoll für das Internet der Dinge geschaffen worden. Es stellt eine IPv6 Kommunikation bereit, nicht nur zwischen Netzknoten eines WPANs, sondern auch zwischen Netzknoten verschiedener Netze. Somit verbindet es das Internet der Dinge mit dem Internet der Computer.

Doch die Etablierung von Sicherheitsdienste ist noch ein viel umstrittenes Problem, insbesondere wenn man eine Kommunikation zwischen den beiden Internettypen sichern möchte. Da 6LoWPAN den Einsatz des IEEE 802.15.4 Standards vorsieht und dieser Standard erste Sicherheitsdienste unterstützt, war es in der Vergangenheit naheliegend, diese zu nutzen. Doch sind diese Sicherheitsdienste zur Sicherung der Kommunikation im Internet der Dinge nicht mehr hinreichend, so dass neue Sicherheitsmaßnahmen gefunden werden müssen, die die genannten Sicherheitsanforderungen umfangreich erfüllen.

4.3.2. Verschlüsselung — Ende-zu-Ende vs. Punkt-zu-Punkt

Die Vertraulichkeit sowie die Integrität der Daten kann sowohl mit Hilfe einer Ende-zu-Ende Verschlüsselung (engl. *end-to-end encryption*) oder mit Hilfe einer Punkt-zu-Punkt Verschlüsselung (engl. *hop-by-hop encryption*) sichergestellt werden.

Eine Punkt-zu-Punkt Verschlüsselung stellt eine Sicherheitsbeziehung zwischen physikalisch benachbarten Netzknoten her – also Netzknoten, die direkt miteinander kommunizieren können. Optimalerweise besitzt jede Sicherheitsbeziehung ein anderes Geheimnis, so dass das Kompromittieren von Netzknoten nur sehr geringe Auswirkungen hat. Das

bedeutet aber zugleich, dass die Wahrung der Geheimnisse sehr aufwendig ist, wodurch das Schlüsselmanagement erschwert wird. Zudem können keine Gruppennachrichten ausgetauscht werden, wenn jeder eine andere Sicherheitsbeziehung hat. Daher müsste es neben den paarweisen Sicherheitsbeziehungen auch noch gruppenweise bzw. netzweite Sicherheitsbeziehungen geben.

Da die Punkt-zu-Punkt Verschlüsselung im IEEE 802.15.4 Standard eine der ersten standardisierten, die Interoperabilität fördernden Verschlüsselungstechniken für drahtlose Sensornetze war, wurde sie zur Sicherung der Kommunikation für das HexaBus Home Automation Systen ausgewählt und implementiert. Diese Punkt-zu-Punkt Verschlüsselung wird auf MAC-Schicht durchgeführt und bewirkt, dass jeder Netzknoten – vor allem jeder Routingknoten auf dem Weg zum Empfänger – die komplette Nachricht entschlüsselt und wieder verschlüsselt. Eine auf MAC-Schicht verschlüsselte Nachricht verschleiert die IP-Adresse des Empfängers. Daher muss das Paket entschlüsselt werden, um an diese Adresse herauszukommen. Somit muss jedem Routingknoten auf dem Weg zum Empfänger vertraut werden. Nur eine zusätzliche Verschlüsselung auf höheren Schichten schützt das Paket weiter vor Einsichtnahme.

Dennoch wird diese Form der Verschlüsselung gerne in Sensornetzen eingesetzt. Es bietet eine einfache Form der Verschlüsselung und erlaubt Aggregationsknoten (Filterknoten) das Zusammenfassen (Filtern) von Nachrichten. Ziel ist es, frühzeitig das Nachrichtenaufkommen im Netz zu reduzieren. In der Hausautomatisierung können beispielsweise gleichwertige Sensorinformationen wie Temperaturen eines Raumes zusammengefasst werden. Eine andere Gelegenheit bietet es sich, wenn nicht-authentische bzw. redundante Pakete im Netzwerk im Umlauf sind, so dass das Filtern dieser Pakete bei der Bekämpfung von Eindringlingen hilft.

Die Sicherung einer Ende-zu-Ende Kommunikation ist in klassischen Netzen beliebter, da der gesicherte Nachrichteninhalt nur vom Empfänger eingesehen und von niemand anderem manipuliert werden kann. Doch auf diesem Gebiet haben sich noch keine Sicherheitsprotokolle im Internet der Dinge etabliert, die die in Abschnitt 5.2 genannten Sicherheitsanforderungen in Kommunikationsprotokollen umfangreich erfüllen.

Eine Ende-zu-Ende Verschlüsselung stellt eine Sicherheitsbeziehungen zwischen Sender und Empfänger einer Nachricht her – also Netzknoten, die benachbart sein können, aber nicht benachbart sein müssen. Der Vorteil dieser Sicherheitsbeziehung ist auch zugleich ein Nachteil. Denn der Empfänger der Nachricht ist der einzige Netzknoten, der in der Lage ist, die Nachricht zu entschlüsseln, aber auch der einzige, der die Nachricht entschlüsseln muss. Alle Routingknoten können die Nachricht weiterleiten, ohne in den vertraulichen Nachrichtenteil einsehen zu müssen.

Da der Empfänger lediglich den vertraulichen Teil einsehen kann, ist der Einsatz von Netzknoten, die eine Filteraufgabe oder eine Aggregationsfunktion besitzen, nicht sinnvoll. Sie müssen explizit in die Kommunikation der Sensor- und Regelwerkknoten einbezogen werden, um tätig werden zu können.

Dadurch bedingt, dass Routingknoten Nachrichten nur weiterleiten, aber nicht entschlüsseln müssen, müssen sie auch keine Verschlüsselungstechnik hierfür mitbringen. Erst, wenn die Routingknoten selbst Nachrichten mittels einer Ende-zu-Ende Verschlüs-

selung versenden/empfangen wollen, benötigen sie die gleichen Techniken wie der Empfänger/Sender.

Auch im Internet der Dinge bzw. im Hausautomatisierungssystem sollte bevorzugt eine *Ende-zu-Ende Verschlüsselung* Anwendung finden, um Daten der Hausautomatisierung optimal während der Weiterleitung durch Router zu schützen. Es entsteht zwar der Nachteil, dass erst der Empfänger die Nachricht filtern kann, allerdings könnte dieses Problem durch ein intelligentes Organisieren des Nachrichtenverkehrs auf Anwendungsebene gelöst werden, indem diese so genannten Filter- bzw. Aggregationsknoten direkt in den Nachrichtenverkehr einbezogen werden.

4.3.3. Schlüsselverteilung und Schlüsselmanagement

Verfahren der symmetrischen Kryptographie sind effizienter und damit energiesparsamer als vergleichbare Verfahren der asymmetrischen Kryptographie [vgl. 11, 12, S. 31 f.]. Daher wird die symmetrische Kryptographie sowohl in klassischen Netzen wie auch in drahtlosen Sensornetzen eingesetzt.

Allerdings bereiten Verfahren der asymmetrischen Kryptographie nicht das Problem, dass Geheimnisse (Schlüssel) auf irgendeine Weise ausgetauscht (geteilt) werden müssen. Das Teilen dieser Geheimnisse darf nicht belauscht werden und birkt daher im Allgemeinen Schwierigkeiten. In klassischen Netzen kommen asymmetrischer Verschlüsselungsverfahren zum Einsatz, damit diese Geheimnisse ausgetauscht werden können.

Doch möchte man sich den Einsatz von asymmetrischen Verfahren in drahtlosen Sensornetzen gerne ersparen, da sie Bandbreite bei der Übertragung, Performance bei der Berechnung, aber vor allem zusätzlichen Speicher benötigen. Dies steigert die Produktionskosten sowie den Energieverbrauch. Aus diesen Gründen ist das Teilen von Geheimnissen in drahtlosen Sensornetzen und damit im Internet der Dinge eine besondere Herausforderung.

Bisherige Kommunikationsprotokolle, die Sicherheitsdienste anbieten, verlagern das Problem des *Schlüsselmanagements* gerne auf andere bzw. zukünftige Protokolle. Ein Beispiel ist der IEEE 802.15.4 Standard, der Folgendes formuliert:

> The cryptographic mechanism in this standard is based on symmetric-key cryptography and uses keys that are provided by higher layer processes. The establishment and maintenance of these keys are outside the scope of this standard. The mechanism assumes a secure implementation of cryptographic operations and secure and authentic storage of keying material. [28, S. 15]

Auch das Routing Protokoll RPL unterstützt Sicherheitsdienste, aber verlagert das Schlüsselmanagement ebenfalls [21, S. 126]. Daher ist das Kernproblem des Schlüsselmanagements ein noch weitgehend ungelöstes Problem in Anwendungen zum Internet der Dinge.

Folgendes Beispiel soll das Problem der Speicherbedarfs bei der Schlüsselverwaltung demonstrieren: Angenommen, jeder Netzteilnehmer in einem drahtlosen Sensornetz will

Tabelle 4.2. *Speicherkapazitäten ausgewählter Ultra-Low-Power Mikrocontroller.*

Mikrocontroller	CPU Frequenz	Flash/FRAM	SRAM	EEPROM[1]
AVR ATmega128rfa1 (8 bit)	16 MHz	128 KiB	16 KiB	4 KiB
AVR ATmega64rfa2 (8 bit)	16 MHz	64 KiB	8 KiB	2 KiB
AVR ATmega128rfa2 (8 bit)	16 MHz	128 KiB	16 KiB	4 KiB
AVR ATmega128rfa2 (8 bit)	16 MHz	256 KiB	32 KiB	8 KiB
MSP430 F1611 (16 bit)	8 MHz	48 KiB	10 KiB	
MSP430 F2618 (16 bit)	16 MHz	116 KiB	8 KiB	
MSP430 F5253 (16 bit)	25 MHz	128 KiB	16 KiB	
MSP430 F5255 (16 bit)	25 MHz	128 KiB	32 KiB	
MSP430 FR5969[2] (16 bit)	25 MHz	64 KiB	2 KiB	

[a]Dieser Speicher hat üblicherweise nur eine begrenzte Anzahl Schreibzyklen (z. B. 100 000 bei 25 °C).
[b]Dieser Mikrocontroller besitzen kein Flash, sondern ein FRAM [58].

mit jedem anderen vertraulich kommunizieren. Dann liegt ein Mesh-Netzwerk vor — ähnlich dem aus Abbildung 3.1a (auf Seite 25). Das bedeutet, dass in einem Netz mit n Teilnehmern maximal $\frac{n \cdot (n-1)}{2}$ vertrauliche Verbindungen benötigt werden (triviale Lösung). Ein Netzteilnehmer muss daher maximal $(n-1)$ unterschiedliche Geheimnisse bewahren, für jede vertrauliche Verbindung eines. Wenn folglich ein Verschlüsselungsverfahren mit einer Schlüssellänge von 3072 bit (z. B. RSA) zum Einsatz kommt, dann speichert jeder Netzknoten in einem Netz mit 50 weiteren Netzknoten maximal $18{,}75$ KiB:

$$(51-1) \cdot 3072 \text{ bit} = 150 \cdot 2^{10} \text{ bit} = 19\,200 \text{ B} = 18{,}75 \text{ KiB}.$$

Diese Größe mag klein erscheinen, insbesondere wenn nicht alle vertraulichen Verbindungen benötigt werden. Doch die Frage, die sich stellt, ist: In welchem Speichermedium werden diese Schlüssel aufbewahrt?

Werden diese Schlüssel im Flash eines Mikrocontrollers gespeichert, so besitzt dieser i. d. R. zwar genügend davon. Doch können diese Schlüssel nicht zur Laufzeit geändert werden und müssen zudem beim Bespielen des Flashs erzeugt worden sein. Werden diese Schlüssel zur Laufzeit generiert, dann müssen sie auf einem beschreibbaren Speichermedium Platz finden.

Ein AVR Mikrocontroller besitzt beispielsweise einen SRAM (flüchtiger Speicher) und ein EEPROM (nicht-flüchtiger Speicher). Sie können dann zwar jederzeit ersetzt werden, doch die Speicherkapazität dieser Speichermedien ist (wesentlich) niedriger. Texas Instruments verwendet für eine seiner MSP430-Reihen FRAM anstelle von Flash, das wesentlich energieeffizienter ist, schnellere Schreibzyklen erlaubt und eine länge Lebensdauer verspricht [58]. In Tabelle 4.2 sind verschiedene Ultra-Low-Power Mikrocontroller und die Größe ihrer Speichermedien dargestellt. Der MS430 FR5969 besitzt solch einen FRAM. Allerdings ist er nur 64 KiB groß, um Code und Daten unterzubringen.

Da die triviale Lösung demnach nicht günstig ist, gibt es eine Reihe von Verfahren,

die den Speicherbedarf, den Berechnungsaufwand der Geheimnisse sowie die Widerstandsfähigkeit gegen Kompromittierungen optimieren wollen. Dennoch bereitet das Schlüsselmanagement allgemein Probleme: Werden alle möglichen vertraulichen Verbindungen vor dem Ausbringen der Netzknoten erzeugt, geht viel Speicher verloren, der andersweitig genutzt oder eingespart werden könnte. Wenn jedoch erst nach Ausbringung eine vertrauliche Verbindung aufgebaut werden muss, so kommen recht komplizierte Schlüsselverteilungsalgorithmen zum Einsatz.

Da zurzeit die asymmetrischen, kryptographischen Verfahren im Gegensatz zu den symmetrischen Alternativen wesentlich mehr Energie (Zeit zur Berechnung, Übertragung und Prüfung) benötigen, kommen symmetrische, kryptographische Verfahren zum Einsatz. Diese machen es erforderlich, dass ein Geheimnis ausgetauscht wird. Daher muss verhindert werden, dass ein Angreifer den Austausch dieses Geheimnisses belauscht.

Oft werden hierzu verschiedene, bereits vorhandene, vertrauliche Verbindungen genutzt. Doch muss allen Kommunikationspartnern auf diesem Wege getraut werden. Bei unentdeckten Kompromittierungen ist dies nicht mehr garantiert, so dass das Geheimnis in Teile zerlegt werden kann und über verschiedene, disjunkte Pfade zur Gegenstelle versendet wird. Dadurch wird dem Angreifer das Zusammenfügen aller Teile erschwert. [43]

Die alternative Lösung ist die Verwendung von Schlüsselmanagementverfahren, die auf Zufallsberechnungen beruhen. [44] Sie sind allerdings nur ab einer bestimmten Netzteilnehmeranzahl geeignet, da es vorkommen kann, dass manche Netzknoten keine vertrauliche Verbindung zu anderen aufbauen können.

Zusammengefasst stellt folglich die Verteilung von Schlüsseln und damit das Schlüsselmanagement eine besondere Herausforderung dar, da asymmetrische Kryptographie in drahtlosen Sensornetzen aufgrund der begrenzten Resourcen nur schwer umsetzbar ist. Aus diesem Grund wird bei der Konzipierung der Sicherheitsarchitektur im nächsten Kapitel gesondert auf diese Problematik bezugnehmend auf das Hausautomatisierungssystem eingegangen.

5 Konzept der Sicherheitsarchitektur

Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi. (The system must not require secrecy and can be stolen by the enemy without causing trouble.)

—Auguste Kerkhoff (1835 – 1903)

Nach der Vorstellung der Voraussetzungen und Anforderungen an die Hausautomatisierung sowie der Analyse von Hausautomatisierungssystemen wird in diesem Kapitel die Sicherheitsarchitektur für das im Abschnitt 3.6 ausgewählte System konzipiert.

In den weiteren Betrachtungen wird angenommen, dass genug Routingknoten zum Einsatz kommen, damit alle Netzknoten mit dem Regelwerkknoten kommunizieren können. Weiterhin wird angenommen, dass die Verteilung der Netzknoten (Position im Raum) nicht bekannt ist.

Weiterhin sei angemerkt, dass im Anhang A ein Beispielszenario zu finden ist, das eine Hausautomatisierungsanlage mit zwei Steuerungen beschreibt. Bei diesem Beispielszenario handelt es sich um den *heimischen Arbeitsraum*, da es in der heutigen Zeit immer häufiger vorkommt, dass Menschen von Zuhause aus arbeiten. Es ist nützlich ein Beispielszenario im Hinterkopf zu haben, wenn dieses Kapitel gelesen wird. Allerdings ist es nicht erforderlich.

Zur Konzipierung der Sicherheitsarchitektur wird dieses Kapitel wie folgt aufgebaut: Zuerst werden Gefahren genannt, von denen eine Bedrohung ausgeht. Danach werden bezugnehmend auf diese Gefahren Sicherheitsanforderungen aufgestellt. Im nächsten Schritt werden Sicherheitsmaßnahmen verwandter Projekte vorgestellt, die im Zusammenhang mit der Sicherheitsarchitektur interessant sind.

Nun folgt die Vorstellung des Konzepts, das anhand der Sicherheitsanforderungen, aber auch aufgrund von Erkenntnissen aus verwandten Projekten erstellt worden ist. Dieses gliedert sich wiederum in zwei Teile: Im ersten werden Sicherheitsmechanismen besprochen. Dort werden Fragestellungen wie „Auf welcher Schicht im Netzwerk-Stack soll eine Nachricht verschlüsselt werden?“ Lediglich Fragestellungen nach dem Schlüsselmanagement werden nicht beantwortet. Sie sind Thema des zweiten Teils, in welchem es insbesondere um die Zugriffssteuerung geht.

Im nächsten Kapitel wird dann die Sicherheitsarchitektur evaluiert. Da die erforderliche Analyse des Hausautomatisierungssystems sowie die ausführliche Konzepterstellung

der Sicherheitsarchitektur Zeit erfordert hat, beschränkt sich die Evaluation auf drei Eigenschaften: Energiebedarf, Kommunikationsaufkommen sowie RAM-Verbrauch.

5.1. Gefahrenanalyse

Der erste Schritt zur Konzipierung einer Sicherheitsarchitektur ist das Aufstellen einer Gefahrenanalyse. In dieser geht es darum, das Hausautomatisierungssystem auf mögliche Gefahren hin zu untersuchen, um festzustellen, welche Sicherheitsanforderungen definiert werden müssen. Je schlimmer eine Gefahr ist, desto strenger sollten diese Anforderungen ausfallen. Die folgenden Gefahren werden hierbei in **drei Stufen** eingeteilt: (1) niedrig, (2) mäßig und (3) hoch.

5.1.1. Verletzung des allgemeinen Persönlichkeitsrechts

Die erste Gefahr, die unmittelbar aus den Anforderungen an die Hausautomatisierung resultiert, ist bereits in der Einführung sowie im Unterabschnitt 2.2.1 angesprochen worden. Zur Akzeptanz des Hausautomatisierungssystems ist es erforderlich, die allgemeinen Persönlichkeitsrechte der Hausbewohner (sowie ihrer Gäste) zu schützen. Demnach wird diese Gefahr zuerst behandelt.

Um Einbrüche zu erkennen, kommt eine Einbruchskontrollsteuerung zum Einsatz. Oft werden Kameras zur Überwachung eingesetzt. Diese haben zum Ziel, die Identifikation nach der Tat zu ermöglichen. Alternative Lösungen erfordern nicht das Identifizeren der Einbrecher, sondern lediglich die Erkennung des Einbruchs. So kann die Polizei schnell benachrichtigt werden. In der Einbruchskontrollsteuerung aus dem Beispielszenario (siehe Anhang A) wurde hierzu der Einsatz von Bewegungsmeldern vorgesehen.

Neben der Nutzung der Bewegungsinformationen zur Einbruchsdetektion wird der Bewegungsmelder auch zur Klimaanlagensteuerung benötigt. Allerdings ist es möglich, dass allein der Einsatz von Bewegungsmeldern ebenfalls eine Verletzung der Privatsphäre bewirkt, obwohl die Hausbewohner nicht gefilmt werden. Denn der Einsatz mehrerer Bewegungsmelder in jedem Raum des Hauses lässt beispielsweise schlussfolgern, welche Tätigkeit ein Hausbewohner ausführt, wodurch seine Bewegungsfreiheit eingeschränkt wäre.

Das Problem aller Überwachungslösungen wird sein, dass die Hausbewohner darauf *vertrauen* müssen, dass die aus den Überwachungslösungen gewonnenen Informationen weder eingesehen noch weitergegeben oder gar veröffentlicht werden. Erst dann werden die Hausbewohner nicht in ihren allgemeinen Persönlichkeitsrechten eingeschränkt.

Ein Beispiel soll die Auswirkung des Missbrauchs verdeutlichen: Die Weitergabe oder Veröffentlichung der Information, dass der Hausbewohner während seiner Krankschreibung nicht zu Hause war – egal, ob dies rechtmäßig oder unrechtmäßig – erweckt bekanntermaßen das Interesse des Arbeitgebers und hat eventuell eine (unberechtigte) Kündigung des Arbeitsverhältnisses zur Folge. Daher wird die Gefahr, die von solchen personifizierten Daten ausgeht, **hoch** eingestuft, da von einem Interesse des Angreifers an diesen Daten auszugehen ist.

Als Hardware-Entwickler von Bewegungsmeldern für das Hausautomatisierungssystem ist es aber schwer einzuschätzen, ob anhand seiner aufgenommenen Bewegungsdaten Personen identifiziert werden können. Hierzu müsste er den genauen Verwendungszweck samt der Hausautomatisierungsregeln kennen. Da dies normalerweise nicht der Fall ist, obliegt dem Hausbewohner diese Verantwortung.

Der Entwickler kann allerdings die Bewegungsdaten auf „angriffssichere“ Weise zum Regelwerkknoten versenden und dann sofort löschen. So stellt zumindest das (längerfristige) Speichern dieser Informationen im Sensorknoten keine Gefahr dar. Der Regelwerknoten muss dann diese Informationen auf sichere Weise verarbeiten und die daraus gewonnenen Informationen angriffssicher an die Aktorknoten weiterleiten.

5.1.2. Neuartige Koordination von Einbrüchen

Die nächste Gefahr, die aus der vorherigen hervorgeht, handelt von der geschickten Koordination von Einbrüchen.

Einbrüche in Wohnhäuser oder Wohnungen passieren, seitdem sie gebaut werden. Oft war es jedoch erfordlich, dass der Einbrecher auf gut Glück den Einbruch begangen hat oder er musste das Haus vorher (persönlich) beobachten.

Verfolgt man das Beispiel aus dem vorherigen Unterabschnitt weiter, können Einbrecher Bewegungsdaten auf intelligente Weise missbrauchen. Denn sie können damit ihre Einbrüche in einer Weise koordinieren, die es bisher nicht gibt: Sie können durch das Hinzufügen eigener (oder nach Kompromittierung vorhandener) Netzknoten eine Fernüberwachung einrichten, um so das Wohnhaus bequem aus der Ferne zu beobachten. Durch die Auswertung der Bewegungsdaten weiß der Einbrecher i. d. R. sofort, ob sich jemand im Haus befindet oder nicht. Es können auch theoretisch andere Sensorinformationen dazu genutzt werden (z. B. Status der Leuchten). Er kann weiterhin Statistiken zum Ein- und Ausgehverhalten der Hausbewohner aufstellen.

In die Zukunft vorausplanend, wenn Hausautomatisierungssysteme in jedem Haushalt vorhanden sind, kann es so passieren, dass dem Einbrecher sogar eine automatisierte und großräumige Überwachung mehrerer Häuser simultan ermöglicht wird, wodurch viele Einbrüche gezielt koordiniert werden können. Dadurch verringert sich die Wahrscheinlichkeit, den Einbruch zu bemerken und den Täter festzunehmen.

Zur Zeit geht von dieser Gefahr eine **niedrige** Bedrohung aus. Doch dies kann sich bald ändern.

5.1.3. Lauschangriff (eavesdropping attack)

Der Lauschangriff ist eine Gefahr, die nicht unterschätzt werden darf. Die im System versendeten Bewegungsdaten kann ein Einbrecher beispielsweise auf diese Weise in Erfahrung bringen.

Mittels dieser Attacke kann ein Angreifer aber auch Informationen über die Funktionsweise des Hausautomatisierungssystems sammeln, indem er die Kommunikation allgemein belauscht. Das Einsehen der Kommunikation verrät einem Angreifer, welche Sensoren und Aktoren im System benutzt werden und wie diese Aktoren angesteuert

werden. Unter Umständen ist der Angreifer sogar in der Lage, die eingespeicherten Regeln des Regelwerkknotens zu erkennen.

Ein weiteres Interesse eines Angreifers ist das Belauschen von ausgetauschten Geheimnissen wie Schlüsseln, die zur Ver- und Entschlüsselung oder zur Authentifikation von Nachrichten genutzt werden. Mit diesen Informationen ist es möglich, die folgende angriffssichere Kommunikation zu belauschen oder zu manipulieren, wodurch die Sicherheit untergraben wurde.

Wenn der Angreifer den Netzverkehr lediglich als Dritter verfolgt, dann handelt es sich um einen passiven Lauschangriff. Bei diesem wird ein Netzknoten in Reichweite des Sensornetzes platziert und der gesamte Netzverkehr aufgezeichnet und anschließend ausgewertet. Wenn diese Methode durch Sicherheitsmaßnahmen blockiert wird, kann er noch einen Man-in-the-Middle Angriff und damit einen aktiven Lauschangriff ausführen, indem er sich als legales Mitglied im Netzwerk ausgibt und sich zwischen die Kommunikation zweier Netzknoten schaltet.

Wie bereits im Unterabschnitt 3.4.4 erwähnt worden ist, bietet die zentralisierte Netzstruktur der dedizierten Regelverarbeitung interessante Angriffsziele in der Nähe des Regelwerkknotens. Dort ist die Wahrscheinlichkeit am größten, dass der Angreifer alle nötigen Informationen erhält.

Weiterhin ermöglicht ein erfolgreicher Lauschangriff das Ausführen weiterer Angriffe. Daher wird das Interesse eines Angreifers an solch einer Attacke und damit die Bedrohung, die von dieser Gefahr ausgeht, (sehr) **hoch** eingestuft.

5.1.4. Gefälschte Ansteuerung von Aktoren

Bei einem erfolgreichen (systemweiten) Lauschangriff kennt der Angreifer die im System vorhandenen Aktoren. Verwendet beispielsweise ein Haushalt eine Einbruchskontrollsteuerung (wie aus Abschnitt A.2), dann versucht ein Einbrecher u. U. zu umgehen. Dies schafft er, indem er die Nachrichten an die Alarmvorrichtung manipuliert oder unterdrückt.

Die Bedrohung, die von dieser Gefahr ausgeht, ist abhängig von den Auswirkungen, die eine gefälschte Ansteuerung an die im System vorhandenen Aktoren bewirkt. Dabei ist für jeden Aktor die Bedrohung gesondert zu bewerten. Das Ansteuern des Heizungsreglers verursacht verhältnismäßig geringen Schaden und wäre eher mit niedrig zu bewerten. Das Rufen der Polizei oder Feuerwehr, obwohl keine Notwendigkeit besteht, wird eine wesentlich höhere finanzielle Schädigung nach sich ziehen und ist damit mit einem hohen Bedrohungspotential versehen.

Es mag sein, dass solche Angriffe nur sehr selten auftreten. Doch wenn sie auftreten, können sie Schaden anrichten. Daher wirkt sich eine gefälschte Ansteuerung der Aktoren negativ auf das Ziel der finanziellen Einsparung aus. Da Schäden soweit gehen *können*, dass keine Einsparungen, sondern Unkosten entstehen, geht von dieser Gefahr eine (sehr) **hohe** Bedrohung aus.

5.1.5. Gefälschte Sensorinformationen

Allerdings ist es nicht nur möglich, Aktoren im Hausautomatisierungssystem zu manipulieren. Der Automationsprozess kann ebenfalls durch das Verändern der Sensorinformationen beeinflusst werden.

Ein einfaches, nicht entferntes Beispiel demonstriert den Einfluss der Änderung der Sensorinformation auf die Ansteuerung des Aktors: Ein Hersteller hat eine smarte Toilettenschüssel [59] entworfen, bei der die Spülung ferngesteuert aktiviert werden kann. Luxus-Varianten besitzen sogar eine fernsteuerbare Musikanlage. Aus einem vorherigen Lauschangriff hat der Angreifer erfahren, dass eine fernsteuerbare Toilettenschüssel im Haushalt vorhanden ist. Angenommen, das System erkennt automatisch, ob der Hausbewohner das Bad betritt (z. B. durch Bewegungsdaten). Ist dies der Fall wird automatisch die Musikanlage angeschalten. Die Erkenntnis, dass die Musikanlage beim Betreten des Bads aktiviert wird, hat er ebenfalls aus dem Lauschangriff erhalten. Nun ist der Angreifer bei einem ungesicherten System in der Lage, das Betreten des Bads vorzutäuschen und so die Musikanlage zu aktivieren. Auch könnte er wahrscheinlich die Lautstärke regulieren und ständig die Spülung betätigen, um die Nachbarn zu provozieren und damit dem Hausbewohner zu schaden.

Ähnliches kann auch bei der Einbruchskontrollsteuerung passieren, indem dem Regelwerkknoten mit Hilfe gefälschter Sensorinformationen (engl. *false data injection*) dem Bewegungsmelder suggestiert wird, dass kein Einbruch stattgefunden hat.

Daher ist es nicht nur wichtig, Befehle an die Aktoren vor Manipulation jeglicher Art zu schützen, sondern auch Sensorinformationen, die im Sensornetz übertragen werden. Aber auch hier gilt, dass die Gefahr abhängig von den Auswirkungen ist, die aus den gefälschten Sensorinformationen hervorgehen. Deswegen geht von dieser Gefahr die gleiche Bedrohung aus, wie bei den gefälschten Aktoransteuerungen, nämlich eine **hohe**.

5.1.6. Kompromittierung des Regelwerkknotens

Aufgrund der geforderten Skalierbarkeit des Hausautomatisierungssystems besitzt ein Angreifer neben der Manipulation vorhandener Nachrichten die Fähigkeit, eigene Netzknoten in das Sensornetz zu integrieren. Er kann daher von einem legitimen Netzknoten Nachrichten verschicken und auch Nachrichten empfangen.

Angenommen, die Kommunikation zwischen einem Sensor- / Aktorknoten und Regelwerkknoten ist *vertraulich*. Dann hilft dem Angreifer das Hinzufügen eigener Netzknoten zum Sensornetz nichts, wenn er diese vertrauliche Verbindung manipulieren möchte. Der Regelwerkknoten akzeptiert auch nur Nachrichten von registrierten Sensor- und Aktorknoten, für die er Regeln parat hat.

Da jeder Netzknoten auf Anwendungsschicht der Hausautomatisierungsanwendung ausschließlich mit dem Regelwerkknoten kommuniziert (siehe Abschnitt 3.6), bleibt dem Angreifer noch der Kompromittierungsversuch des Regelwerkknotens (z. B. per Fern-Zugriff auf die gespeicherten Schlüsselinformationen). Denn bei erfolgreicher Kompromittierung ist er in der Lage jede angriffssichere Komunikation auf Anwendungsschicht zu belauschen oder zu manipulieren.

Deswegen sind nicht nur um den Regelwerkknoten umliegende Routingknoten beliebte Angriffsziele (siehe Unterabschnitt 3.4.4), sondern der Regelwerkknoten gehört als zentrale Instanz auch zu diesen. Daher geht von der Bedrohung der Kompromittierung des Regelwerkknotens eine **hohe** Gefahr aus.

5.1.7. Klonen von Netzknoten

Angenommen, der Regelwerkknoten ist nicht kompromittierbar und das Hinzufügen von Netzknoten des Angreifers mit eigener Identität wirkt sich nicht negativ auf das Gesamtsystem aus, so besteht immer noch die Gefahr, dass vorhandene Netzknoten geklont werden können (engl. *node replication attack*). Beim Klonen wird die Identität eines bereits im Netz vorhandenen Netzknotens dupliziert (z. B. die IPv6-Adresse). Auch der Regelwerkknoten ist nicht aufgrund seiner Nicht-Kompromittierbarkeit vor dem Duplizieren seiner Identität geschützt.

Die Frage, die sich stellt, ist: Was erhofft sich der Angreifer vom Duplizieren der Identität, wenn er in der Lage ist, (a) nur nicht-vertrauliche Nachrichten oder (b) auch vertrauliche Nachrichten zu versenden und abzufangen?

Das Versenden und Abfangen vertraulicher Nachrichten stellt eine **hohe** Gefahr dar. Wenn es sich um einen Sensorknoten handelt, dann kann er den Automationsprozess mittels gefälschter Sensorinformationen manipulieren. Das Duplizieren der Identität von Aktoren kann zum Abfangen von Aktoransteuerungen genutzt werden. Wenn es sich beim duplizierten Netzknoten allerdings um den Regelwerkknoten handelt, dann kann der Angreifer sogar das gesamte Hausautomatisierungssystem übernehmen.

Angenommen, der Angreifer kann keine vertraulichen Nachrichten versenden und abfangen, dann kann er den Automationsprozess nicht manipulieren, sondern lediglich stören, indem er bestimmte Netzknoten überlastet. Diese Gefahr stellt, verglichen mit der vorherigen, eine **niedrige bis mäßige** Bedrohung dar.

5.1.8. Jamming Attacke

Die Funkfrequenz (engl. *radio frequency*) ist ein offenes Medium. Deshalb kann jeder diesen Kanal nutzen, um zu kommunizieren. Wenn eine Überlagerung der Kommunikation stattfindet, versteht der eine den anderen nicht. Daher wird ein Medium Access Control (MAC) Protokoll eingesetzt, das die Kommunikation in einem bestimmten Frequenzbereich organisiert und ein durcheinanderreden verhindert. Dies gilt allerdings nur solange, wie sich alle Teilnehmer an diesen Standard halten.

Diese Schwachstelle wird nun vom Angreifer ausgenutzt, indem eine Kommunikation durch absichtliche Störsendung (engl. *jamming*) unterbrochen wird. Die effektivste Methode des Jammings ist das *Radio Jamming*. Bei dieser Methode kommen so genannte Störsender zum Einsatz, die das Hintergrundrauschen in einem Frequenzbereich deutlich verschlimmern und so bereits auf Bitübertragungsschicht (Schicht 1 des OSI-Referenzmodels) eine Kommunikation verhindert.

Das Radio Jamming ist nur schwer zu bekämpfen, insbesondere weil es im MAC Protokoll definiert sein muss. Eine lahmgelegte Kommunikation kann zwar nicht ver-

hindert werden, allerdings kann das mehrmalige Nichtantworten auf eine Anfrage als Angriff aufgefasst werden und entsprechend ein alarmierender Systemzustand eingeleitet werden.

Das Jamming bewirkt lediglich eine Störung, aber nicht eine Manipulation des Automationsprozesses. Deswegen wird die Bedrohung, die von dieser Gefahr ausgeht, **niedrig** eingestuft.

5.1.9. Verschiedene Angriffe auf das Routing

Neben den bereits genannten Gefahren existieren eine ganze Reihe weiterer, die allerdings die Manipulation des Routings zum Ziel haben [vgl. 45, 46, für RPL-basierte Netze 47]. Beim Klonen von Netzknoten beispielsweise helfen diese Angriffe, um diejenigen Nachrichten abzufangen, die an die geklonten Netzknoten versendet werden.

Gefälschte Routing Informationen Beim adaptiven (dynamischen) Routing werden die Routing Tabellen automatisch anhand von Routing-Metriken erzeugt. Ziel des dynamischen Routings ist das Vermeiden von Netzwerkfehlern und Blockierungen.

Wird dieser Prozess durch gefälschte Routing-Informationen manipuliert, kommen andere Routing Tabellen zustande. Solch ein Angriff wird immer dann gestartet, wenn der Netzverkehr vermehrt über einzelne, schadhafte Netzknoten fließen soll. Dadurch können noch mehr Informationen aus dem Netzverkehr gezogen oder der Netzverkehr stärker beeinflusst werden.

Gefälschte ICMP Informationen Es können Fehlermeldungen, Diagnosen und Autokonfigurationen gefälscht werden. Dieser Angriff dient meist zur Verschleierung anderer Angriffe. Er kann aber auch angewendet werden, um den normalen Verkehrsfluss zu stören: Ein *Echo-Request* kann beispielsweise mit einem *Host unreachable* beantwortet werden, obwohl der Zielrechner erreichbar ist.

Blackhole Attacke Sobald ein schadhafter Netzknoten es geschafft hat, dass ein Teil des Netzverkehrs über ihn fließt, können die über ihn fließenden Nachrichten auf stille Weise verworfen werden (engl. *silent drop*). Dadurch verzögert sich der Netzverkehr und die Nachricht muss erneut gesendet werden.

Das stille Verwerfen von Nachrichten dient nicht nur dem Stören der Kommunikation, sondern auch zur Verschleierung anderer Angriffe: Wenn ein Netzknoten eine doppelt vorhandene Identifikation entdeckt und versucht zu melden, dann möchte der Angreifer diese manipulieren, verzögern oder verwerfen.

Sofern jede Nachricht verworfen wird, entsteht metaphorisch ein schwarzes Loch (engl. *blackhole*). Daher wird diese Attacke auch Blackhole Attacke genannt.

Grayhole Attack (Selective Forwarding Attacke) Das Verwerfen jeder Nachricht (Blackhole Angriff) verrät den Angreifer. Das Sensornetz versucht neue Routing-Pfade zu bilden, um diesen fehlerhaften Netzknoten zu umgehen.

Damit dies nicht passiert, gibt es den Grayhole Angriff, bei welchem sich metaphorisch das schwarze Loch kurz öffnet und wieder schließt. Einige Nachrichten werden folglich weitergeleitet, andere nicht. Deshalb wird diese Attacke auch Selective Forwarding Attacke genannt. Ergebnis des Angriffs ist es daher, den Nachrichtenverkehr gezielt oder willkürlich zu filtern.

Zur gezielten Störung bzw. zur Verschleierung anderer Angriffe können Nachrichten eines bestimmten Typs gefiltert werden. Alternativ kann der Netzverkehr wahllos in regelmäßigen oder unregelmäßigen Zeitabständen unterbrochen werden. Dies sorgt für eine Verzögerung des Nachrichtenflusses.

Spoofing Attacke Das Ziel eines Angreifers ist es, möglichst lange unentdeckt zu bleiben, um entsprechend lange Schaden anrichten zu können. Hierzu verwendet er verschiedene Täuschungsmethoden, um Authentifikations- und Identifikationsverfahren zu untergraben. Das Klonen von Netzknoten ist nur eine Methode, eine Spoofing Attacke durchzuführen. Das ACK-Spoofing ist eine weitere, bei der die Ankunft einer Nachricht beim Zielknoten vorgetäuscht wird.

Im Allgemeinen erfordert das Detektieren von Spoofing Attacken das Überwachen des Netzverkehrs auf auffällige Aktivitäten oder die stetige Authentifikation bei einer zentralen Instanz.

Sybil Attacke (Pseudospoofing Attacke) Sobald sich ein schadhafter Netzknoten authentifizieren und identifizieren konnte, besteht die Gefahr, dass der Angreifer eine Sybil Attacke ausführt: Der Angreifer erzeugt neben seiner Identität mehrere pseudonyme Identitäten, die alle einen physikalischen Netzknoten repräsentieren. Durch diese zusätzlichen Identitäten erhofft sich der Angreifer einen stärkeren Einfluss auf den Netzverkehr im Allgemeinen und auf Mehrheitsentscheidungen im Speziellen.

HELLO-flood Attacke In selbst organisierenden Sensornetzen kommen so genannte HELLO Nachrichten zum Einsatz. Durch diese kündigen Netzknoten ihren Nachbarn ihre Anwesenheit an. Nicht in das Netzwerk integrierte Netzknoten werden durch diese Ankündigungsart bemerkt und aufgenommen, sofern diese sich authentifizeren und identifizieren können.

Ein Angreifer nutzt nun diese Ankündigungsnachricht aus und verschickt sie in Massen, so dass eine HELLO Flut entsteht. Benachbarte Teilnehmer sind, bedingt durch diese Flut an Nachrichten, kaum noch in der Lage andere Nachrichten zu empfangen, wodurch der Nachrichtenfluss gestört wird.

Dynamische Routing-Protokolle setzen Routing-Metriken ein, um geeignete Nachbarknoten zu finden, die geringe Übertragungskosten zu anderen Knoten besitzen. Zur Ankündigung dieser Übertragungskosten werden HELLO Nachrichten eingesetzt. Anhand der eingegangenen Übertragungskosten benachbarter Knoten kann ein Netzknoten entscheiden, welcher Nachbar geringe Übertragungskosten zu anderen aufweist. Auf diese Weise ist das Netz fähig, sich selbst zu organisieren

und ungünstige Verbindungen zu vermeiden. Diese Selbstorganisation funktioniert allerdings nur, wenn *allen* Netzknoten vertraut werden kann.

Auch in dieser Situation kann der Angreifer mit Hilfe dieser Attacke das Vertrauen der anderen Netzteilnehmer missbrauchen und Konkurrenten unterdrücken, um sich selbst besser darzustellen. Der Netzverkehr wird folglich über den schadhaften Netzknoten geleitet.

Nutzt der Angreifer zusätzlich einen laptop-class Netzknoten, um diese Nachrichten zu verschicken, erreicht er u. U. Netzknoten, die ihm nicht antworten können, da er sich außerhalb der (normalen) Reichweite befindet. Dennoch versuchen sie sich zuerst mit ihm zu verbinden, so dass sich das Sensornetz während dieser Zeit in einen inkonsistenten Zustand befinden kann.

Sinkhole Attacke Ein Sinkhole Angriff ist ein nicht zu unterschätzender Routing-Angriff. Denn Ziel dieses Angriffs ist es, den Netzverkehr in einem Bereich des Netzwerks über einen einzigen, kompromittierten Netzknoten zu leiten, so dass metaphorisch ein Senkloch (engl. *sinkhole*) entsteht. Fließt ein beträchtlicher Teil der Nachrichten über solche Senklöcher, sind Folgeangriffe wie die Selective Forwarding Attacke leichter auszuführen. Je näher solche Angriffe an zentrale Instanzen ausgeführt werden, desto einflussreicher ist dieser Angriff.

Die einfachste Möglichkeit, den benachbarten Netzverkehr über den kompromittierten Netzknoten fließen zu lassen, ist es, bessere Übertragungskosten als benachbarte Netzknoten aufzuweisen oder vorzutäuschen.

Wormhole Attacke Der Angreifer ist in der Lage, laptop-class Netzteilnehmer zu verwenden, die im Gegensatz zu den meisten anderen Netzknoten mehr als nur eine Funktechnologie besitzen. Diese zweite Funktechnologie dient dann zum Aufbau eines Tunnels zwischen mind. zwei vom Angreifer kontrollierten Netzteilnehmern.

Erreicht der Tunnel wesentlich höhere Übertragungsgeschwindigkeiten bzw. geringere Latenzen als vergleichbare Routen im Sensornetz (vgl. WLAN mit LoWPAN), so kann der Angreifer (1) ein Wurmloch (engl. *wormhole*) zwischen Nachbarschaften erzeugen, (2) seine Angriffe unbemerkt koordinieren, (3) den Einfluss eines Sinkhole Angriffs verstärken.

Die unbemerkte Koordination der Angriffe durch Nutzung eines alternativen Funkkanals kann vom Sensornetz nicht entdeckt werden. Allerdings können nicht nur Angriffe koordiniert werden. Es ist möglich, die HELLO Nachrichten der anderen Teilnehmer (unbemerkt) weiterzuleiten. Durch diesen Tunnel können dann Netzknoten mit nur ein oder zwei Hops kommunizieren, obwohl sie wesentlich weiter entfernt sind. Sie glauben folglich, dass sie sich in der gleichen Nachbarschaft befinden. Dies kann insbesondere bei Routing-Protokollen, die Cluster bilden, Verwirrung stiften.

Wormhole-Sinkhole-Kombination Aber auch in zentralisierten Netzen hat die Platzierung eines schadhaften Netzknotens Nahe der zentralen Instanz eine besonders

große Auswirkung hat: Verbreitet nun der auf der anderen Seite des Tunnels befindliche, schadhafte Netzknoten, dass die zentrale Instanz nur wenige Hops entfernt ist, wird aufgrund der sehr geringen Übertragungskosten dieses Wurmloch für die Kommunikation bevorzugt. Dadurch fließt der gesamte Nachbarschaftsverkehr über diesen Netzknoten und es ist ein besonders großes Senkloch entstanden.

Daher ist die Kombination des Wormhole Angriffs mit der Sinkhole Attacke sehr attraktiv und effektiv, sofern der Angreifer eigene Netzknoten in das Netzwerk einschleusen kann. Die hierfür benötigten Identitäten hat er entweder von legithimen Netzknoten geklaut oder mit Hilfe des Sybil Angriffs erzeugt.

Es ist wichtig, einen Überblick zu möglichen Routing-Angriffen zu besitzen. Wie oft und in welcher Kombination diese Angriffe ausgeführt werden, ist schwer einzuschätzen. Die Wormhole-Sinkhole-Kombination sowie die Selective Forwarding Attacke stellen interessante Möglichkeiten dar, um sensible Hausautomatisierungsnachrichten wie das Melden eines Einbruchs zu unterdrücken.

Angenommen, es werden (auf Anwendungsschicht) vertrauliche Hausautomatisierungsnachrichten zwischen dem Regelwerkknoten und einem beliebigen Sensor- oder Aktorknoten verschickt. Dann kann der Angreifer trotz der Routing-Manipulation diese Nachrichten nicht einsehen. Um sensible Nachrichten dann garantiert zu unterdrücken, bleibt ihm folglich der Blackhole-Angriff (oder eine Jamming Attacke), wodurch er früher oder später entdeckt wird. Im Falle eines Einbruchs kann es dann bereits zu spät sein.

Das Ausführen eines Routing-Angriffs ist leichter, als ihn zu entdecken oder ihn abzuwehren. Da allerdings lediglich eine Störung, aber keine Manipulation des Automationsprozesses bewirkt wird, geht von dieser Bedrohung (bis auf spezielle Einzelfälle) eine **niedrige bis mäßige** Gefahr aus.

5.2. Sicherheitsanforderungen

Die Gefahrenanalyse zusammenfassend, gibt es zwei Angriffsarten, wie ein Angreifer ein Hausautomatisierungssystem – vor allem den Automationsprozess – belauschen oder manipulieren kann:

Angriffe auf Anwendungsebene greifen nicht nur die Anwendungsschicht an, sondern alle oberen Schichten des OSI-Referenzmodels. Zu diesen Schichten gehören ebenfalls die Transport-, Sitzungs- und Darstellungsschicht — folglich alle Schichten, die für eine Ende-zu-Ende-Verbindung (Multi-Hop-Kommunikation) benötigt werden.

Angriffe auf Routingebene haben die unteren Schichten des OSI-Referenzmodels zum Ziel, um direkt auf Punkt-zu-Punkt-Verbindungen einwirken zu können. Zu diesen Schichten zählen die Bitübertragungs-, Sicherungs- und Vermittlungsschicht. Mit diesen Angriffen kann der Angreifer auf die Kommunikation auf Anwendungebene (indirekt) Einfluss nehmen.

Deswegen werden die aufzustellenden Sicherheitsanforderungen nach diesen beiden Kriterien unterschieden.

5.2.1. Authentisierung auf Anwendungsebene

Auf Anwendungsebene kann ein Angreifer gefälschte Hausautomatisierungsdaten einschleusen. Solche Daten können gefälschte Sensorinformationen, aber auch gefälschte Aktoransteuerungen sein. Daher müssen Nachrichten auf Anwendungsebene mit einer Authentifikation versehen werden, anhand der sich die Kommunikationpartner gegenseitig ohne Zweifel erkennen können.

Automationsebene

In der Automationsebene des Hausautomatisierungssystems erfolgt die Kommunikation der Anwendungsebene zwischen Sensorknoten und Regelwerkknoten sowie zwischen Regelwerkknoten und Aktorknoten, da eine zentral-dedizierte Regelverarbeitung eingesetzt wird. Da der Regelwerkknoten die zentrale Instanz ist, die alle Hausautomatisierungsinformationen verarbeitet, ist es nicht erforderlich, dass zwei Sensorknoten und/oder Aktorknoten miteinander kommunizieren, d. h., ein Kommunikationspartner ist immer der Regelwerkknoten.

Jeder Sensor- und Aktorknoten hat demnach die Aufgabe, die Authentizität des Regelwerkknotens sicherstellen zu können. Im Gegenzug muss auch der Regelwerkknoten die Authentizität der anderen Netzteilnehmer prüfen können. Somit ist gewährleistet, dass kein beliebiger Angreifer (Outsider) die Identität eines anderen Teilnehmers missbrauchen kann, um gefälschte Hausautomatisierungsinformationen wie Sensorinformationen oder Aktoransteuerungen einzuschleusen.

Managementebene

In der Managementebene des Hausautomatisierungssystems erfolgt die Kommunikation der Anwendungsebene i. A. zwischen dem Regelwerkknoten und dem Hausbewohner, der ein beliebiges (mobiles) Endgerät verwendet und über das Internet kommuniziert. Der Regelwerkknoten beinhaltet das Interface sowie den Hausautomationskoordinator.

Auch hier ist es wichtig, dass sich beide Kommunikationspartner gegenseitig authentifizieren können, damit kein beliebiger Angreifer (Outsider) die Identität des Hausbewohners annehmen kann. Denn sonst wäre er nicht nur in der Lage das Hausautomatisierungssystem und all seine Prozesse zu überwachen, sondern er könnte sogar eigene Regeln zur Steuerung des Automationsprozesses integrieren oder vorhandene ändern bzw. löschen. Demnach ist es zusätzlich zur Authentisierung wichtig, eine Zugriffskontrolle zu intergrieren.

5.2.2. Integrität auf Anwendungsebene

Das Einfügen, Entfernen, Duplizieren, Ändern oder Fälschen jeglicher Informationen ist problematisch für ein Hausautomatisierungssystem, insbesondere auf Anwendungsebene.

Damit eine mögliche Gefährdung der Daten erkannt werden kann, ist der Schutz der Integrität eines der wichtigsten Schutzziele der Sicherheitsarchitektur.

Automationsebene

Wie in der Gefahrenanalyse gezeigt wurde, ist die Manipulation von Hausautomatisierungsdaten (Sensorinformationen und Aktoransteuerung) während des Automationsprozesses ein großes Problem. Unabhängig davon, ob die Daten eingesehen werden können oder nicht, können so gefälschte Informationen verbreitet werden. Dadurch ist ein Angreifer (Outsider) beispielsweise in der Lage, eine Einbruchskontrollsteuerung außer Gefecht zu setzen.

Managementebene

Aber auch auf Managementebene ist der Schutz der Integrität ein wichtiges Schutzbedürfnis. Schließlich darf es nicht möglich sein, dass ein Angreifer diejenigen Regeln, die der Hausbewohner neu konfiguriert, verändert werden. Auch darf es nicht passieren, dass das Hausautomatisierungssystem dem Nutzer mitteilt, dass alles läuft, obwohl der komplette Automationsprozess lahmgelegt wurde.

5.2.3. Vertraulichkeit auf Anwendungsebene

Das Belauschen des Hausautomatisierungssystem ist meist die Grundvoraussetzung für weitere Angriffe, da durch Belauschen der Angreifer das System kennen lernt. Zu diesen Informationen gehören harmlose Sensorinformationen wie die Temperatur oder die Luftfeuchtigkeit, aber auch Bewegungsdaten oder gar Aktoransteuerungen.

Zum Schutz der allgemeinen Persönlichkeitsrechte – vor allem zum Schutz der Privatsphäre – ist es daher sinnvoll, alle Daten auf Anwendungsebene vertraulich zu behandeln. So kann der Angreifer nicht sofort erkennen, welche Daten wirklich vertraulich behandelt werden müssen und welche nicht. Zudem kann es vorkommen, dass einige Daten zuerst nicht-vertraulich eingestuft werden und dann festgestellt wird, dass sie doch der Vertraulichkeit unterliegen. Bereits in der Einführung wurde darauf hingewiesen, dass Bewegungsdaten missbraucht werden können, obwohl sie lediglich zur Steuerung der Heizungsanlage verwendet werden.

Die Verwendung einer vertraulichen Kommunikation schützt demnach nicht nur die allgemeinen Persönlichkeitsrechte der Hausbewohner (und ihrer Gäste), sondern auch das System selbst vor dem Ausspähen durch den Angreifer. Denn das Überwachen des Hauses durch einen Einbrecher könnte zu neuartigen Koordinationen bei Einbrüchen führen und die Aufklärung durch die Polizei erschweren. Diese Situation mag zwar heute noch unvorstellbar sein, allerdings bei der Entwicklungsgeschwindigkeit des Stands der Technik in ein paar Jahren denkbare Realität.

Auch wenn in der heutigen Zeit neuartige Einbruchskoordinierungen noch unwahrscheinlich sind, so kann ein Einbrecher dennoch ein Haus mit einem unsicheren Hausautomatisierungssystem überwachen und so feststellen, ob jemand zuhause ist oder nicht.

Wenn der Angreifer aber erst die Vertraulichkeit brechen muss, um an diese Informationen heranzukommen, werden die meisten Angreifer abgeschreckt sein und erst gar nicht in Versuchung kommen, da es zu viel Aufwand bedeutet.

Bei der Vertraulichkeit auf Anwendungsebene muss allerdings beachtet werden, dass die IP-Adressen der Kommunikationspartner nicht geschützt sind. Daher kann zwar die Kommunikation nicht belauscht werden, doch können IP-Adressen der Kommunikationspartner sowie Zeitpunkt der Kommunikation festgehalten werden. Ein Angreifer wäre in der Lage die IP-Adresse bzw. das komplette IP-Paket (bis zur Schicht 3) zu manipulieren. Dieses muss durch entsprechende Maßnahmen auf Routingebene unterbunden werden. Dennoch sollte zur Identifikation der Kommunikationspartner nicht nur die IP-Adresse verwendet werden.

5.2.4. Authentisierung und Zugriffskontrolle auf Routingebene

Ein skalierbares, aber ungeschütztes Hausautomatisierungssystem erlaubt es dem Hausbewohner auf einfache Weise neue Netzknoten in das drahtlose Sensornetz zu integrieren. Doch erlaubt dies auch dem Angreifer (Outsider) eigene Netzknoten zu integrieren.

Wie in der Gefahrenanalyse gezeigt wurde, existieren auf Routingebene sehr viele Gefahren. Viele dieser Gefahren werden voraussichtlich zur Störung des Automationsprozesses eingesetzt werden. Die meisten dieser Angriffe erfordern die Voraussetzung, dass der Angreifer Netzknoten im Sensornetz kontrollieren kann, indem er entweder eigene Netzknoten verwendet oder aber andere mit schadhaften Programmen bespielt hat. Ist diese Voraussetzung erfüllt, versucht er den Netzverkehr zu beeinflussen, um eine möglichst hohe Kontrolle über das Routing zu erhalten. Dabei können verschiedene Angriffe ausgeführt werden, die beispielsweise einen (vorübergehenden) Ausfall des Systems herbeiführen können (z. B. per Jamming Attacke oder Blackhole Attacke). Andererseits kann auch die Kommunikation im Netz durch gezielte Störungen (z. B. mittels Selective Forwarding Attacken) oder durch Umleitungen (z. B. durch Sinkhole und Wormhole Attacken) beeinflusst werden, wodurch viele der Routing-Angriffe – vor allem in Kombination ausgeführt – ungeahnte Auswirkungen haben.

Geschickt koordinierte Angriffe stellen eine signifikante Herausforderung dar, selbst für sichere Routing-Protokolle. Mit begrenzten Mitteln – nahezu alle Netzknoten sind leistungsschwach – ist es nur mit hohem Aufwand möglich, diese Angriffe zu entdecken. Da dieser Aufwand der Anforderung nach Energieeffizienz (und langer Lebensdauer für batteriebetriebene Netzknoten) nicht gerecht wird und damit unwirtschaftlich ist, sollte auf eine Detektion von Routing-Angriffen verzichtet werden — oder nur in speziellen Hausautomatisierungsszenarien zum Einsatz kommen.

Daher ist es notwendig, dass die o. g. Voraussetzung für diese Angriffe nicht gegeben ist. Dieses lässt sich mit Hilfe einer Zugriffskontrolle (auf Routingebene) regeln. Das Verwenden einer Zugriffskontrolle auf Routingebene bewirkt allerdings, dass Routingknoten nicht auf einfache Weise hinzugefügt werden können. Sie müssen sich gegenüber dem Sensornetz authentisieren und als Routingknoten anmelden. Authentisierung wird

folglich zur Umsetzung der Zugriffskontrolle benötigt, damit Netzknoten nicht unerlaubt im Namen anderer handeln können.

5.2.5. Integrität auf Routingebene

Die Zugriffskontrolle auf Routingebene macht es erforderlich, dass sich Routingknoten als Router im Sensornetz anmelden müssen. Erst nach einer erfolgreichen Anmeldung werden sie in die Routing-Tabellen der anderen Router eingetragen.

Netzknoten ohne Routingfunktionalität (Endknoten) können demnach dem Sensornetz auf Routingebene ohne Probleme beitreten, da sie *theoretisch* keine Routing-Angriffe ausführen können. Da Zugriffskontrolle nicht vor Manipulation der Routingdaten – oder ähnlich sensibler Daten (bis zur Schicht 3) – schützt, können Gefahren von solchen schadhaften Endknoten ausgehen, auch wenn sie i. A. kein besonders hohes Sicherheitsrisiko darstellen.

Anders verhält es sich, wenn sich ein Angreifer (Outsider) durch Kompromittierung eines im Sensornetz befindlichen Netzknotens Zugang zum Sensornetz verschafft. Dann umgeht er die Zugriffskontrolle, wodurch er zum Insider wird. Dann ist er in der Lage, als Endknoten oder als Routingknoten Nachrichten von anderen zu manipulieren, sofern keine Integritätsprüfung die Daten schützt.

Daher ist es wichtig, dass die Sicherheitsarchitektur zusätzlich zur Zugriffskontrolle auch die Integrität der Daten auf Routingebene gewährleistet.

Wenn der Angreifer (Insider) einen Sinkhole Angriff durchführt, ohne dass er Nachrichten anderer Netzknoten manipulieren muss (z. B. mit Hilfe einer Sinkhole-Wormhole Attacke), ist es unmöglich, dieses mit geringem (energiesparendem) Aufwand zu verhindern. In solchen Situationen hilft es, die Auswirkungen eines erfolgreichen Angriffs zu minimieren. Durch geschickte Auswahl geeigneter Sicherheitsmaßnahmen in der Sicherheitsarchitektur kann die Gefahr, die von solch starken Angriffe ausgeht, vermindert werden, da der Erfolg im Gegensatz zum Aufwand zu gering ausfällt. Aufgrund dieser schlechten Erfolgswahrscheinlichkeit sind viele Angreifer abgeneigt, überhaupt einen Angriff auszuführen.

5.2.6. Zugriffskontrolle auf Anwendungsebene

Bisher muss die Sicherheitsarchitektur Maßnahmen zu folgenden Sicherheitsanforderungen bereitstellen:

1. Maßnahmen zur Authentisierung der Kommunikationsteilnehmer auf Anwendungsebene und auf Routingebene,

2. Maßnahmen zur Prüfung der Integrität der Daten kommunizierender Teilnehmer auf Anwendungsebene und auf Routingebene,

3. Maßnahmen zum Schutz der Vertraulichkeit kommunizierenden Teilnehmer auf Anwendungsebene,

4. eine Zugriffskontrolle auf Routingebene, um insbesondere das Anmelden unerlaubter Routingknoten zu erschweren.

Da allerdings auch die Anwendungsebene nicht vor Kompromittierungen geschützt ist, kann ein Angreifer vom Outsider zum Insider werden und so das System auf Anwendungsebene schädigen. Erst durch eine Zugriffskontrolle werden die Auswirkungen einer Kompromittierung vermindert: Wenn die Identität eines beliebigen Sensorknotens (Komponente der Automationsebene) auf das Interface des Regelwerkknotens (Komponente der Managementebene) zugreifen möchte, ist dies verdächtig und sollte unterbunden werden. Daher ist auch eine Zugriffskontrolle auf Anwendungsebene zwingend erforderlich.

Wenn das Hausautomatisierungssystem einen Mehrbenutzerbetrieb zulässt, dann ist bereits durch diesen eine Zugriffskontrolle nötig: Hierbei geht es nicht nur um die Trennung der Automationsebene von der Managementebene, sondern auch um die Steuerung des Zugriffs der einzelnen Nutzer anhand spezifischer Sicherheitsregeln.

5.2.7. Sicheres Speichern längerfristiger Informationen

Sollte es erforderlich oder vom Hausbewohner gewünscht sein, (personenbezogene) Informationen zu verarbeiten und längerfristig zu speichern, dann ist eine weitere Sicherheitsanforderung notwendig: Diese (personenbezogenen) Informationen müssen auf sichere Weise aufbewahrt werden und genauso sicher mit Hilfe der Zugriffskontrolle abrufbar sein.

Da der Automationsprozess über den Regelwerkknoten verläuft und dieser zugleich das Interface beinhaltet, ist es sinnvoll, die über ihn fließenden (personenbezogenen) Informationen in ihm längerfristig zu speichern, da er ein laptop-class Netzknoten ist und damit nicht den Einschränkungen der mote-class Geräte unterliegt. Damit stellt der Regelwerkknoten ein lohnendes Angriffsziel dar und muss somit vor Kompromittierungsversuchen so gut wie möglich geschützt werden.

Da diese Arbeit den Schwerpunkt auf die Kommunikation im Sensornetz des Hausautomatisierungssystem legt, wird diese Sicherheitsanforderung zwar definiert, aber nicht weiter betrachtet.

5.2.8. Unökonomische Sicherheitsanforderungen

Zuletzt werden im Abschnitt *Sicherheitsanforderungen* nützliche, aber unökonomische Sicherheitsanforderungen diskutiert. Sie sind unökonomisch, weil sie den Anforderungen eines Hausautomatisierungssystems nicht gerecht werden, insbesondere aufgrund der Eigenschaften der verschiedenen Netzknoten und aufgrund der Energiesparsamkeit des Hausautomatisierungssystems.

Nicht-Abstreitbarkeit auf Anwendungsebene

Durch eine Gewährleistung der Nicht-Abstreitbarkeit bzw. Verbindlichkeit der Daten ist es möglich, eine Nachricht eindeutig einem Sender oder Empfänger zuzuordnen, so dass

diese Nachricht nicht geleugnet werden kann. Ein Online-Händler möchte sicherstellen, dass seine Pakete beim richtigen Empfänger ankommen. Die Postzustellung nutzt hierfür die Unterschrift des Empfängers, der dann nicht leugnen kann, dass er ein bestimmtes Paket bekommen hat.

Im Hausautomatisierungssystem verhält es sich ähnlich: Der Regelwerkknoten ist dafür verantwortlich, dass der Automationsprozess einwandfrei funktioniert. Hierzu muss er garantieren können, dass alle Netzknoten seinen Nachrichten und Anweisungen Folge leisten. Das bekannteste kryptographische Schutzkonzept sind digitale Signaturen und Zertifikate. Da solche Signaturen bzw. Zertifikate allerdings Algorithmen benötigen, die nicht energiesparsam genug sind, ist dieses Schutzziel nicht ohne übermäßig erhöhtem Aufwand umsetzbar.

Vertraulichkeit auf Routingebene

Eine vertrauliche Nachricht ist nur für einen bestimmten Empfängerkreis vorgesehen. Je kleiner dieser Empfängerkreis ist, desto mehr Aufwand muss betrieben werden, um die Vertraulichkeit aufrechtzuerhalten. Eine vertrauliche Kommunikation zwischen zwei Kommunikationspartnern bietet die größte Sicherheit: Wenn alle Geheimnisse eines Netzteilnehmers durch Kompromittierung entwendet wurden, bleiben alle anderen vertraulichen Verbindungen weiterhin erhalten.

Angenommen, eine Nachricht wird von einem Sensorknoten S zum Regelwerkknoten R versendet und muss über die Routingknoten r_1 und r_2 weitergeleitet werden. Dann bewirkt das auf Routingebene vertrauliche Versenden der Nachricht (unter Nutzung einer Verbindungsverschlüsselung, engl. *link encryption*), dass sie:

1. bei S mit dem Schlüssel k_{S,r_1} verschlüsselt wird,
2. von r_1 mit dem Schlüssel k_{S,r_1} entschlüsselt werden muss
   (um bspw. die IP-Adresse von R auszulesen),
3. danach von r_1 wieder verschlüsselt wird und zwar mit dem Schlüssel k_{r_1,r_2},
4. von r_2 mit dem Schlüssel k_{r_1,r_2} entschlüsselt werden muss
   (um bspw. die IP-Adresse von R auszulesen),
5. danach von r_2 wieder verschlüsselt wird und zwar mit dem Schlüssel $k_{r_2,R}$,
6. bei R mit dem Schlüssel $k_{r_2,R}$ entschlüsselt wird.

Bei Verwendung eines Netz-weiten Schlüssels ($k_{S,r_1} = k_{r_1,r_2} = k_{r_2,R}$) werden überflüssige Ver- und Entschlüsselungen durchgeführt. Es vereinfacht aber die Verteilung des Schlüssels. Das Verwenden von paarweisen Schlüsseln ($k_{S,r_1} \neq k_{r_1,r_2} \neq k_{r_2,R}$) erfordert eine wesentlich aufwändigere Verteilung, da diese Geheimnisse auf sichere Weise ausgetauscht werden müssen (symmetrischer Kryptographie) oder mit viel Rechenaufwand erzeugt werden müssen (asymmetrische Kryptographie).

Deswegen ist die Umsetzung der Vertraulichkeit nur mit Netz-weiten Schlüsseln denkbar. Befinden sich kompromittierter Routingknoten auf dem Routingpfad, so hilft auch das Verschlüsseln nicht vor einem Routing-Angriff, da eine Sicherheitsbeziehung auf Routingebene lediglich zwischen zwei physikalisch benachbarten Netzknoten bestehen kann.

5.3. Sicherheitsmaßnahmen verwandter Projekte

5.3.1. HexaBus Home Automation System

Das HexaBus Home Automation System (siehe auch Unterabschnitt 3.5.1) ist eines der ersten Hausautomatisierungsprojekte, das eine Sicherheitsarchitektur für drahtlose Sensornetze mit 6LoWPAN in der Hausautomatisierung umgesetzt hat. In den Forschungsnachrichten der Frauenhofer-Gesellschaft wird es daher unter der Überschrift „Guaranteed data security“ wie folgt beschrieben: „Users have no need to worry about the security of their data – all information is transmitted in encrypted form.“ [60]

Dabei verwendet das System vorzugsweise eine in Hardware integrierte AES-CCM-128 Verschlüsselung, so dass diese transparent eingesetzt werden kann. Sie wird auf der Sicherungsschicht, dem IEEE 802.15.4 Standard entsprechend, in Form einer Verbindungsverschlüsselung durchgeführt. Dabei kommt ein einziger Netz-weiten Schlüssel zum Einsatz, der als Pre-Shared-Key auf allen Netzknoten des Sensornetzes vorinstalliert wird. Somit entfällt ein Schlüsselaustausch, aber auch ein Schlüssel-Update (Re-Keying), so dass mit dieser Methode lediglich Angreifer ohne Insider-Informationen (Outsider) abgehalten werden können. Ist ein Angreifer durch eine Kompromittierung eines Netzknotens in den Besitz des Schlüssels gelangt, dann hilft nur noch ein Neu-Ausbringen (bzw. ein manuelles Schlüssel-Update) aller Netzknoten.

Folglich obliegt dem Hausbewohner die Gewährleistung der Sicherheit und erfüllt nicht hinreichend die Anforderung zum Schutz des allgemeinen Persönlichkeitsrechts (siehe Unterabschnitt 2.2.1).

5.3.2. Thingsquare System

Thingsquare ist eine in 2012 gegründete Firma, die Lösungen in Bereichen „smart metering, smart lighting, city networking, connected home devices, and wireless sensors“ entwickelt [61]. Dabei steht nicht die Entwicklung eines eigenständigen Hausautomatisierungssystems im Vordergrund, sondern die Entwicklung von Anwendungen zum Internet der Dinge: Es soll eine Kommunikation zwischen (Smartphone) Apps und Dingen wie Thermostaten, Glühlampen sowie Straßenlaternen ermöglicht werden [62, 63].

Thingsquare entwickelt eine open-source Firmware, die unter anderem das Contiki-OS Betriebssystem enthält. Weiterhin ist eine AES-CCM-128 Implementation für dieses Betriebssystem enthalten, das auf Sicherungsschicht eingesetzt werden kann. Diese hält sich im Gegensatz zur HexaBus Implementation nicht an den IEEE 802.15.4 Standard, da eine Netz-weite Ver- und Entschlüsselung ohne Prüfung des Security-Bits im 802.15.4-Paket durchgeführt wird, so dass fehlerhaft entschlüsselte Pakete stillschweigend verworfen werden. Auf diese Weise wird der Zugriff auf das Sensornetz nur denjenigen Netzknoten gestattet, die diesen Pre-Shared-Key besitzen.

Seit November dieses Jahres sind erste Informationen zu einem Thingsquare Evaluation Demo System [64] verfügbar, das einem Hausautomatisierungssystem augenscheinlich nahe kommt. Eingesetzt werden Geräte, die mittels 6LoWPAN mit einem Router kommunizieren, der am Ethernet LAN angeschlossen ist. So wird das Sensornetz mit dem

Internet verbunden, um das Internet der Dinge zu schaffen.

Über einen HTTP Webserver (Thingsquare Demo Server) können nun die im Sensornetz vorhandenen Sensorknoten registiert und verwaltet werden. Nach der Registrierung sind verschiedene Übersichten zum Sensornetz einsehbar, u. a. eine Auflistung aller Sensoren und Leuchten sowie eine Wireless Network Map (ähnlich der im Cooja Simulator von Contiki).

Der Registrier-Vorgang läuft dabei wie folgt ab: Das „Ding“, das registriert werden soll, ist mit einem LCD Display ausgestattet und zeigt einen PIN an (bspw. 80850). Dieser wird dann im Regristrierungsbildschirm im Webbrowser eingegeben und das Gerät kann konfiguriert werden. Auch das Updaten der Firmware über dieses System ist möglich.

5.4. Sicherheitsmechanismen

In diesem Abschnitt werden nun die Sicherheitsmechanismen besprochen, die die Sicherheitsarchitektur integrieren soll. Hierbei werden die zuvor in Abschnitt 5.2 definierten Sicherheitsanforderungen beachtet.

Der bisherige Ansatz verwandter Hausautomatisierungsprojekte, die eine Sicherheitsarchitektur integriert haben, ist die Integration von Sicherheitsmechanismen auf Sicherungsschicht gewesen. Diese sind für den Endverbraucher allerdings nicht hinreichend geeignet — wie es in Unterabschnitt 5.3.1 diskutiert worden ist. Daher müssen alternative Sicherheitsmechanismen gefunden werden. Insbesondere ist die Verwendung von Ende-zu-Ende-Verschlüsselungen notwendig.

Beim Thingsquare System scheint es eine Anwendung zu geben, die einen PIN generiert und über die der Server den Netzknoten erkennt und ins System aufnimmt. Hierzu ist nur ein kleiner Schritt des Nutzers nötig. Allerdings sind nicht genug Informationen vorhanden, um bewerten zu können, ob anhand der PIN eine vertrauliche Ende-zu-Ende-Kommunikation stattfindet.

Daher wird in diesem Abschnitt damit begonnen, wie die Vertraulichkeit und Integrität auf Anwendungsebene hergestellt wird. Danach wird die Gewährleistung der Authentisierung thematisiert und zuletzt auf die Integritätssicherung auf Routingebene eingegangen.

5.4.1. Integrität und Vertraulichkeit auf Anwendungsebene

Das mit einer dedizierten Regelverarbeitung ausgestattete Hausautomatisierungssystem (siehe Abschnitt 3.6) bildet eine zentrale Netzinfrastruktur, in welcher der Regelwerkknoten das Zentrum ist. Daher kommuniziert jeder Netzknoten auf Anwendungsebene (ausschließlich) mit diesem. Hinzu kommt, dass er auch leistungsstark ist, folglich über weit mehr Ressourcen verfügt, als leistungsschwache Netzknoten. Diese Eigenschaft kann genutzt werden, um die Authentisierung, Integrität und Vertraulichkeit auf einfache und flexible Weise zu gewährleisten.

Durch diese Unicast-Kommunikation besitzt jeder Netzknoten (Sensor-, Aktor- und Routingknoten sowie der User-Netzknoten des Hausbewohners) lediglich ein Geheimnis,

welches mit dem Regelwerkknoten geteilt wird. Somit muss sich jeder nur vor ihm authentisieren können. Daher sind folgende Sicherheitsmaßnahmen geplant:

1. Der Einsatz von paarweisen, symmetrischen Schlüsseln ist aufgrund der zentralisierten Kommunikationsweise vorgesehen.

2. DTLS soll die Einigung auf einen Verschlüsselungsalgorithmus ermöglichen.

Weiterhin wird der Einsatz von AEAD Verschlüsselungsalgorithmen empfohlen (siehe Unterabschnitt 4.1.6). Beispiele sind AES-CCM und AES-GCM. Dadurch wird die Integrität und Vertraulichkeit der Daten mittels symmetrischer Kryptographie und unter Verwendung eines Schlüssels sichergestellt. Dabei sorgt ein regelmäßiger Re-Handshake durch DTLS für die fortwährende Gewährleistung der Vertraulichkeit sowie der Datenintegrität.

Verwendung von paarweisen symmetrischen Schlüsseln

Da jeder Netzknoten eine eigene vertrauliche Verbindung mit dem Regelwerkknoten eingeht, ist der Einsatz paarweiser Schlüssel (engl. *dedicated pairwise keys*) geeignet. Damit verwaltet der Regelwerkknoten bei einer Netzgröße von n Netzknoten $n-1$ Geheimnisse. Jeder andere Netzknoten bewahrt lediglich ein Geheimnis. Damit ist der Speicherverbrauch auf den leistungsschwachen Netzknoten für das Bewahren der Geheimnisse optimal ausgenutzt.

Verwendung von DTLS als Verschlüsselungsprotokoll

In einem offenen Hausautomatisierungssystem ist es wichtig, dass die Interoperabilität durch existierende Sicherheitsprotokolle garantiert bleibt und weiterhin gefördert wird, insbesondere wenn sich die Technologien weiterentwickeln. Diese Sicherheitsstandards beschreiben, wie neue Technologien in vorhandene Sicherheitsmechanismen integriert werden, um eine maximale Sicherheit erreichen zu können, ohne die Sicherheitsarchitektur grundlegend ändern zu müssen.

Weiterhin kann die vorgeschriebene Verwendung eines einzigen Verschlüsselungsalgorithmus (bspw. AES-128-CCM-8) dazu führen, dass das Sicherheitssystem über die Jahre an Sicherheit abnimmt, so dass es irgendwann komplett ausgetauscht werden muss. Daher wird ein Verschlüsselungsprotokoll benötigt, das die Verschlüsselungstechnik auf sichere Weise aushandelt.

Aus diesem zwei Gründen kommen folgende, standardisierten Sicherheitsprotokolle in Frage, die die Authentisierung, Integrität und Vertraulichkeit auf Anwendungsebene organisieren:

- IPsec (auf Vermittlungsschicht)
- TLS bzw. DTLS (zwischen Transport- und Anwendungsschicht)

Unterschiede zwischen diesen beiden Verschlüsselungsprotokollen wurden bereits in Abschnitt 4.2 diskutiert.

Neben diesen gibt es allerdings noch ein weiteren Unterschied: Denn wenn sich der Regelwerkknoten außerhalb des 6LoWPAN Sensornetzes befindet, kommt ein 6LoWPAN

Border Router zum Einsatz, um die Nachrichten aus dem Sensornetz zum Regelwerkknoten weiterzuleiten (und umgekehrt). Im 6LoWPAN kommt eine Header Compression (HC) zur Verringerung des Kommunikationsaufkommens zum Einsatz. Diese wird am Border Router entfernt. Solche Header Compressions finden auch auf Protokolle statt, die über der Vermittlungsschicht liegen (bspw. UDP). Wird nun auf Vermittlungsschicht wie bei IPsec der IP-Payload verschlüsselt, so wird auch die UDP-HC mit verschlüsselt, die der Border Router nicht entfernen kann. Bei Verwendung von DTLS wird lediglich die Anwendungsschicht verschlüsselt, so dass der Border Router nicht in seiner Komprimierungsaufgabe gestört wird.

Weiterhin unterscheidet sich die Art und Weise der Implementation: Während IPsec in den Betriebssystemkern integriert wird, ist dies bei DTLS nicht erforderlich und kann somit in jedem Betriebssystem nachgerüstet werden. Dies betrifft insbesondere Betriebssysteme von mobilen Endgeräten, bei denen der Nutzer i. A. keine Änderungen am System vornehmen darf.

Aufgrund der genannten Nachteile von IPsec soll folglich das Datagram Transport Layer Security (DTLS) Protokoll die Aufgabe übernehmen, einen Verschlüsselungsalgorithmus auszuhandeln.

Kombination von gegenwärtigen und zukünftigen Sicherheitsstandards

Aufgrund der zentralen Netzinfrastruktur und der Verwendung eines Verschlüsselungsprotokolls, das ein sicheres Aushandeln von Verschlüsselungstechniken ermöglicht, konnte folgende Erkenntnis gewonnen werden:

Da der Regelwerkknoten ein laptop-class Netzknoten ist und damit nicht den Einschränkungen der anderen Netzknoten unterliegt, kann er (theoretisch) beliebig viele Geheimnisse bewahren und kennt beliebig viele kryptographische Verfahren. Das Kennen beliebig vieler kryptographischer Verfahren hat den entscheidenden Vorteil, dass verschiedene Sicherheitstechnologien in einem Hausautomatisierungssytem vereint werden können: Extrem energiesparende Sensorknoten (wie solche mit Temperatursensoren) verwenden DTLS_PSK_WITH_AES_128_CCM_8[1]. Andere nutzen einen Schlüsselaustausch auf Basis von DHE_PSK. Bewegungsmelder oder andere potentiell gefährliche Sensorknoten können stärkere Sicherheitstechnologien wie beispielsweise DTLS_DHE_RSA_WITH_AES_256_GCM_SHA384 verwenden und sind mit hinreichend hohem Speicher ausgestattet. Daher können neu hinzugekaufte Netzknoten aktuelle und sichere kryptographische Verfahren mitbringen und müssen nicht alte und unsichere Standards utnerstützen, um mit dem Hausautomatisierungssystem kompatibel zu sein.

Zusammengefasst, stellt diese Erkenntnis ein wichtiger Schritt in Richtung offene Hausautomatisierung dar, da im selben Hausautomatisierungsnetzwerk sowohl gegenwärtige wie auch zukünftige Sicherheitsstandards verwendet werden können.

[1] An dieser Stelle wird nochmal darauf hingewiesen, dass es nicht empfohlen ist, lediglich PSK mit einem beständigen Schlüssel zu verwenden, da er wahrscheinlich nicht aktualisiert wird.

5.4.2. Authentisierung auf Anwendungsebene

Die Sicherheitsanforderung zur Gewährleistung der Authentisierung auf Anwendungsebene besagt, dass es einem Angreifer (Outsider) nicht möglich sein darf, im Namen eines anderen Netzteilnehmers zu handeln. In der Managementebene können hierzu digitale Signaturen eingesetzt werden, da zwei leistungsstarke Netzknoten miteinander kommunizieren.

In der Automationsebene ist diese Anforderung schwieriger umzusetzen, da leistungsschwache Netzknoten vorhanden sind. So soll die Authentizität anhand eines Pre-Shared-Keys (PSK) in Kombination mit einer Identität (PSK ID) sichergestellt werden, da dieser ein Geheimnis beider Kommunikationspartner darstellt. Dadurch wird eine Signaturerzeugung und -verifikation überflüssig. Wie diese Schlüssel ausgetauscht werden, wird im folgenden Abschnitt 5.5 diskutiert.

Weiterhin ist, durch den NSA Skandal verursacht, die Anforderung nach Perfect Forward Secrecy (PFS) gestiegen [65], weshalb sie an dieser Stelle Erwähnung findet. Denn Perfect Forward Secrecy kann auch unter Verwendung von PSKs erreicht werden.

Einsatz von Pre-Shared Keys zur Authentisierung

Die Aufgabe des Datagram Transport Layer Security Protokolls ist es, dass sich zwei Kommunikationspartner auf ein Verschlüsselungsverfahren sowie auf einen symmetrischen Schlüssel einigen können. Dieses Schlüsselaustauschverfahren basiert meist auf asymmetrischer Kryptographie (bspw. RSA oder Diffie-Hellman).

Damit asymmetrische Verfahren – vor allem die aufwendige Generierung und Verifizierung digitaler Signaturen – in Geräten mit eingeschränkten Ressourcen eingespart werden können, ist DTLS deshalb seit Ende 2005 um ein weiteres Schlüsselaustausch-Prinzip erweitert worden: Pre-Shared Keys (PSKs). Sie sind in RFC 4279 [29] definiert worden. Durch PSKs wird eine Authentisierung ermöglicht, ohne eine Signatur oder gleichwertige Verfahren anzuwenden. Dabei wird der PSK *nicht* über den Kommunikationskanal ausgetauscht, sondern auf andere Weise — manchmal auch Out-of-Bound-Kanal genannt. Allerdings muss beachtet werden, dass die Entropie dieses Pre-Shared Keys nicht zu niedrig ist. Diese fällt niedrig aus, wenn ein Schlüssel fester Länge oder ein lesbarer Schlüssel eingesetzt wird.

Ferner dienen PSKs nicht nur als Ersatz für andere Schlüsselaustauschverfahren. Sie können auch mit ihnen kombiniert werden:

- RSA_PSK — Wie bei Einsatz von RSA wird ein (mit dem öffentlichen RSA-Schlüssel des Servers) verschlüsseltes PreMasterSecret vom DTLS Client an den Server gesendet. Zusätzlich wird die PSK ID mitgeteilt, die verwendet werden soll. Der Server sendet seinerseits einen Hinweis (PSK ID HINT), damit der Client weiß, welcher PSK zum Einsatz kommt. PreMasterSecret und PSK bilden dann die Grundlage zur Erzeugung des gemeinsamen symmetrischen Schlüssels.
- DHE_PSK — Es wird ein kurzlebiger (engl. *ephemeral*) Diffie-Hellman-Schlüsselaustausch durchgeführt, bei dem die Authentisierung nicht gewährleistet ist und damit theoretisch eine Man-in-the-Middle Attacke zulässt. Doch die Verwendung

eines PSK verhindert eine erfolgreiche Ausführung, da sonst die Erzeugung des gemeinsamen Schlüssels fehlschlägt. Denn auch bei dieser Methode wird von beiden Seiten eine Identität beim Schlüsselaustausch mitgesendet.

Je nach Netzknoten kann eines dieser Schlüsselaustauschverfahren, basierend auf PSKs, zur Anwendung kommen.

Perfect Forward Secrecy

Ein weiteres Ziel, das zwar nicht zwingend notwendig ist, aber verfolgt werden sollte, ist die Gewährleistung einer Perfect Forward Secrecy (PFS) zur zukunftssicheren Verschlüsselung [65]. Das bedeutet, dass ein neuer Schlüsselaustausch die Vertraulichkeit wiederherstellt, auch wenn der vorherige Schlüssel kompromittiert werden konnte.

In der Automationsebene können PSK und RSA_PSK nicht eingesetzt werden, um Perfect Forward Secrecy zu erreichen, wenn auf irgendeine Weise der PSK oder der private RSA-Schlüssel des DTLS Servers erraten wird. Dann ist der Angreifer in der Lage, aufgezeichnete Kommunikationen zu entschlüsseln. DHE_PSK bietet dagegen schon PFS, sofern bei jedem Handshake ein neuer privater Diffie-Hellman-Schlüssel zum Einsatz kommt [29, S. 9]. Lediglich beim Kenntnis des PSK ist ein Man-in-the-Middle Angriff möglich, der allerdings bei jedem Handshake ausgeführt werden muss. Dieser Angriff kann nur ausgeführt werden, wenn der Angreifer Zugang zum Sensornetz (den netzweiten Verbindungsschlüssel) besitzt.

Auf Managementebene kann das Schlüsselaustauschverfahren DHE_RSA zum Einsatz kommen, bei dem ein Diffie-Hellman-Schlüsselaustausch stattfindet, der mit einer RSA-Signatur geschützt ist und damit einen Man-in-the-Middle Angriff verhindert.

5.4.3. Authentisierung und Integrität auf Routingebene

Wie in der Gefahrenanalyse gezeigt wurde, bestehen vielfältige Angriffe im Sensornetz auf Routingebene. Datenintegrität soll folglich eine Manipulation der gerouteten Nachrichten verhindern. Zudem müssen sich Netzknoten authentisieren können.

Auf Sicherungsschicht (OSI-Schicht 2) stehen zur Gewährleistung von Vertraulichkeit, Datenintegrität und Datenauthenzität drei Cipher Suites durch den IEEE 802.15.4-2003/2006 Standard bereit:

- AES-CTR – zur Gewährleistung der Vertraulichkeit ohne Datenintegrität und Authenzität. Ein CRC überprüft die Korrektheit der übermittelten Daten. Vor dem Einsatz dieser Cipher Suite wird in [48] gewarnt.
- AES-CBC-MAC – zur Gewährleistung der Datenintegrität und Authenzität ohne Vertraulichkeit.
- AES-CCM* – zur Gewährleistung der Vertraulichkeit, Datenintegrität und Datenauthenzität.

Die Authentisierung wird, wie bereits im vorherigen Unterabschnitt diskutiert, durch die Verwendung eines gemeinsamen Geheimnisses (pre-shared AES-Key) in Verbindung mit einem Message Authentication Code (MAC) sichergestellt, da kein anderer diesen PSK kennt. Im Falle von AES-CBC-MAC (und AES-CCM*) wird ein Message Authentication

Code (MAC) berechnet, der entweder 4 B, 8 B oder 16 B lang sein kann und der Nachricht angehangen wird. Diesen überprüft die Gegenseite, um festzustellen, ob die Nachricht authentisch und integre ist.

Durch die Verwendung eines netzweiten Schlüssels entstehen weitere Probleme: Die 802.15.4-Pakete besitzen keinen Wiedereinspielungsschutz, wodurch die Stärke der Datenintegrität abgeschwächt wird. Zudem ist bei der Verwendung von 802.15.4-ACK-Paketen kein Schutz der Integrität vorgesehen, wodurch sie vermieden werden sollten. Die Ankunft der Pakete sollte dann auf höheren Schichten bestätigt werden (bspw. der Anwendungsschicht durch CoAP). Beim DTLS-Handshake wird üblicherweise ein Timeout-Interval verwendet, der die letzte Nachricht erneut sendet, wenn die erwartete Antwort nicht innerhalb einer bestimmten Zeitspanne angekommen ist.

Weiterhin muss bedacht werden, dass eine „ganz oder gar nicht“ Zugriffssteuerung durch einen einzigen netzweiten Verbindungsschlüssel ermöglicht wird: Ist ein Netzknoten im Besitz dieses Schlüssels, wird es ihm ermöglicht, sich am Routing (RPL) ohne weitere Prüfung zu beteiligen.

5.5. Schlüsselmanagement und Zugriffskontrolle

Da die Managementebene ein klassisches Netzwerk repräsentiert, bereitet sas Schlüsselmanagement auf Managementebene wenig Probleme. Interessant wird es auf der Automationsebene, wenn neue Netzknoten in das drahtlos kommunizierende Sensornetz mit 6LoWPAN aufgenommen werden sollen. Neben der Aufnahme (dem erstmaligen Anmelden) gilt es auch ein regelmäßiges Schlüssel-Update (Schlüsselaustausch) sicherzustellen. In diesem Abschnitt wird folglich das Schlüsselmanagement und die Zugriffskontrolle vorwiegend auf der Automationsebene diskutiert, da diese Aufgabe im Internet der Dinge und damit auch in der drahtlosen, internetfähigen und offenen Hausautomatisierung wenig diskutiert worden ist.

Die Sicherheitsmechanismen, die zur Anwendung kommen, erfordern, dass folgende zwei Schlüsseltypen sowie ein Pre-Shared Key verwaltet werden müssen:

1. auf Sicherungsschicht ein netzweiter Verbindungsschlüssel (auch L2-Schlüssel genannt), der eine integre sowie authentische Hop-by-Hop-Kommunikation im 6LoWPAN-Sensornetz ermöglicht

2. auf Transportschicht ein paarweiser Schlüssel (auch L4-Schlüssel genannt), der eine vertrauliche, integre sowie authentische Ende-zu-Ende-Kommunikation zwischen Regelwerkknoten und Sensor- / Aktorknoten ermöglicht;

3. zur Erzeugung des L4-Schlüssels wird ein Pre-Shared Key (auch L4-PSK genannt) benötigt, der über einen Out-of-Bound-Kanal ausgetauscht werden muss.

Zur Beschreibung des Schlüsselmanagements und der Zugriffskontrolle wird nachfolgend wie folgt vorgegangen: (1) Zuerst wird der L4-Schlüsselaustausch beschrieben, da dieser ein wesentlicher Bestandteil der Sicherheitsarchitektur ist und für eine zukunftssichere Verschlüsselung sorgen soll. (2) Danach werden drei verschiedene Methoden

beschrieben, wie der L2-Schlüsselaustausch aussehen kann. Dabei ist es wichtig zu wissen, wie der L4-Schlüsselablauf funktioniert, um zu verstehen, warum diese Reihenfolge (erst L2, dann L4) gewählt worden ist. (3) Zum Schluss wird der komplette Anmeldevorgang geschildert, der eine Zugriffssteuerung auf Anwendungsebene ermöglicht.

Wie das Thingsquare System demonstriert, gibt es weitere Lösungsmöglichkeiten, wovon jede ihre Vor- und Nachteile hat. An dieser Stelle sei darauf hingewiesen, dass beim Anmeldevorgang im Thingsquare System davon ausgegangen wird, dass die Netzknoten ein Display besitzen. Diese Voraussetzung ist in diesem Hausautomatisierungssystem nicht erfüllt. Daher müssen alternative Methoden gefunden werden, um eine Zugriffssteuerung durch den Hausbewohner zu ermöglichen.

5.5.1. L4-Schlüsselaustausch

Es wird angenommen, dass zum Zeitpunkt des L4-Schlüsselaustauschs der L2-Schlüssel bereits ausgetauscht ist, so dass die Kommunikation bereits auf Sicherungsschicht vor Angreifern (Outsidern) manipulationssicher geschützt ist. Dabei ist es nebensächlich, dass der Angreifer dennoch alle auszutauschenden Informationen einsehen kann.

Weiterhin wird angenommen, dass auch der L4-Pre-Shared Key über den Out-of-Bound-Kanal bereits ausgetauscht worden ist, so dass beide Kommunikationspartner diesen sowie die zugehörige Identität kennen. Allerdings sollte anhand der Identität nicht die Funktionalität des Netzknotens ableitbar sein (bspw. „Bewegungsmelder-Flur“), da diese für den Angreifer einsehbar ist. Diese Annahme ist zudem nützlich, wenn der Angreifer sich Zugang zum Sensornetz verschaffen konnte, indem er in den Besitz des L2-Schlüssels gelangt ist, aber die vertrauliche Kommunikation auf Transportschicht nicht einsehen kann. Dieser Angreifer könnte bei Verwendung aussagekräftiger Identitäten gezielt wichtige Kommunikationen angreifen (bspw. durch Störungen auf Routingebene).

Wie bereits in Unterabschnitt 5.4.1 erklärt worden ist, kommt das Datagram Transport Layer Security (DTLS) Protokoll zum Einsatz, um den L4-Schlüssel auszutauschen. Dabei soll ein Verfahren eingesetzt werden, das nach Unterabschnitt 5.4.2 möglichst eine zukunftssichere Verschlüsselung (Perfect Forward Secrecy) ermöglicht. Zusammen mit dem L4-PSK kommt bisher nur DHE_PSK in Frage. Dabei ist der kurzlebige Diffie-Hellman-Schlüsselaustausch eine gern genutzte Technologie in klassischen Netzwerken, so dass versucht werden sollte, diese Technologie auch in drahtlosen Sensornetzen einzusetzen.

Die Alternative ist die Verwendung von RSA_PSK, bei welchem der DTLS Server (typischerweise der Regelwerkknoten) dem Client seinen öffentlichen RSA-Schlüssel übermittelt, damit der Client ihn zur Verschlüsselung des Pre-Master-Secret verwenden kann. Dann ist nur der Server in der Lage, diese Verschlüsselung wieder zu entfernen. Da der Regelwerkknoten leistungsstark ist, kann er jederzeit den RSA-Schlüssel erneuern, da das Verwenden einer Zertifizierungsinstanz im Hausautomatisierungssystem nicht vorgesehen ist. Solange der PSK dem Angreifer nicht bekannt ist, ist diese Zertifizierungsinstanz nicht erforderlich, da ein gefälschter RSA-Schlüssel durch einen Man-in-the-Middle Angriff dennoch zu einem falschen Master-Secret und damit zu einem falschen symmetrischen L4-Schlüssel führt.

Bei der Verwendung von Pre-Shared Keys ist allerdings Vorsicht geboten: Werden zu oft fehlerhafte Schlüsselaustauschsversuche oder fehlerhaft verschlüsselte Nachrichten bemerkt, handelt es sich mit hoher Wahrscheinlichkeit um einen Brute-Force Angriff, bei dem versucht wird, den PSK zu erraten. Im Zweifelsfall muss die Kommunikation eingestellt werden und ein Warnsignal an den Regelwerkknoten oder dem Hausbewohner direkt übermittelt werden.

Solange kein Angriff vorliegt, wird der L4-Schlüsselaustausch regelmäßig (bspw. einmal am Tag) wiederholt. Die genaue Wiederholungsrate ist allerdings davon abhängig, wie viele und wie oft Nutzdaten übermittelt werden und welche symmetrische Verschlüsselungstechnik eingesetzt wird. Im Regelfall hilft ein Counter, der diesen Zeitpunkt bestimmt und bei Überschreitung eines Schwellwerts einen DTLS Re-Handshake durchführt.

5.5.2. L2-Schlüsselaustausch

Es wird angenommen, dass der L4-Schlüsselaustausch noch nicht stattgefunden hat, so dass dieser nicht genutzt werden kann, um den L2-Schlüssel zu übertragen. Diese Reihenfolge ist damit begründet worden, dass ansonsten beim erstmaligen L4-Schlüsselaustausch ein Man-in-the-Middle Angriff durch einen Angreifer (Outsider) durchgeführt werden könnte, da die Datenintegrität auf Sicherungsschicht noch nicht gewährleistet werden kann. Somit entfällt ein vertrauliches Übertragen des L2-Schlüssels durch den Regelwerkknoten an den hinzuzufügenden Netzknoten.

Aus diesem Grund werden in diesem Unterabschnitt verschiedene Vorschläge *diskutiert*, um den L2-Schlüsselaustausch zu ermöglichen, ohne diesen für alle sichtbar zu übertragen. Im Folgenden wird die erste Nachricht, die der hinzuzufügende Netzknoten an seine Nachbarschaft versendet, *HELLO*-Nachricht genannt.

Reduzierung der Übertragungsreichweite

Viele IEEE 802.15.4 Funksensoren sind in der Lage, die Übertragungsreichweite zu reduzieren. Dadurch wird die Wahrscheinlichkeit, dass ein Angreifer einen böswilligen Netzknoten in der Nähe positioniert hat, minimiert. Dies hat den Nachteil, dass auch derjenige, der die HELLO-Nachricht empfängt und daraufhin den L2-Schlüssel überträgt, ebenfalls seine Sendeleistung reduzieren muss.

Solch ein Verfahren wird auch beim ZigBee Standard vorgesehen. Dort wird das Anmeldeverfahren der „Standard Security“ zugeordnet. [49, S. 67 f.]

Zusammenfassend wird diese Methode zum L2-Schlüsselaustausch nicht favorisiert, da dennoch beim unverschlüsselten Übertragen die Gefahr des Belauschens besteht.

Verwendung asymmetrischer Verfahren

Wenn der hinzuzufügende Netzknoten beim L4-Schlüsselaustausch asymmetrische Verfahren (wie RSA, RSA_PSK, DHE_PSK) einsetzen kann, dann kann geschlussfolgert

werden, dass diese auch für einen L2-Schlüsselaustausch zur Verfügung stehen. Doch diese Annahme stimmt nur bedingt.

Das Datagram Transport Layer Security Protokoll verwendet diese asymmetrischen Verfahren um ein Pre-Master-Secret auszutauschen, das dann in Kombination mit den zuvor ausgetauschten Zufallszahlen sowie einer Hash-Funktion zu einem symmetrischen Schlüssel umgewandelt wird. Danach wird mit diesem erzeugten Schlüssel die Finished-Nachricht verschlüsselt, die vor allem den Hash-Wert über alle bisher ausgetauschten Handshake-Nachrichten enthält. So wird der ausgetauschte Schlüssel nur akzeptiert, wenn zum einen der Schlüssel zur Entschlüsselung der Finished-Nachricht erfolgreich angewendet werden konnte und wenn der Hash-Wert mit dem eigenen übereinstimmt.

Dabei muss bedacht werden, dass lediglich ein netzweiter Schlüssel ausgetauscht wird, so dass ein fehlerhafter Schlüsselaustausch (ein böswilliger Nachbar schickt dem Netzknoten eine falsche Information) eher geringe Folgen haben wird und zudem durch Nachbarknoten entdeckt werden kann, wenn der MAC der Nachricht mit dem echten L2-Schlüssel untersucht wird.

Zur Demonstration wird ein L2-Schlüsselaustausch mit RSA durchgeführt: Derjenige Netzknoten, der ins Netz integriert werden möchte, besitzt einen privaten wie auch öffentlichen RSA-Schlüssel. Dieser könnte von ihm selbst erstellt worden sein oder wird vom Hersteller bereitgestellt und auf dem Netzknoten vorinstalliert. Der Netzknoten versendet mit der HELLO-Nachricht seinen öffentlichen Schlüssel mit.

Da man aufgrunder der Bedingung, dass nicht alle Netzknoten die gleichen Sicherheitstechnologien auf Transportschicht besitzen müssen, davon ausgehen kann, dass umliegende Netzknoten keine derartigen asymmetrischen kryptographischen Verfahren unterstützen, wird diese Nachricht – analog zum ZigBee Standard [49, S. 67 f.] – zum Regelwerkknoten weitergeleitet, der seinerseits dadurch Kenntnis vom Zutrittsversuch erlangt.

Nun übersendet er eine Antwort an den neuen Netzknoten. Diese enthält den L2-Schlüssel und wird mit dem öffentlichen RSA-Schlüssel des neuen Knotens verschlüsselt, so dass nur dieser Zugriff auf die Information hat.

Diese Methode hat zwei entscheidene Nachteile: (1) Einerseits muss es erlaubt sein, unauthentische Nachrichten im Sensornetz weiterzuleiten. Dadurch wird das Risiko einer Denial-of-Service Attacke erhöht, da Routingknoten gezielt überlastet werden können. (2) Andererseits müssen Netzknoten, die solche asymmetrischen Kryptographieverfahren nicht unterstützen, auf andere Weise in den Besitz des L2-Schlüssels gelangen können. Dies gilt im Speziellen für Routingknoten, die keine Hausautomatisierungsfunktionalität besitzen und so auch keinen L4-Schlüssel benötigen. Das Unterbringen einer RSA-Technologie wäre demnach unwirtschaftlich, wenn es alternative Methoden gibt. Besser ist es, eine Methode zu verwenden, die bei allen Netzknoten funktioniert. Doch diese wird es wohl nicht geben.

Verwendung eines physikalischen Out-of-Bound-Kanals

Die nächste Methode, die untersucht wird, ist die Nutzung eines physikalischen Out-of-Bound-Kanals. Diese hat den offensichtlichen Nachteil, dass ein zusätzlicher Kanal benötigt wird, inklusive der dafür nötigen Betriebssystemtreiber. Kommuniziert dieser Kanal zudem nicht drahtlos, geht eine gewisse vorteilhafte Eigenschaft des Sensornetzes verloren. Andererseits wird dadurch ein sicherer Austausch von Informationen ermöglicht.

Doch die Frage, die sich stellt, ist: Welcher Kompromiss wird eingegangen, um den L2-Schlüssel auf sichere Weise auszutauschen? Um diese Frage anzugehen, wird zuerst ein einfacher, physikalischer Out-of-Bound-Kanäle vorgestellt: Dieser ist eine serielle Schnittstelle wie USART oder USB. Insbesondere ist die USART-Implementierung bereits in den meisten eingebetteten Betriebssystemen wie Contiki-OS vorhanden. Ein USART-USB-Converter ermöglicht den Anschluss an einen USB-Port eines Hosts. Das Netzwerkprotokoll SLIP (engl. *Serial Line IP*, definiert in RFC 1055 [30]) ermöglicht eine Kommunikation über diese USART/USB-Schnittstelle. Sie wird beispielsweise im 6LoWPAN Border Routers in Contiki-OS benutzt, damit dieser mit einem am Internet angeschlossenen Host kommunizieren kann, um Pakete vom und ins Sensornetz zu routen.

Wann wird folglich diese physikalische Verbindung benötigt? Der erste Fall ist offensichtlich: Sie wird benötigt, wenn der zu hinzufügende Netzknoten keinen L2-Schlüssel und auch nicht den L4-Pre-Shared Key kennt. Sind diese beiden Geheimnisse auf sichere Weise ausgetauscht, dann ist ein erneutes Austauschen nicht mehr erforderlich.

Der netzweite Verbindungsschlüssel ist lediglich eine schwache Abwehrmaßnahme, um Angreifer (Outsider) vor der Manipulation von 802.15.4-Paketen abzuhalten. Daher ist es nicht erforderlich, diesen Schlüssel in regelmäßigen Abständen auszutauschen. Zudem müsste garantiert werden, dass alle Netzknoten zeitnah diesen Schlüssel erhalten. Wenn ein Netzknoten für einige Zeit vom Netz gekoppelt ist und sich erneut verbinden möchte, verbindet er sich im Glauben, dass der alte Schlüssel noch gültig ist. Bei paarweisen Schlüssel-Updates kann dies nicht passieren.

Ferner sind diejenigen Gefahren, von denen die größte Bedrohung ausgeht, auf der Anwendungsschicht zu finden. Deswegen ist es wichtiger, einen regelmäßigen L4-Schlüsselaustausch zum fortwährenden Schutz der Vertraulichkeit der Hausautomatisierungsnachrichten auszuführen.

Aber auch diese Methode hat ihre Nachteile: Wenn diese Methode ohne Zugriffssteuerung angewandt wird, kann ein Angreifer, der sich Zutritt zur Wohnung verschafft hat, Zugang zum netzweiten L2-Schlüssel sowie zu einem paarweisen L4-PSK erlangen.

5.5.3. Erstmaliger Anmeldevorgang (Zugriffskontrolle)

Eine Zugriffssteuerung ist erforderlich, damit Netzknoten sich nicht willkürlich in das Sensornetz einklinken können. Dabei reicht es nicht aus, zwischen Netzknoten zu unterscheiden, die den L2-SchlÜssel besitzen und die ihn nicht besitzen. Denn wie in der

zuletzt genannten Methode ersichtlich geworden ist, ist das Verbinden mit dem Sensornetz dennoch möglich. Nur eine weitere Zugriffskontrolle unterbindet dies.

Hierzu ist logischerweise eine Interaktion in der Managementebene, also vom Hausbewohner, erforderlich, um Netzknoten zur Automationsebene hinzuzufügen. Ähnlich dem Thingsquare System, ist es nötig, dass der Hausbewohner sich über ein Interface mittels Benutzername und Passwort (oder auf ähnliche Weise) anmeldet und dann dem Hausautomatisierungssystem mitteilt, welcher Netzknoten sich gerade registrieren möchte. Zudem muss für die meisten Netzknoten eh eine Konfiguration von Hausautomatisierungsregeln vorgenommen werden. Lediglich für Routingknoten fällt der Registrieraufwand höher aus.

Ferner ist es so möglich, eine differenziertere Zugriffssteuerung umzusetzen: Netzknoten, die erfolgreich registriert werden konnten und eine Routingfunktionalität mit sich bringen, erhalten zwei L2-Schlüssel. Der erste dient der integren und authentischen Kommunikation mit Netzknoten, die nicht routen können. Der zweite dient der integren und authentischen Kommunikation mit anderen Regelwerkknoten. Werden folglich Sensorknoten ohne Routingfunktionalität kompromittiert und der L2-Schlüssel ausgelesen, dann ist der Angreifer nicht in der Lage, Routing-Angriffe auszuführen.

Ein zusätzlicher Vorteil der Verwendung eines Out-of-Bound-Kanal ist ein kompletter Bootstrap: Es können nicht nur L2-Schlüssel und ein einzigartiger L4-PSK ausgetauscht werden, sondern auch weitere organisatorische Informationen wie

- die PAN-ID des 802.15.4-Netzes und
- die RPL-Instanz-ID, wenn RPL zum Einsatz kommt.

Damit läuft der komplette Anmeldevorgang, in Kürze dargestellt, wie folgt ab:

1. Der Hausbewohner nimmt den neuen Netzknoten X und verbindet ihn über den Out-of-Bound-Kanal (bspw. USART, USB) mit dem Regelwerkknoten;
2. dann meldet sich der Hausbewohner am Interface an und startet den Anmeldevorgang;
3. daraufhin werden wichtige Informationen sowie die Schlüssel ausgetauscht;
4. der Hausbewohner entkoppelt den Netzknoten und bringt ihn an der geplanten Stelle im Haus an, so dass er dann eine Verbindung mit dem drahtlosen Sensornetz herstellen kann;
5. bei Sensor- und Aktorknoten wird im nächsten Schritt der L4-Schlüssel mit einem DTLS-Handshake ausgetauscht, um zu testen, ob die Kommunikation auf Anwendungsebene funktioniert
6. bei Erfolg ist der Anmeldevorgang abgeschlossen.
7. In regelmäßigen Abständen wird der L4-Schlüssel durch einen DTLS-Re-Handshake aktualisiert.

Es ist sinnvoll, dass der Hausbewohner den Anmeldevorgang am Interface nachvollziehen kann. Das gilt sowohl für Sensor-/Aktorknoten wie auch für Routingknoten.

6 Evaluation der Sicherheitsarchitektur

Free software is simply software that respects our freedom — our freedom to learn and understand the software we are using.

—Free Software Foundation

Abschließend soll die Sicherheitsarchitektur, die im letzten Kapitel aufgestellt worden ist, anhand des Energieverbrauchs, des erhöhten Kommunikationsaufkommens durch das Datagram Transport Layer Security (DTLS) Protokoll sowie anhand einer Analyse des RAM-Verbrauchs evaluiert werden. In den einzelnen Abschnitten werden dabei Vorwie auch Nachteile besprochen. Zuerst soll aber ein Überblick geschaffen werden, welche Hard- und Software während der Ausarbeitung zum Einsatz gekommen ist.

6.1. Verwendete Hardware und Software

6.1.1. Mote-class Netzknoten

Als leistungsschwache Netzknoten kommen Ultra-Low-Power Mikrocontroller der Firma Atmel zum Einsatz. Dabei handelt es sich um einen AVR ATmega128rfa1, der den 802.15.4-2006/2003 kompatiblen Funksensor (2,4 GHz) bereits integriert hat (Single-Chip-Lösung) und für ZigBee-, Smart Metering- und IPv6- / 6LoWPAN-Funkapplikationen gezielt entwickelt worden ist. Dieser Mikrocontroller ist beispielsweise auf einem Dresden Elektronik deRFmega128 Board verbaut.

Für diese Evaluation sind folgende Eigenschaften interessant: CPU-Geschwindigkeit, verfügbarer Speicher und Stromverbrauch. Die ersten beiden sind in Tabelle 4.2 (siehe Seite 50) gezeigt worden. Einer CPU-Geschwindigkeit von bis zu 16 MHz ist ausreichend. Er besitzt durchschnittliche Speicherkapazitäten (128 KiB Flash und 16 KiB SRAM), wodurch er als mote-class Netzknoten für das Internet der Dinge in Frage kommt. Wenn dieser Netzknoten es schafft, so oft wie möglich in den Deep-Sleep Modus zu wechseln, kann sehr viel Strom eingespart werden. Nur die Differenz des Stromverbrauchs bei angeschaltetem Transceiver zwischen Senden (TX) und Empfangen (RX) fällt eher niedrig aus. Bei Nachfolgemodellen ist es möglich, diese Differenz zu steigern. Der Stromverbauch in unterschiedlichen Betriebsmodi ist in Abbildung 2.2 (siehe Seite 17) gezeigt worden.

Ferner ist es erwähnenswert, dass dieser Mikrocontroller eine in Hardware integrierte AES-128-Verschlüsselung wie auch einen Zufallszahlengenerator besitzt.

Das Betriebssystem, das zur Anwendung kommt, ist Contiki-OS (Version 2.6.x; Lizenz 3-clause-BSD). Es beinhaltet bereits einen kompletten 6LoWPAN-Netzwerkstack und speicherschonende Prozessunterstützung (Protothreads) in Verbindung mit einem kooperativem Scheduling [13].

6.1.2. Laptop-class Netzknoten

Die Auswahl an leistungsstarken Netzknoten ist sehr groß. Es könnte ein einfacher Server zum Einsatz kommen. Dieser besitzt allerdings einen hohen Stromverbrauch. Oft reichen auch energiesparende Mikroprozessoren wie die von ARM Limited.

Einer dieser ARM Mikroprozessoren ist der ARM1176JZF-S, der über $700\ \mathrm{MHz}$ verfügt. Er kommt beispielsweise auf einem Raspberry Pi Board, das von der Raspberry Pi Foundation entwickelt worden ist, zum Einsatz. In der Modell-Variante B bietet dieses Board $512\ \mathrm{MiB}$ SDRAM, zwei USB-Anschlüsse, einen SD-Kartenslot zum Einlesen des Betriebssystems sowie einen 10/100-MBit-Ethernet-Controller. Zusätzlich hat er mehrere GPIO-Pins (u. a. I2C, SPI, UART). Der Stromverbrauch beträgt maximal $3{,}5\ \mathrm{W}$.

Somit ist dieses Board geeignet, um als Regelwerkknoten im Netz zu fungieren. Lediglich eine 802.15.4-Funkschnittstelle muss nachgerüstet werden. Hierzu kann der obige AVR Mikrocontroller zum Einsatz kommen, indem er über UART und dem SLIP Protokoll an den Rasberry PI angeschlossen wird. Alternativ kann auch ein Funksensor direkt angeschlossen werden (bspw. CC2520).

Das Betriebssystem dieses Boards kann eins von zahlreichen Linux Distributionen sein. Auch Gentoo Linux (mit dem Raspberry Pi Kernel 3.6.11) kann auf diesem zur Anwendung kommen. Die Sicherheitsbibliothek CyaSSL (Version 2.8 oder höher; Lizenz GPLv2) kann – neben OpenSSL – die DTLS-Funktionalität für dieses Betriebssystem bereit stellen. Diese ist speziell für eingebettete Systeme entwickelt worden.

6.2. Kalkulierter Energiebedarf

In den Anforderungen zur Hausautomatisierung ist gesagt worden, dass die Energieeffizienz des Systems wichtig ist. Bei der Konzipierung der Sicherheitsarchitektur ist die Annahme aufgestellt worden, dass der Energieverbrauch des leistungsstarken Netzknotens im Verhältnis zu den anderen vernachlässigbar ist. In diesem Abschnitt gilt es daher diese Annahme zu überprüfen und den Energieverbrauch des gesamten Hausautomatisierungssystems zu kalkulieren.

Bei dieser Kalkulation wird einerseits zwischen mote-class Netzknoten unterschieden, die ein Duty-Cycling (siehe Unterabschnitt 2.2.5) verwenden, und solche, die es nicht verwenden. Weiterhin wird zur Vereinfachung der Kalkulation bei batteriebetriebenen Netzknoten angenommen, dass sie eine konstante Spannung von $3{,}3\ \mathrm{V}$ haben. Normalerweise sinkt die Spannung mit abnehmender Batteriekapazität, bis der AVR ATmega128rfa1 bei $1{,}8\ \mathrm{V}$ seinen Dienst einstellt. Ferner wird ebenfalls zur Vereinfachung angenommen, dass die mote-class Netzknoten mit Duty-Cycling beim Schlafen im Deep Sleep Modus ($250\ \mathrm{nA}$) verweilen und sich im aktiven Zustand im Transceiver TX Modus ($18{,}6\ \mathrm{mA}$)

Tabelle 6.1. *Kalkulierter Energiebedarf des Hausautomatisierungssystems: Vergleich von Regelwerkknoten und mote-class Netzknoten.*

mote-class Netzknoten		Energiebedarf der mote-class N.			Energiebedarf
ohne DC	mit DC	DC = 3 %	DC = 5 %	DC = 10 %	Regelwerkknoten
0	1	1,84 mW	3,07 mW	6,14 mW	3,5 W
0	10	18,42 mW	30,7 mW	61,43 mW	3,5 W
0	50	0,092 W	0,153 W	0,306 W	3,5 W
0	100	0,184 W	0,307 W	0,614 W	3,5 W
1	0		61,38 mW		3,5 W
10	0		0,614 W		3,5 W
50	0		3,069 W		3,5 W
100	0		6,138 W		3,5 W
1	1	63,22 mW	64,45 mW	67,52 mW	3,5 W
10	10	0,632 W	0,645 W	0,675 W	3,5 W
50	50	3,161 W	3,222 W	3,376 W	3,5 W
100	100	6,322 W	6,445 W	6,752 W	3,5 W

Regelwerkknoten und mote-class Netzknoten sind getrennt aufgeführt. Die mote-class Netzknoten werden danach unterschieden, ob sie Duty-Cycling (DC) verwenden oder nicht.

befinden. Solche ohne Duty-Cycling sind dauerhaft im Transceiver TX Modus. Daher wird folgende Rechnung durchgeführt:

$$\begin{aligned} P &= A \cdot V \\ P_{\text{sleep}} &= 250\,\text{nA} \cdot 3{,}3\,\text{V} = 0{,}825\,\mu\text{W} \\ P_{\text{TransTX}} &= 18{,}6\,\text{mA} \cdot 3{,}3\,\text{V} = 61{,}38\,\text{mW} \\ P_{\text{DC}}(d) &= d \cdot P_{\text{sleep}} + (1-d) \cdot P_{\text{TransTX}} \end{aligned}$$

Werden diese Zahlen mit der Anzahl an mote-class Netzknoten multipliziert, so erhält man den kalkulierten Energieverbrauch der mote-class Netzknoten. Der Energiebedarf des Regelwerkknotens ist konstanz angenommen und zum Vergleich in Tabelle 6.1 mit aufgeführt.

In dieser Tabelle – beim Vergleich von 100 Netzknoten mit Duty-Cycling und 100 ohne – ist zu erkennen, dass das Duty-Cycling einen enorme Auswirkung auf den Gesamtenergiebedarf des Hausautomatisierungssystems hat: 100 mote-class Netzknoten ohne Duty-Cycling verbrauchen fast doppelt so viel Energie wie der Regelwerkknoten.

Insgesamt aber – bei Betrachtung der letzten Zeile – fällt der Gesamtverbrauch bei Verwendung von 200 mote-class Netzknoten und einem laptop-class Netzknoten mit etwa 10 W sehr niedrig aus, wenn man bedenkt, dass das Hausautomatisierungssystem einerseits den Komfort des Hausbewohners erhöht und andererseits erlaubt, gegenwärtige mit zukünftigen Sicherheitsstandards zu kombinieren.

Tabelle 6.2. *Von DTLS verursachtes Kommunikationsaufkommen.*

DTLS Nachricht		Schlüsselaustausch (in B)				
		PSK	DHE_PSK	RSA[1]	RSA_PSK	DHE_RSA[1]
$C \longrightarrow S$	**Client Hello**	67 B	67 B	79 B	67 B	79 B
$C \longleftarrow S$	Hello Verify Request			48 B		
$C \longrightarrow S$	Client Hello (cookie)	87 B	87 B	99 B	87 B	99 B
$C \longleftarrow S$	**Server Hello**			95 B		
$C \longleftarrow S$	Certificate	–	–	902 B	902 B	902 B
$C \longleftarrow S$	Server Key Exchange	40 B	815 B	–	40 B	1188 B
$C \longleftarrow S$	Certificate Request	–	–	–	–	–
$C \longleftarrow S$	**Server Hello Done**			25 B		
$C \longrightarrow S$	Certificate	–	–	–	–	–
$C \longrightarrow S$	**Client Key Exchange**	42 B	428 B	411 B	428 B	411 B
$C \longrightarrow S$	Certificate Verify	–	–	–	–	–
$C \longrightarrow S$	**[Change Chipher Spec]**			14 B		
$C \longrightarrow S$	**Finished**[2]			53 B		
$C \longleftarrow S$	**[Change Chipher Spec]**			14 B		
$C \longleftarrow S$	**Finished**[2]			53 B		
Handshake gesamt		538 B	1699 B	1793 B	1826 B	2981 B
$C \longleftrightarrow S$	AppData Overhead[2]			29 B		

Gewählte Parameter: RSA-Certificate x509, RSA 3072bit Schlüssel, DHE 3072bit Schlüssel, AES 128bit Schlüssel, AES_128_CCM_8, PSK ID mit 15 B, PSK ID Hint mit 13 B, Cookie mit 20 B, Session ID mit 32 B, Cipher Liste enthält nur ein Cipher und Nullkompression. RSA Schlüssel und Certificate sowie DHE Parameter wurden mit OpenSSL erzeugt.

[a] Client Hello mit Extensions.
[b] Verschlüsselte Daten sind abhängig vom Verschlüsselungsalgorithmus.

6.3. Kommunikationsaufkommen durch DTLS

Die nächste zu untersuchende Eigenschaft der konzipierten Sicherheitsarchitektur ist die Verwendung des Datagram Transport Layer Security (DTLS) Protokolls, um eine Cipher Suite sowie den symmetrischen Schlüssel zwischen Regelwerkknoten und einem beliebigen anderen Netzknoten auszuhandeln. Da dieses Protokoll ein zusätzliches Kommunikationsaufkommen verursacht, wird in diesem Abschnitt dieses evaluiert.

In Tabelle 6.2 ist das durch DTLS verursachte Kommunikationsaufkommen für verschiedene Schlüsselaustauschverfahren (basierend auf RSA, PSK und DHE) aufgeführt, aufgeschlüsselt nach den verschiedenen DTLS Nachrichten (siehe auch Abbildung 4.5). In dieser Tabelle ist zu erkennen, dass folgende Nachrichten …

- Hello Verify Request,
- Server Hello und Server Hello Done sowie

Tabelle 6.3. *Vergleichbare Schlüsselstärken verschiedener Kryptographieverfahren. [31]*

Sicherheitsstufe	Sym. Verschl.	Diffie-Hellman	RSA	ECC	Zeitspanne
80 bit	2TDEA	1024 bit	1024 bit	160 bis 223 bit	$\leq$ 2010
112 bit	3TDEA	2048 bit	2048 bit	224 bis 255 bit	$\leq$ 2030
128 bit	AES-128	3072 bit	3072 bit	256 bis 383 bit	2031+
192 bit	AES-192	7680 bit	7680 bit	384 bis 511 bit	2031+
256 bit	AES-256	15 360 bit	15 360 bit	512 bit	2031+

Diffie-Hellman zählt zur Finite-Field Cryptography (FFC); RSA zählt zur Integer-Factorizating Cryptography (IFC).

- Change Cipher Spec und Finished

nicht von den Schlüsselaustauschverfahren abhängig sind. Auch kann man der Tabelle entnehmen, dass folgende Nachrichten …

- (Server) Certificate,
- Server Key Exchange und
- Client Key Exchange

am meisten Kommunikationsaufkommen erzeugen. Das lässt sich nicht vermeiden. Doch kann man dies reduzieren, wenn die Sicherheitsstufe (d. h. die Schlüssellänge) reduziert wird. Vergleichbare Schlüsselstärken und ihre geschätzte Gültigkeitsdauer können der Tabelle 6.3 entnommen werden, wobei die Gültigkeitsdauer (Zeitspanne) besagt, wie lange eine Sicherheitsstufe als sicher anzunehmen ist. Somit ist es erforderlich, dass beim Diffie-Hellman- oder RSA-Schlüsselaustausch eine Schlüssellänge von mindestens 2048 bit eingesetzt wird.

Weiterhin bietet das Schlüsselaustauschverfahren PSK, das lediglich Identitäten austauscht, das mit Abstand geringste Kommunikationsaufkommen beim DTLS Handshake. Doch wie bereits erwähnt worden ist, bietet die alleinige Verwendung von Pre-Shared Key kein Perfect Forward Secrecy (PFS) und die Geheimnisse können nicht so einfach über den Out-of-Bound-Kanal ersetzt werden.

Zudem erkennt man auch sehr gut, dass DHE_RSA ein fast doppelt so hohes Kommunikationsaufkommen besitzt wie RSA, DHE_PSK oder RSA_PSK. Das liegt daran, dass sowohl Diffie-Hellman-Parameter wie auch der RSA-Schlüssel ausgetauscht werden. Folglich ist dieses Verfahren nachweislich für mote-class Netzknoten ungeeignet.

Folglich wird die Verwendung eines der drei übrig gebliebenen Schlüsselaustauschverfahren empfohlen. Es muss allerdings bedacht werden, dass RSA_PSK nicht in Zusammenhang mit AES-CCM verwendet werden kann [26].

6.4. Analyse des RAM-Verbrauchs asymmetrischer Kryptographie

In den Voraussetzungen an die Hausautomatisierung ist gesagt worden, dass mote-class Netzknoten nur über wenig Speicher verfügen. Dieses hat sich während der Diskussion zur sicheren Kommunikation im Internet der Dinge im Unterabschnitt 4.3.3 zur Schlüsselverteilung und zum Schlüsselmanagement bestätigt. Da Schlüssel austauschbar sein sollen (und der EEPROM bei AVRs i. A. zu klein ist), muss er im RAM (SRAM bei AVRs) zwischengelagert werden. Doch müssen nicht nur Schlüssel (Ergebnisse der Berechnungen) gespeichert werden, sondern auch Zwischenergebnisse der Berechnungen. In diesem Abschnitt geht es folglich um die Bestimmung des RAM-Bedarfs, um diese Zwischenergebnisse speichern zu können.

Aufgrund der Tatsache, dass das Hausautomatisierungssystem im Hintergrund des Alltags arbeitet, ist es nebensächlich, wie lange die Berechnungen dauern. Ein paar Minuten sind akzeptabel. Daher wird darauf verzichtet, eine Laufzeitanalyse durchzuführen. Diese ist erst lohnend, wenn der RAM-Verbrauch optimiert worden ist.

6.4.1. Multi-Precision Math

Da mit sehr großen Zahlen (Big Integers), die weit mehr als $64\ \mathrm{bit}$ umfassen, gerechnet werden muss, kommt eine so genannte Multi-Precision Math zum Einsatz. Diese ermöglicht es, Rechenoperationen auf sehr große Zahlen auszuführen, ohne an Genauigkeit zu verlieren. Diese Operationen benötigen i. A. viel Rechenaufwand, so dass bisher die allgemeine Meinung besteht, dass asymmetrische Verfahren aufgrund dieser Berechnungen auf Mikrocontrollern nicht umsetzbar sind. Dabei wurde vor allem auf den zeitlichen Aspekt eingegangen. [vgl. 50, 51]

Dem entgegenstehend sind den asymmetrischen ECC Cipher Suites eine höhere Bedeutung aufgrund ihrer geringeren Schlüssellänge zugeschrieben worden (siehe Tabelle 6.3). Aufgrund aktueller Vorfälle aufgrund des NSA Skandals wird einigen ECC Cipher Suites nicht mehr getraut. Allerdings ist auch bereits wieder eine Teilentwarnung gegeben worden [66].

Nichtsdestoweniger ist der RAM-Verbrauch nicht direkt auf die einzelnen Cipher Suites bestimmt worden, sondern auf die Ausführung der Multi-Precision Math. Einerseits sollen so zukünftige Ergebnisse einfacher miteinander verglichen werden können. Andererseits benötigt die mathematische Berechnung des Ergebnisses von allen ausgeführten Funktionen am meisten Zeit. Bei diesen Berechnungen wird ebenfalls RAM benötigt, da Zwischenergebnisse gespeichert werden müssen.

Daher bezieht sich die Analyse des RAM-Verbrauchs asymmetrischer Kryptographie allgemein auf die Messung des RAM-Verbrauchs während der Ausführung der mathematischen Berechnungen. Diese sind unabhängig von der jeweiligen Architektur des Mikroprozessors oder Mikrocontrollers vergleichbar.

6.4.2. Bibliothek zur Multi-Precision Math

Damit eine Analyse ausgeführt werden kann, galt es in erster Linie vorhandene Lösungen zu suchen und in Contiki-OS zu integrieren.

GMP Die Bibliothek *GMP* (Lizenz LGPL) ist eine der bekanntesten, die eine solche Multi-Precision Math ermöglicht. Doch ist sie weder für eingebettete Systeme noch für mote-class Netzknoten entwickelt worden. Es wird derzeit an einer minimalistischen Variante von GMP gearbeitet, Mini-GMP. Diese ist allerdings noch nicht einsatzfähig.

Libtommath Die Bibliothek *Libtommath* (Lizenz GPLv2 oder WTFPL) ist im Gegensatz zu GMP in portablem C geschrieben worden, so dass nur die benötigten Funktionen auf dem eingebetteten System integriert werden können. Sie wird von der CyaSSL Bibliothek verwendet.

Bei der Kompilation unter AVR gab es allerdings einen schweren Compiler-Fehler, so dass diese Bibliothek nicht zum Einsatz kam. Die Kompilation mit nativer CPU (x86/x86_64) dagegen hat in Contiki-OS einwandfrei funktioniert.

AVR-Crypto-Lib Zuletzt wurde die AVR-Crypto-Lib (das-labor.org, Lizenz GPLv3) betrachtet [67]. Die veröffentlichte Version ist als veraltet gekennzeichnet. Die BigInt Funktionalität arbeitet mit der dynamischen Speicherverwaltung malloc().

Nach Betrachtung aller drei Bibliotheken ist eine eigene Variante umgesetzt worden, die sich an der BigInt Funktionalität der AVR-Crypto-Lib orientiert hat. Performance (Assemblerumsetzungen) standen dabei nicht im Vordergrund, sondern eine portable Programmierung, so dass sie auf zukünftigen Mikrocontrollern (zu Testzwecken) eingesetzt werden kann. Kennzeichnend für diese Umsetzung ist die Verwendung von Arrays, die nicht mittels malloc() alloziiert werden, so dass der Anwender die Möglichkeit geboten wird, seinen Speicherbereich und Speicherverbrauch besser zu bestimmen. So muss der Aufrufer ein Big Integer übergeben, welcher zur Berechnung des Ergebnisses genügend Platz bietet. Dennoch werden noch temporäre Big Integers benötigt, um Zwischenergebnisse zu speichern.

6.4.3. Berechnung

Zur Berechnung einer RSA-Verschlüsselung und -entschlüsselung (die RSA-Schlüsselerzeugung ist nicht miteinbezogen) sowie bei der Berechnung des gemeinsamen Diffie-Hellman-Geheimnisses kommt eine modulare Exponentiation (expmod) zum Einsatz:

$$R = \text{expmod}(B, E, N) = B^E \mod N$$

Damit werden drei Big Interger übergeben. Da die nacheinander ausgeführte Berechnung (erst exp, dann mod) sowohl im Speicherverbrauch wie auch zeitlich zu aufwändig ist, gibt es einige Verfahren, die diese Berechnung unter bestimmten Bedingungen beschleunigen. Dieser Unterschied ist i. A. um so größer, je größer die übergebenen Parameter sind.

Diffie-Hellman

Die Voraussetzung für einen Diffie-Hellman (DH) Schlüsselaustausch ist es, dass der DTLS Server die öffentlichen Parameter P (eine Primzahl der Länge des DH-Schlüssels) und g (ein Generator, meist 2) vorliegen hat. DTLS Server wie auch Client wählen eine beliebige Zahl der Länge des DH-Schlüssels, die der privaten DH-Schlüssel X ist. Der zugehörige öffentliche DH-Schlüssel Y wird wie folgt berechnet:

$$Y = g^X \mod P$$

Nun erfolgt der Schlüsselaustausch, wobei der Client Y_S und der Server Y_C erhält. Zur Berechnung des gemeinsamen Diffie-Hellman-Geheimnisses wird von beiden folgende Gleichung angewandt:

$$\begin{aligned} Z_{S,C} &= \exp(Y_C, X_S) \mod P \quad \text{(Server)} \\ Z_{S,C} &= \exp(Y_S, X_C) \mod P \quad \text{(Client)} \end{aligned}$$

Somit muss zweimal eine expmod-Berechnung durchgeführt werden, um das gemeinsame Geheimnis zu berechnen. Dabei werden folgende Eingabelängen (in bit) verwendet:

- expmod(g, X, P): $g = 2$ ist $2\,\mathrm{bit}$ lang. X und P haben die Länge k. Bei DH-2048 ist $k = 2048\,\mathrm{bit}$.
- expmod(Y, X, P): Alle drei Parameter haben die Länge k.

RSA-Verschlüsselung und -Entschlüsselung

Wird eine Nachricht M RSA-verschlüsselt, kommt das öffentliche RSA-Schlüsselpaar (E, N) des Empfängers zum Einsatz:

$$C = \exp(M, E) \mod N$$

wobei die Länge von M kleiner ist als die Länge von N. Der Empfänger entschlüsselt diese kryptische Nachricht C mit seinem privaten RSA-Schlüsselpaar (D, N):

$$M = \exp(C, D) \mod N$$

wobei die Länge von C kleiner oder gleich der Länge von N ist. Der RSA-Modul N gibt die Stärke von RSA an und hat damit die Länge k. Der private Schlüssel benötigt eine gewisse Länge, besitzt aber eine kleinere Länge wie k. Da der öffentliche Schlüssel bekannt ist, wird er i. d. R. klein gewählt (bspw. $E = 2^{16} + 1$).

Beim **RSA_PSK** Schlüsselaustausch des DTLS Handshakes verschlüsselt der Client den Pre-Master-Secret mit dem öffentlichen RSA-Schlüsselpaar des Servers. Dieses Pre-Master-Secret besteht aus folgenden Werten:

- einer Längenangabe in Byte ($2\,\mathrm{B}$ lang),
- der verwendeten DTLS Versionsnummer ($2\,\mathrm{B}$ lang),
- einer Zufallszahlenfolge ($46\,\mathrm{B}$ lang),
- der Länge des Pre-Shared Keys ($2\,\mathrm{B}$ lang) und

- dem Pre-Shared Key (variable Länge)

Zusammengefasst wird bei Verwendung eines $32\ \mathrm{B}$ langen PSK eine Nachricht M mit einer Länge von $672\ \mathrm{bit}$ ($84\ \mathrm{B}$) verschlüsselt versendet. Je nach Länge von M werden folglich eine oder mehrere expmod-Berechnungen durchgeführt. Dabei werden folgende Eingabelängen (in bit) verwendet:

- expmod(M_i, E_S, N_S): N_S hat eine Länge von k ($k \geq 2048\ \mathrm{bit}$). M_i ist kleiner oder gleich k. Bei Verwendung von $E = 2^{16} + 1$ hat E eine Länge von $17\ \mathrm{bit}$.

Der Berechnungsaufwand dieses Verfahrens scheint damit niedriger als der Diffie-Hellman-Schlüsselaustausch zu sein. Doch kann der RSA-Schlüssel kompromittiert werden, dann ist der PSK dem Angreifer bekannt.

Kommt nicht RSA_PSK, sondern **RSA** als Schlüsselaustauschverfahren zum Einsatz, dann entfällt der PSK-Teil aus dem Pre-Master-Secret und die Länge von M ist kleiner. Eine RSA-Signatur wird nicht erzeugt. Wenn jedoch das Zertifikat des DTLS Servers nicht bei einer Authentisierungsinstanz kontrolliert wird, kann ein Angreifer einen Man-in-the-Middle Angriff ausführen. Bisher ist keine Zertifizierungsinstanz im Hausautomatisierungssystem eingeplant, da der Diffie-Hellman-Schlüsselaustausch in Zusammenhang mit einem PSK aufgrund der Perfect Forward Secrecy Eigenschaft vorgezogen wird.

6.4.4. Implementationsaspekte

Es gibt viele Verfahren, die eingesetzt werden können, um den Berechnungsaufwand zu minimieren. Für die Berechnung der expmod-Funktion kam dabei ein Verfahren zum Einsatz, das lediglich $\mathcal{O}(\log |E|)$ Multiplikationen und Reduktionen benötigt [52], wobei $|E|$ die Länge des Exponenten E ist.

Weiterhin wird ein Verfahren benötigt, das eine Reduktion beschleunigt. An dieser Stelle wurde speicherschonend gearbeitet und eine Reduktion auf Basis von Shift- und Subtraktionsoperatoren gewählt, da Shift- und Additions- wie auch Subtraktionsoperationen im Gegensatz zur Multiplikation oder Division schnell verarbeitet sind. Es gibt auch andere Verfahren, die eine Reduktion durch Multiplikationen ersetzen (z. B. Barrett Reduction) oder auf andere Weise lösen (z. B. Montgomery Reduction).

Zuletzt wird ein schnelles Verfahren für die Multiplikation benötigt. Einer dieser Verfahren ist der Karatsuba Multiply Algorithm mit einer Laufzeitkomplexität von $\mathcal{O}(|N|^{\log_2(3)})$, wobei $|N|$ die Länge des längsten Eingabeparameters ist. Er teilt nach dem Teile-und-Herrsche-Prinzip die beiden Eingabeparameter x und y mittig und reduziert so den Berechnungsaufwand, indem mehrere Teilberechnungen durchgeführt werden.

6.4.5. Analyse der expmod-Funktion

Da nun die Berechnungsgrundlagen (und die Größe der verwendeten Eingabeparameter) sowie die verwendeten mathematischen Verfahren vorgestellt worden sind, erfolgt nun die Analyse des RAM-Verbrauchs bei unterschiedlichen Eingabe-Parametern. Dabei werden zuerst alle Parameter klein gewählt. Dann wird immer nur ein Parameter verändert und das Resultat betrachtet.

Tabelle 6.4. *Temporärer RAM-Verbrauch bei unterschiedlichen Eingabeparametern für die modulare Exponentiation expmod*$(B, E, N) = B^E \mod N$.

Länge der Eingabeparameter			RAM	Bemerkungen
Basis B	Exponent E	Modul N		
63 bit	63 bit	37 bit	352 B	
63 bit	160 bit	37 bit	352 B	nur die Berechnungszeit erhöht sich!
63 bit	320 bit	37 bit	352 B	
63 bit	640 bit	37 bit	352 B	
63 bit	2048 bit	37 bit	352 B	
63 bit	3072 bit	37 bit	352 B	
80 bit	63 bit	37 bit	360 B	linearer Anstieg mit Steigung 0.5
160 bit	63 bit	37 bit	400 B	
240 bit	63 bit	37 bit	440 B	
320 bit	63 bit	37 bit	480 B	
640 bit	63 bit	37 bit	640 B	
1024 bit	63 bit	37 bit	832 B	
2048 bit	63 bit	37 bit	1344 B	
3072 bit	63 bit	37 bit	1856 B	
63 bit	63 bit	80 bit	594 B	nicht-linearer Anstieg mit Steigungen über 1.5
63 bit	63 bit	160 bit	914 B	Steigung 4 zum Vorgänger
63 bit	63 bit	240 bit	1240 B	Steigung 4.1 zum Vorgänger
63 bit	63 bit	320 bit	1354 B	Steigung 1.4 zum Vorgänger
63 bit	63 bit	1024 bit	2816 B	Steigung 2.08 zum Vorgänger
63 bit	63 bit	2048 bit	4552 B	Steigung 1.7 zum Vorgänger
63 bit	63 bit	3072 bit	6082 B	Steigung 1.5 zum Vorgänger
1024 bit	63 bit	1024 bit	2816 B	$\|B\| \leq \|N\| \Longrightarrow \|B\|$ vernachlässigbar!
2048 bit	63 bit	2048 bit	4552 B	
3072 bit	63 bit	3072 bit	6082 B	

In Tabelle 6.4 sind die Ergebnisse aufgelistet. Bei den Messungen ist festgestellt worden, dass das Erhöhen der Länge des Exponenten keine Auswirkung auf den temporären RAM-Verbrauch hat, lediglich die Berechnungszeit erhöht sich stark. Das Erhöhen der Länge der Basis sowie des Moduls wirkt sich dagegen negativ auf den Verbrauch aus. Kleine Module verursachen dabei noch keine Probleme. Doch je größer der Modul wird, desto mehr RAM wird zur Berechnung benötigt.

Der Modul von DH wie auch RSA besitzt eine Länge von k und stellt den größten Faktor dar. Das bedeutet folglich, dass sowohl für RSA wie auch für DH der gleiche temporäre RAM-Verbrauch gilt. Somit unterscheiden sich die beiden Verfahren insbesondere an der Länge des Exponenten, so dass RSA schneller berechnet werden kann. Dies nützt allerdings nichts, wenn beide Verfahren einen zu großen RAM-Bedarf mit sich bringen, so dass an dieser Stelle noch optimiert werden muss.

7 Schlussbemerkungen

Free software is designed to free the user from restrictions put in place by proprietary software, and so using free software lets you join a global community of people who are making the political and ethical choice assertion of our rights to learn and to share what we learn with others.

—Free Software Foundation

Diese Arbeit gibt abschließend einen kurzen Ausblick auf noch zu erledigende Aufgaben, die aus dieser Arbeit resultieren. Zuletzt wird der gesamte Themenkomplex zusammengefasst und ein Fazit gezogen.

7.1. Ausblick

Da das Thema dieser Masterarbeit sehr umfangreich ist, gibt es eine ganze Reihe an Fragestellungen, die beantwortet werden wollen. Daher soll diese Masterarbeit den Grundstein für weitere Arbeiten legen.

Die wichtigste Aufgabe wird es sein, den Arbeitsspeicherverbrauch der Big Integer Funktionen (Multi-Precision Math) weiter zu reduzieren. Dabei sollten die einzelnen mathematischen Teilfunktionen genauer untersucht und ein Vergleich zwischen verschiedenen Verfahren gezogen werden.

Weiterhin ist die Implementierung der DTLS Funktionalität in Contiki-OS noch nicht vollständig und muss noch ergänzt werden. Dabei sollte auf das Verwalten der Schlüssel Rücksicht genommen werden. Eine Analyse verschiedener Speichermöglichkeiten für die beim Schlüsselaustausch verwendeten asymmetrischen Schlüssel ist ein interessanter Ansatzpunkt.

Die Alternative zur Schlüsselverwaltung ist das Verwenden eines separaten Mikrocontrollers, der nur für den Schlüsselaustausch erwacht und sonst die restliche Zeit im Deep Sleep Modus verbringt. In diesem Zusammenhang ist der Energieverbrauch des Netzknotens interessant.

Auch ist gesagt worden, dass das Schlüsselaustauschverfahren RSA_PSK in Frage kommt. Eine Implementation dieses Schlüsselaustauschsverfahrens in Contiki-OS ist demnach eine weiterer Ansatz, um die Sicherheitsarchitektur zu evaluieren. Dabei muss

bedacht werden, dass RSA_PSK nicht in Verbindung mit AES-CCM eingesetzt werden kann. Daher sollte AES-GCM zur Anwendung kommen.

7.2. Zusammenfassung

Das Marktsegment der drahtlosen, internetfähigen und offenen Hausautomatisierung bietet vielfältige Anwendungsmöglichkeiten. Insbesondere in Kombination mit dem Internet der Dinge gibt es noch unvorstellbare Szenarien. Thingsquare versucht beispielsweise dieses Gebiet von einer anderen Seite zu beleuchten, indem Dienste auf leistungsschwachen Netzknoten einer Serveranwendung zur Verfügung stehen (und umgekehrt).

Mit dieser Arbeit steht ein Konzept einer Sicherheitsarchitektur für eine auf dedizierter Regelverarbeitung basierende Hausautomatisierung zur Verfügung. Dieses Konzept gilt es noch in einer kompletten Hausautomatisierungsanwendung zu evaluieren. Fhem ist ein Regelwerksystem, das zur Anwendung kommen kann.

Wichtig in der konzipierten Sicherheitsarchitektur ist die Gewährleistung des Schutzes der allgemeinen Persönlichkeitsrechte. Diese wurden auf Anwendungsebene umgesetzt, indem das Datagram Transport Layer Security (DTLS) Verschlüsselungsprotokoll eingesetzt wird. Dieses vollzieht in regelmäßigen Abständen ein Schlüsselaustausch zur Gewährleistung der Vertraulichkeit der Hausautomatisierungsdaten. Dabei ist es sinnvoll, Schlüsselaustauschverfahren anzuwenden, die ein Perfect Forward Secrecy (PFS) ermöglichen. Ein eher schwacher Schutz sichert die Integrität auf Routingebene. Ein stärkerer Schutz ist aber auf dieser Ebene nicht erforderlich, da der eigentliche Schutz auf Anwendungsebene umgesetzt ist und so vor den schlimmsten Gefahren schützen soll.

Ein Out-of-Bound-Kanal in Kombination mit einer Anmeldung am Interface sorgt für die Zugriffskontrolle. Das bedeutet, dass der Hausbewohner lediglich einen neu hinzuzufügenden Netzknoten per USART oder USB (oder vergleichbare Technologien) an den Regelwerkknoten anschließt und dann nur noch im Interface auf „Netzknoten anmelden“ drückt. Alles weitere (mit Ausnahme der Konfiguration der neuen Regeln) wird automatisch vollzogen und der Hausbewohner hat so wenig wie möglich mit der Sicherheitsarchitektur zu tun. Dadurch wird der Endverbraucher so wenig wie möglich in den Prozess, der die Sicherheit der Hausautomatisierungsdaten gewährleistet, integriert. Sollte einmal die Gefahr bestehen, dass ein Netzknoten kompromittiert worden ist, dann schließt der Hausbewohner ihn einfach wieder am Regelwerkknoten an und führt einen zweiten Anmeldeprozess durch, wodurch alle wichtigen Geheimnisse ausgetauscht werden.

Beispielszenario — „der heimische Arbeitsraum“

A

In der heutigen Zeit kommt es häufiger vor, dass Menschen von Zuhause aus arbeiten. Deshalb wird der heimische Arbeitsraum immer wichtiger. Im Folgenden werden drei Beispiele genannt, wie die Unterstützung des Hausbewohners erfolgen kann.

Studien sowie Erfahrungen haben gezeigt, dass sich das Wohlfühlen am Arbeitsplatz positiv auf die Effizienz des Arbeitenden auswirkt [68]. Doch im häuslichen Umfeld kann der Arbeitsplan individuell eingerichtet werden. Ein Hausautomatisierungssystem für den heimischen Arbeitsraum kann folglich die Aufgabe haben, ein angenehmes Arbeitsklima zu erzeugen, wodurch der Arbeitende eine effizientere Leistung erbringen kann. Denn diese ist zuhause genauso wichtig wie am Firmenarbeitsplatz.

Auch bei der Beaufsichtigung der Kinder bietet das Hausautomatisierungssystem am heimischen Arbeitsplatz eine Komforterhöhung und kann Eltern unterstützen. Ein Babyphone ist beispielsweise eine bereits vorhandene und genutzte Technik und könnte auch in ähnlicher Form in ein Hausautomatisierungssystem integriert werden.

Der Umgang mit vertraulichen Informationen ist zuhause wie auch in Bürokomplexen ein großes Problem. Es ist erforderlich, dass diese Informationen so gut wie möglich geschützt werden. Das Hausautomatisierungssystem bietet auch an dieser Stelle eine (eingeschränkte) Unterstützung. Als Überwachungssystem ist es in der Lage, Einbrüche zu erkennen und zu melden.

Zur Durchführung einer Gefahrenanalyse werden nun zwei Steuerungen betrachtet, die durchaus typische Anwendungen am heimischen Arbeitsplatz darstellen: Die Klimaanlagensteuerung und die Einbruchskontrollsteuerung. Zur Beschreibung dieser zwei Steuerungen werden ihre Sensoren und Aktoren vorgestellt und anschließend ein Überblick zur Kommunikation gegeben.

A.1. Klimaanlagensteuerung

Aufgrund der Ausübung der beruflichen Tätigkeiten ist es notwendig, die Raumtemperatur im Arbeitsraum so zu regeln, dass ein angenehmes Arbeitsklima entsteht. Die Wohlfühltemperatur liegt normalerweise bei 22 °C. Wenn der Arbeitsraum allerdings nicht genutzt wird, dann muss er nicht auf die Wohlfühltemperatur geregelt werden.

Es reicht dann aus, wenn die Temperatur im Raum an kalten Tagen bei 17 °C und an warmen Tagen über 22 °C liegt. Eine intelligente Steuerung spart dadurch Energie und bewirkt folglich eine finanzielle Einsparung.

Damit das Hausautomatisierungssystem die Klimaanlagensteuerung sinnvoll umsetzen kann, ist der Einsatz folgender Sensoren und Aktoren denkbar[1]:

Temperatursensor Temperatursensoren dienen der Temperaturmessung und sind im Raum verteilt angebracht. Eine Aggregation der Temperaturwerte führt zur Raumtemperatur.

Kontaktsensor für Tür oder Fenster Kontaktsensoren prüfen, ob der Kontakt geschlossen oder offen ist. Demnächst prüft ein Kontaktsensor für ein Fenster den Zustand der Schließvorrichtung von diesem. So muss nicht erst abgewartet werden, bis sich die Raumtemperatur ändert, um schlussfolgern zu können, dass ein Fenster oder eine Tür geöffnet ist.

Bei Fenstern oder Außentüren ist es sinnvoll, eine *Innen- und Außentemperaturmessung* vorzunehmen, damit das System den Unterschied zwischen diesen beiden Temperaturen kennt und so einschätzen kann, ob der Handlungsbedarf dringend ist oder nicht.

Bewegungsmelder Bewegungsmelder zeigen eine Bewegung im Arbeitsraum auf, wodurch erkannt werden kann, ob sich jemand im Raum aufhält oder nicht. Wenn längere Zeit keine Bewegung detektiert wird, dann kennzeichnet dies eine aktuelle Nicht-Benutzung des Raumes und der Raum muss nicht mehr auf die Wohlfühltemperatur geregelt werden.

Heizungsregler Der Heizungsregler ist für die Ansteuerung der Heizung zuständig und regelt damit die Beheizung des Raumes.

Eine Möglichkeit ist die Einstellung des Heizungsventils, also wie weit das Ventil geöffnet ist. Die andere Möglichkeit ist ein Vergleich zwischen Ist- und Soll-Temperatur. Diese müssen allerdings in regelmäßigen Abständen übertragen werden, damit der Heizungsregler den Raum entsprechen beheizen kann.

Eine durchaus geschicktere Variante ist die Verwendung eines Heizungsreglers in Kombination mit einem *Temperatursensor*. So muss die Steuerung nicht regelmäßig Ist- und Soll-Temperatur übertragen. Der Aktor führt stattdessen eine Kalibration des Temperaturwertes durch. Durch Übertragung der Ist-Temperatur wird ein Offset zur gemessenen Temperatur gesetzt und bei allen weiteren Temperaturmessungen entsprechend berücksichtigt und dann mit der Soll-Temperatur verglichen.

Kühlvorrichtung An zu heißen Tagen liegt die Raumtemperatur i. A. über der Wohlfühltemperatur. Ein Ventilator könnte beispielsweise als eine einfache Kühlvorrichtung

[1] Es sei angemerkt, dass es weitere Lösungsmöglichkeiten gibt. An dieser Stelle wird auf eine umfangreiche Begründung der Auswahl verzichtet.

dienen. Eine Alternative ist eine Klimaanlage, die analog zum Heizungsregler eine Ist- und Soll-Temperatur benötigt.

Der Regelwerkknoten des Hausautomatisierungssystems erhält von den Temperatursensoren die Temperaturwerte und führt eine Aggregation dieser Werte aus, um die aktuelle Raumtemperatur (Ist-Temperatur) zu berechnen. Aufgrund der Verwendung von Fensterkontaktsensoren können unerwartete Temperaturschwankungen i. A. nicht vorkommen, da die Fensterkontaktsensoren dem Regelwerkknoten Bescheid geben, sobald Fenster geöffnet bzw. geschlossen werden, insbesondere wenn Innen- und Außentemperatur stark voneinander abweichen.

Sofern sich der Arbeitende im Raum befindet, erkennt der Bewegungsmelder seine Bewegungen. Aufgrund dieser Bewegungserkennung weiß das Hausautomatisierungssystem, dass die Raumtemperatur auf Wohlfühltemperatur geregelt werden muss. Daher werden regelmäßig Nachrichten an die Heizungsregler bzw. Kühlvorrichtungen geschickt, um diese den aktuellen Umständen entsprechend einzustellen.

Neben der Beheizung des Raumes aufgrund von Bewegungsdetektion existieren meist noch weitere Regeln, die den Zustand der Aktoren beeinflussen: Beispielsweise ist es eine Stunde vor Ende der Arbeit nicht mehr nötig, den Raum zu beheizen, da der Heizkörper auch nach dem Schließen des Heizungsventils noch Wärme abgibt. Auf der anderen Seite ist es sinnvoll, dem System anhand von Zeitangaben mitzuteilen, wann die Arbeitszeit des Arbeitenden anfängt. So wird nicht erst mit Beginn dieser Zeit der Raum auf die Wohlfühltemperatur geregelt, sondern bereits im Vorfeld.

A.2. Einbruchskontrollsteuerung

In den meisten Fällen sind Arbeitsunterlagen vertraulich. Sie müssen vor unberechtigter Einsicht geschützt werden. Sofern sie in Papierform oder elektronisch auf einem tragbaren Medium abgelegt sind, kann ein Safe helfen, diese Daten zu schützen.

Wenn das Entwenden dieser Informationen lediglich erkannt werden muss, dann ist der Safe aus ökonomischer Sichtweise ein übertriebener Schutz. In diesen Fällen hilft eine Einbruchskontrollsteuerung im Hausautomatisierungssystem. Mit dieser werden zwar Einbrüche nicht aktiv verhindert, aber es kann der Einbruchszeitraum aufgenommen und der Benutzer oder die Polizei zeitnah benachrichtigt werden.

Eine solche Einbruchskontrollsteuerung kann beispielhaft folgende Sensoren und Aktoren besitzen:

Türkontaktsensor Ein Türkontaktsensor an jeder Tür nach draußen bestimmt, ob die Tür geöffnet ist oder nicht. Im Falle der Öffnung der Tür muss dieses an die Einbruchskontrollsteuerung gemeldet werden, damit diese kontrollieren kann, ob ein Einbruch stattgefunden hat.

Ein erster Indiz ist das Offenlassen der Tür: Je länger die Eingangstür offen ist, desto höher ist der Einbruchsverdacht. Allerdings wird es dann zu spät sein, wenn dieses Kriterium das einzige ist, um einen Einbruch zu detektieren. Einige Wohnungen

bzw. Häuser besitzen eine Balkon- bzw. Terrassentür. Bei dieser Tür ist diese Form der Einbruchsdetektion ungeeignet, da solche Türen durchaus länger geöffnet sind.

Fensterkontaktsensor Fensterkontaktsensoren können ein Indiz auf ein Einbruchsversuch geben, wenn der Zustand des Fensters mit den Erkenntnissen eines Bewegungsmelders abglichen wird.

Bewegungsmelder Der Bewegungsmelder erkennt eine Bewegung im Raum. Wenn also der Fensterkontaktsensor ein Öffnen des Fensters bemerkt, aber im Raum keine Bewegung detektiert wird, dann kann schlussgefolgert werden, dass jemand das Fenster von außen öffnet.

Alarmvorrichtung Nachdem ein Einbruchsversuch bzw. ein Einbruch erkannt worden ist, wird die Alarmvorrichtung benachrichtigt, damit diese Alarm schlägt. Fehlalarme sollten so weit wie möglich vermieden werden.

Diese Alarmvorrichtung benachrichtigt den Benutzer beispielsweise per SMS oder E-Mail. Fehlalarme sind bei dieser Benachrichtigungsart noch vernachlässigbar. Hier gilt der Grundsatz: Lieber einen Fehlalarm zu viel (false positive) als einen Alarm bei erfolgtem Einbruch zu wenig (true negative).

Eine nicht-stille Benachrichtigungsart ist die Verwendung einer Sirene, die Einbrecher abschrecken kann. Fehlalarme zu Ruhezeiten sind zu vermeiden.

Eine dritte Alarmvorrichtung ist die direkte Benachrichtigung der Polizei. Hier gilt der o. g. Grundsatz nicht. Fehlalarme sollten so niedrig wie möglich ausfallen.

Der Regelwerkknoten des Hausautomatisierungssystems erhält von den verschiedenen Sensoren Indizes auf einen möglichen Einbruchsversuch. Nur wenn bestimmte Voraussetzungen wie das Öffnen des Fensters ohne Bewegungsdetektion erfüllt sind, wird das System einen Alarm in der Alarmvorrichtung auslösen.

Sensoren und Aktoren zusammen bilden die Einbruchskontrollsteuerung und in Kombination mit bestimmten Regeln entsteht eine Alarmanlage. Die genauen Regeln, wie und wann Alarm gegeben wird, ist an dieser Stelle nicht relevant. Es sei darauf hingewiesen, dass anhand der Werte der Sensoren die Entscheidung gefällt wird, ob ein Alarm gegeben wird oder nicht.

Wenn alle Eingänge und Fenster der Wohnung bzw. des Hauses mit entsprechenden Schutzmechanismen versehen sind, dann kann die Einbruchskontrollsteuerung auf die gesamte Wohnung bzw. das gesamte Haus ausgeweitet werden und dennoch ist erkennbar, ob ein Einbruch im Arbeitsraum stattgefunden hat.

B Weiterführende Grundlagen

B.1. Netzwerktechnik

B.1.1. Internet Protocol Version 6 (IPv6)

Das Internet Protokoll (IP) ist ein standardisiertes Verfahren zur Übertragung von Daten als Pakete auf Vermittlungsschicht (Netzwerkschicht). Es wird in der Version 4 (IPv4) schon seit weit mehr als 30 Jahren erfolgreich mit hoher Interoperabilität eingesetzt.

Die Version 6 des Internet Protokoll (IPv6) ist eine Weiterentwicklung von IPv4 und wurde von der Internet Engineering Task Force (IETF) im Dezember 1998 in RFC 2460 [32] standardisiert. Aufgrund des ausgehenden Adressraumes in IPv4 wurde in IPv6, dem Internet Protokoll der nächsten Generation, der Adressraum von 2^{32} ($\approx 10^9$) auf 2^{128} ($\approx 10^{38}$) Adressen ausgeweitet.

Eine der Aufgaben der Vermittlungsschicht und somit von IPv6 ist die Herstellung einer gültigen Adressierung zwischen allen Teilnehmern eines Netzwerkes, aber auch zwischen verschiedenen Netzen. Das Internet Protokoll kümmert sich zusammen mit dem Internet Control Message Protocol (ICMP) zusätzlich um ein effizientes Routing zwischen diesen Teilnehmern. Hierzu kommen meist weitere Protokolle, die sich in das IP eingliedern.

Weiterführende Informationen können beispielsweise in [9] nachgelesen werden.

Adressnotation

Im Gegensatz zu IPv4-Adressen ist eine IPv6-Adresse schwerer zu merken. Daher gibt es verschiedene Regeln, um diese Adresse zu notieren:

1. IPv6-Adressen in nicht-verkürzter Schreibweise werden hexadezimal notiert, eingeteilt in acht Gruppen zu je vier Zahlen. Diese Gruppen werden mittels eines Doppelpunktes getrennt.

   ```
   2013:0000:00ad:0000
   :0000:0000:c0a8:0001
   ```

2. In jeder Gruppe dürfen führende Nullen weggelassen werden.

   ```
   2013:0:ad:0:0:0:c0a8:1
   ```

3. Eine Gruppe, die nur aus Nullen besteht („:0:“), muss nicht notiert werden und kann durch zwei Doppelpunkte gekennzeichnet werden („::“). Grenzen mehrere Gruppen, die nur aus Nullen bestehen, aneinander, so dürfen alle ausgelassen werden. Allerdings darf diese Regel nur einmal angewendet werden. `2013:0000:00ad::c0a8:0001`

4. Die Kombination der obigen zwei Regeln ist ebenfalls erlaubt. `2013:0:ad::c0a8:1`

5. Es ist auch möglich, die letzten zwei Gruppen in IPv4-Notation aufzuschreiben. `2013:0:ad::192.168.0.1`

Arten der Kommunikation

Es gibt drei grundlegend verschiedene Arten, wie in IPv6-Netzen kommuniziert werden kann:

Unicast bezeichnet die Kommunikation von einem Netzknoten zu einem anderen.

Multicast bezeichnet die Kommunikation von einem Netzknoten zu einer Gruppe von Netzknoten. Es ersetzt die Broadcast-Kommunikation des IPv4.

Alle Adressen im Bereich 0xff00::/8 entsprechen IPv6-Multicast-Adressen [33]. Der Sichtbereich dieser Adressen wird über eine *Scope ID* bestimmt. Die Art der Multicast-Adresse wird mit einem Flag ausgewählt. Ein Beispiel für eine permanente Multicast-Adresse ist ff02::2, die oft verwendet wird, wenn alle link-local Router in einem Netzwerk adressiert werden sollen.

Ein spezieller Typ von IPv6-Multicast-Gruppenadressen sind Unicast-basierte Adressen (nach RFC 3306 [34]). Bei diesem Typ ist der Unicast-Präfix in der Gruppenadresse enthalten. Folgendes Beispiel enthält den Präfix „2013:0:ad:0“:

ff38:30:2013:0:ad::<Group ID>

Anycast bezeichnet eine Kommunikation von einem Netzknoten zu einer Gruppe von Netzknoten. Im Gegensatz zum Multicast antwortet aber nur einer der Netzknoten. Ein Unternehmen kann so den Ausfall einer Verbindung durch eine alternative Verbindung kompensieren. Der Empfänger hat hierbei keine Kontrolle, zu welchem Mitglied der Anycast-Gruppe das Paket gesendet wird. Das Ziel wird ausschließlich durch verwendete Routing-Protokoll bestimmt.

Tabelle B.1. *Vergleich von Low-power and Lossy Network (LLN) Routing Protokollen [36, Table 1].*

	Topology	*Algorithm*	*Mobility*	*Scalability*	*Memory Usage*	*Energy Usage*	*Supported Traffic*
Dymo-low	Flat		Static mobile	low	high	high	P2P
LOAD	Flat	Distance Vector	Static mobile	high	low	low	P2P
TinyAODV	Flat	Distance Vector	Mobile	high	low	low	P2P
Hilow	Hierarchical	Tree-based	Static mobile	high	low	low	MP2P P2MP
Hydro	Hierarchical flat	Source Routing	Static	high	medium	high	MP2P, P2MP, P2P
RPL	Hierarchical flat	Distance Vector, Source Routing / In-network Routing	Static mobile	high	low	low	MP2P, P2MP, P2P

B.1.2. Routing Protocol for Low-power and Lossy Networks (RPL)

Das Routing Protocol for Low-power and Lossy Networks (RPL) (RFC 6550 [21]) ist ein dynamisches Routing-Protokoll und ist speziell für LLNs, zu denen auch drahtlose Sensornetze mit 6LoWPAN zählen, entwickelt worden. Es ist ein Route-Over Routing-Protokoll, d. h., es routet auf der Netzwerkschicht des OSI-Referenzmodells. Jeder Knoten besitzt hierbei eine eigene IPv6-Adresse, die i. A. zustandslos (stateless) aus der MAC-Adresse des Netzwerkadapters gebildet wird.

Warum RPL?

Es gibt eine ganze Reihe an Routing-Protokollen für LLNs. Darunter sind beispielsweise TinyAODV, Hilow, LOAD, Dymo-Low, Hydro und RPL, die eine Internet Protokoll Version 6 (IPv6)-Unterstützung besitzen. Tabelle B.1 gibt eine ausgewählte Übersicht zu diesen Routing-Protokollen. Im Folgenden werden die wichtigen Merkmale von RPL genannt:

Geringer Energie- und Speicherverbrauch RPL besitzt im Vergleich zu anderen Routing-Protokollen i. A. einen *geringen Energie- und Speicherverbrauch*. Diese beiden Eigenschaften sind vor allem in der Hausautomatisierung bei batteriebetriebenen Netzknoten wichtig. Allerdings wird die Anforderung an ein energie- und ressourcensparendes Routing-Protokoll auch von LOAD oder Hilow umgesetzt.

Distance-Vector-Algorithmus RPL, LOAD sowie TinyAODV nutzen den Distance-Vector-Algorithmus, um ein dynamisches Routing-Protokoll umzusetzen. Dieser

Algorithmus arbeitet nach dem selbstorganisierenden Prinzip: „Teile selbstständig deinen Nachbarn mit, wie die Umwelt für dich aussieht“. Diejenigen Netzknoten, die Routing-Tabellen pflegen, stellen Kostenmatrizen auf, um Pfade mit den geringsten Kosten zu ermitteln. Hierbei müssen die Netzknoten sich auf korrekte Informationen sowie auf korrekte Weitergabe dieser verlassen, damit das Routing effizient funktioniert.

Proactive vs. On-Demand Bis auf RPL arbeiten alle in Tabelle B.1 angegebenen Routing-Protokolle *On-demand*, d. h., sie bauen nur bei Bedarf eine Verbindung zwischen Netzknoten auf. RPL arbeitet im Gegensatz dazu *Proactice*, es werden also regelmäßig Verbindungsinformationen gepflegt, dafür entfällt der erhöhte Kommunikationsaufwand beim Verbindungsaufbau.

In der Hausautomatisierung wird üblicherweise regelmäßig und in kurzen Abständen kommuniziert – in Temperatursensor wird im Winter spätestens jede Minute einen Temperaturwert verschicken – daher ist ein proaktives Routing kein negatives Kriterium. Es kann sogar von Vorteil sein, da bei einem stabilen Netz die effizientesten Routen bereits bekannt sind.

Source Routing oder In-network Routing Bei RPL ist es möglich, entweder Source Routing oder In-network Routing zu benutzen, je nachdem wie viele Ressourcen die RPL-Router besitzen und was für die jeweilige Sensornetz-Anwendung besser geeignet ist. Ein Mix-Mode wird allerdings (noch) nicht unterstützt.

Beim Source Routing entscheidet der Sender einer Nachricht, über welche Router die Nachricht versendet wird, sofern dieser Netzknoten eine Routing-Tabelle besitzt. Diese Informationen müssen jeder Nachricht in Form eines Source-Routing-Headers (SRH) angehangen werden. In RPL ist im Source Routing Modus nur ein spezieller Netzknoten im Besitz einer Routing-Tabelle. Allerdings ist jeder Knoten in der Lage, eine Nachricht an diesen Knoten bzw. über diesen Knoten an andere zu senden.

Beim In-network Routing werden im Gegensatz zum Source Routing Routing-Tabellen in jedem RPL-Router gespeichert. In diesem Modus ist es möglich, Multicast-Nachrichten zu verschicken.

6LoWPAN-Unterstützung RPL ist zur Verwendung mit 6LoWPAN entwickelt worden, kann aber auch allgemein in IPv6 Netzwerken eingesetzt werden. Beim Design wurde zusätzlich darauf geachtet, dass mehrere Link-Layer-Protokolle in einem RPL-Netzwerk verwendet werden können. Damit ist es möglich, zukünftige Link-Layer-Technologien mit bereits vorhandenen zu kombinieren und dennoch ein effizientes und dynamisches Routing zu ermöglichen.

Eigene Sicherheitsmechanismen RPL definiert eigene Sicherheitsmechanismen, sofern andere Sicherheitsmechanismen wie Link-Layer-Sicherheit für die Sensornetzanwendung nicht ausreichend sind. Im *Preinstalled Mode* können nur Netzknoten kommunizieren, wenn sie einen Pre-Shared-Key besitzen. Im *Authenticated Mode*

müssen sich RPL-Router zusätzlich zum Pre-Shared-Key einen zweiten Schlüssel von einem Netzknoten anfordern, der eine Authentifizierung ermöglicht.

Diese Eigenschaften haben dazu geführt, dass RPL sich als Routing-Protokoll für LLNs weitgehend durchgesetzt hat.

RPL Netzwerk-Topologie

RPL bildet eine zentralistisch organisierte Netzwerkinfrastruktur. Das Zentrum dieser Infrastruktur ist der RPL Root. Alle anderen Netzknoten bilden Pfade zu diesen RPL Root. Die RPL Hosts sind hierbei Endknoten, die selbst keine RPL-Routingfunktionalität besitzen. Zwischen RPL Hosts und dem RPL Root finden sich ggf. RPL Router, die die Nachrichten der RPL Hosts an den Root weiterleiten. Damit existieren drei verschiedene Typen von Knoten in einem RPL-Netz:

RPL Root (Border Router) Der RPL Root organisiert das gesamte RPL Netzwerk. Typischerweise ist der RPL Root ein Border Router, der die RPL Domaine an andere Netzwerke koppelt und somit dafür sorgt, dass alle RPL-Netzknoten nach außen kommunizieren können und umgekehrt.

Der Vorteil einer zentralistisch organisierten Infrastruktur ist es, dass alle Nachrichten effizient (mit minimalen Kosten) zum Root gesendet werden können. Der Root muss damit auf jeden Fall eine Rooting-Tabelle verwalten und alle RPL-Netzknoten kennen.

RPL Hosts RPL Hosts suchen sich den nächstgelegenen RPL Router bzw. den RPL Root, sofern dieser sich in Reichweite befindet. Anders als im IEEE 802.15.4-Standard ist es für RPL Hosts erlaubt, mit anderen RPL Hosts direkt zu kommunizieren, falls diese sich in Reichweite befinden. Damit wird eine optimale Übertragungszeit zwischen zwei Hosts mit nur einem Hop erreicht.

RPL Router RPL Router haben die Aufgabe, andere RPL Router sowie RPL Hosts miteinander effizient zu verbinden. Dazu werden die Kosten einer Verbindung nach einem vom RPL Root angegebenen Muster berechnet und an den RPL Root gesendet.

Literaturverzeichnis

Grundlagen

[1] A. H. Chowdhury, M. Ikram, H.-S. Cha, H. Redwan, S. M. S. Shams, K.-H. Kim und S.-W. Yoo, „Route-over vs mesh-under routing in 6LoWPAN“, in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, Ser. IWCMC '09, Leipzig, Germany: ACM, 2009, S. 1208–1212, ISBN: 978-1-60558-569-7. DOI: 10.1145/1582379.1582643 (siehe S. 10).

[2] F. Mattern und C. Floerkemeier, „Vom Internet der Computer zum Internet der Dinge“, *Informatik-Spektrum*, Bd. 33, Nr. 2, S. 107–121, Apr. 2010 (siehe S. 10 f.).

[3] S. Kaufmann und K. Kudra, *Der Computer und der Mensch, Der Einfluss des Computers und des Internets auf die Werte unserer Gesellschaft*, KGS-Semesterarbeit, Studienrichtung Umweltingenieurwesen, Juni 2005. Adresse: http://www.semestra.ch/data/files/computer_und_der_mensch.pdf (besucht am 15. 08. 2013) (siehe S. 11).

[4] M. Weiser. (1991). The computer for the 21st century, Adresse: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html (besucht am 16. 05. 2013) (siehe S. 11).

[5] AFUL Interoperability Working Group. (o. J.). Definition der Interoperabilität. Übersetzung: André Schnabel, Adresse: http://interoperability-definition.info/de/ (besucht am 08. 08. 2013) (siehe S. 14).

[6] A. Drossos und H. Lorenz, „Ansteuerung von Sensoren und Aktoren in der Hausautomatisierung“, Paper zum Forschungsprojekt Sensornetze, University of Applied Sciences Dresden, März 2013 (siehe S. 15).

[7] Bundesamt für Sicherheit in der Informationstechnik, „Leitfaden Informationssicherheit, IT-Grundschutzkompakt“, Feb. 2012, Diese Broschüre ist Teil der Öffentlichkeitsarbeit des BSI; sie wird kostenlos abgegeben und ist nicht zum Verkauf bestimmt. Adresse: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Leitfaden/GS-Leitfaden_pdf.pdf?__blob=publicationFile (besucht am 15. 08. 2013) (siehe S. 35).

[8] J. Black, „Authenticated encryption“, 2004 (siehe S. 37).

[9] H. Wiese, *Das neue Internetprotokoll IPv6, Mobilität, Stabilität, unbeschränkter Adressraum und einfaches Management*. Hanser, 2002, ISBN: 3-446-21685-5 (siehe S. 41, 97).

[10] C. Eckert, *IT-Sicherheit: Konzepte, Verfahren, Protokolle*. Oldenbourg, 2006, ISBN: 9783486578515 (siehe S. 41).

[11] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Ser. Programming / Security. Wiley, 1995, ISBN: 9780471117094 (siehe S. 49).

[12] A. J. Menezes, S. A. Vanstone und P. C. V. Oorschot, *Handbook of Applied Cryptography*, 5th. Boca Raton, FL, USA: CRC Press, Inc., 2001, ISBN: 0849385237 (siehe S. 49).

[13] A. Drossos und H. Lorenz, „Einführung in Contiki-OS, Implementierungsaspekte und Netzwerk-Stack“, Paper, University of Applied Sciences Dresden, März 2013 (siehe S. 82).

Standards und RFCs

[14] R. Shirey, *Internet Security Glossary, Version 2*, RFC 4949 (Informational), Internet Engineering Task Force, Aug. 2007. Adresse: `http://www.ietf.org/rfc/rfc4949.txt` (siehe S. 36).

[15] D. McGrew, *An Interface and Algorithms for Authenticated Encryption*, RFC 5116 (Proposed Standard), Internet Engineering Task Force, Jan. 2008. Adresse: `http://www.ietf.org/rfc/rfc5116.txt` (siehe S. 37).

[16] *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, National Institute of Standards and Technology, Mai 2004 (siehe S. 38).

[17] M. Dworkin, *Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC*, SP 800-38D, National Institute of Standards und Technology, Nov. 2007 (siehe S. 38).

[18] *Advanced Encryption Standard (AES)*, FIPS 197, National Institute of Standards and Technology, Nov. 2001 (siehe S. 38).

[19] G. Montenegro, N. Kushalnagar, J. Hui und D. Culler, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC 4944 (Proposed Standard), Updated by RFCs 6282, 6775, Internet Engineering Task Force, Sep. 2007. Adresse: `http://www.ietf.org/rfc/rfc4944.txt` (siehe S. 38).

[20] N. Kushalnagar, G. Montenegro und C. Schumacher, *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, RFC 4919 (Informational), Internet Engineering Task Force, Aug. 2007. Adresse: `http://www.ietf.org/rfc/rfc4919.txt` (siehe S. 39).

[21] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur und R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, RFC 6550 (Proposed Standard), Internet Engineering Task Force, März 2012. Adresse: http://www.ietf.org/rfc/rfc6550.txt (siehe S. 40, 49, 99).

[22] S. Kent und K. Seo, *Security Architecture for the Internet Protocol*, RFC 4301 (Proposed Standard), Updated by RFC 6040, Internet Engineering Task Force, Dez. 2005. Adresse: http://www.ietf.org/rfc/rfc4301.txt (siehe S. 40).

[23] C. Kaufman, P. Hoffman, Y. Nir und P. Eronen, *Internet Key Exchange Protocol Version 2 (IKEv2)*, RFC 5996 (Proposed Standard), Updated by RFC 5998, Internet Engineering Task Force, Sep. 2010. Adresse: http://www.ietf.org/rfc/rfc5996.txt (siehe S. 41).

[24] C. Kaufman, *Internet Key Exchange (IKEv2) Protocol*, RFC 4306 (Proposed Standard), Obsoleted by RFC 5996, updated by RFC 5282, Internet Engineering Task Force, Dez. 2005. Adresse: http://www.ietf.org/rfc/rfc4306.txt (siehe S. 41).

[25] T. Dierks und E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246 (Proposed Standard), Updated by RFCs 5746, 5878, 6176, Internet Engineering Task Force, Aug. 2008. Adresse: http://www.ietf.org/rfc/rfc5246.txt (siehe S. 42, 45).

[26] IANA. (Aug. 2005). Transport Layer Security (TLS) Parameters. DTLS-Kompatiblität angegeben mit DTLS-OK, letztes Update erfolgte am 19.08.2013, Adresse: http://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml (besucht am 22. 09. 2013) (siehe S. 43, 46, 85).

[27] E. Rescorla und N. Modadugu, *Datagram Transport Layer Security Version 1.2*, RFC 6347 (Proposed Standard), Internet Engineering Task Force, Jan. 2012. Adresse: http://www.ietf.org/rfc/rfc6347.txt (siehe S. 44).

[28] IEEE Standards Association, *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, aktuell verfügbare Version ist IEEE 802.15.4-2011, 16. Juni 2011. Adresse: http://standards.ieee.org/about/get/802/802.15.html (besucht am 06. 10. 2013) (siehe S. 49).

[29] P. Eronen und H. Tschofenig, *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*, RFC 4279 (Proposed Standard), Internet Engineering Task Force, Dez. 2005. Adresse: http://www.ietf.org/rfc/rfc4279.txt (siehe S. 73 f.).

[30] J. Romkey, *Nonstandard for transmission of IP datagrams over serial lines: SLIP*, RFC 1055 (INTERNET STANDARD), Internet Engineering Task Force, Juni 1988. Adresse: http://www.ietf.org/rfc/rfc1055.txt (siehe S. 79).

[31] E. Barker, W. Barker, W. Burr, W. Polk und M. Smid, „Recommendation for key management - part 1: general (revision 3)“, in *NIST Special Publication 800-57 Part 1*, Juli 2012 (siehe S. 85).

[32] S. Deering und R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460 (Draft Standard), Updated by RFCs 5095, 5722, 5871, 6437, 6564, Internet Engineering Task Force, Dez. 1998. Adresse: http://www.ietf.org/rfc/rfc2460.txt (siehe S. 97).

[33] R. Hinden und S. Deering, *Internet Protocol Version 6 (IPv6) Addressing Architecture*, RFC 3513 (Proposed Standard), Obsoleted by RFC 4291, Internet Engineering Task Force, Apr. 2003. Adresse: http://www.ietf.org/rfc/rfc3513.txt (siehe S. 98).

[34] B. Haberman und D. Thaler, *Unicast-Prefix-based IPv6 Multicast Addresses*, RFC 3306 (Proposed Standard), Updated by RFCs 3956, 4489, Internet Engineering Task Force, Aug. 2002. Adresse: http://www.ietf.org/rfc/rfc3306.txt (siehe S. 98).

Verwandte Forschungsarbeiten

[35] P. Sharma und S. Nain, *A comparative study of security protocols for WSNs*, Feb. 2013 (siehe S. 35).

[36] O. Gaddour und A. KoubíA, „Rpl in a nutshell: a survey“, *Comput. Netw.*, Bd. 56, Nr. 14, S. 3163–3178, Sep. 2012, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2012.06.016 (siehe S. 40, 99).

[37] S. Raza, H. Shafagh, K. Hewage, R. Hummen und T. Voigt, „Lithe: Lightweight Secure CoAP for the Internet of Things“, *Sensors Journal, IEEE*, Bd. 13, Nr. 10, S. 3711–3720, Okt. 2013, zum Zeitpunkt der Erstellung noch unveröffentlicht, ISSN: 1530-437X. DOI: 10.1109/JSEN.2013.2277656 (siehe S. 46 f.).

[38] S. Raza, T. Voigt und U. Roedig, „6LoWPAN extension for IPsec“, in *Interconnecting Smart Objects with the Internet Workshop*, März 2011 (siehe S. 47).

[39] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt und U. Roedig, „Securing communication in 6LoWPAN with compressed IPsec“, in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, Juni 2011, S. 1–8. DOI: 10.1109/DCOSS.2011.5982177 (siehe S. 47).

[40] S. Raza, T. Voigt und V. Juvik, „Lightweight IKEv2, A key management solution for both compressed IPsec and IEEE 802.15.4 security“, *Proceedings of the IETF International Workshop on Smart Object Security*, März 2012 (siehe S. 47).

[41] S. Raza, D. Trabalza und T. Voigt, „6LoWPAN compressed DTLS for CoAP“, in *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, 2012, S. 287–289, ISBN: 978-1-4673-1693-4. DOI: 10.1109/DCOSS.2012.55 (siehe S. 47).

[42] M. Brachmann, O. Garcia-Morchon und M. Kirsche, „Security for practical CoAP applications, Issues and solution approaches“, in *Proc. 10th GI/ITG KuVS Fachgespraech Sensornetze*, H. Frey, Hrsg., University of Paderborn, Sep. 2011 (siehe S. 47).

[43] A. R. Wacker, *Key Distribution Schemes for Resource-Constrained Devices in Wireless Sensor Networks*. University of Stuttgart, Mai 2008 (siehe S. 51).

[44] S. Bendeich, *Random key distribution verfahren in wsns – vergleich verschiedener probabilistischer ansätze*, 2011 (siehe S. 51).

[45] L. Buttyán und J. Hubaux, *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge University Press, 2008, ISBN: 9781139466608 (siehe S. 59).

[46] C. Karlof und D. Wagner, „Secure routing in wireless sensor networks: attacks and countermeasures“, *Ad Hoc Networks*, Bd. 1, Nr. 2–3, S. 293–315, 2003, ISSN: 1570-8705. DOI: 10.1016/S1570-8705(03)00008-8 (siehe S. 59).

[47] L. Wallgren, S. Raza und T. Voigt, „Routing attacks and countermeasures in the RPL-based Internet of Things“, *International Journal of Distributed Sensor Networks*, Bd. 2013, 2013. DOI: 10.1155/2013/794326 (siehe S. 59).

[48] N. Sastry und D. Wagner, „Security considerations for IEEE 802.15.4 networks“, in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, ACM Press, 2004, S. 32–42 (siehe S. 74).

[49] Machemehl, *Sicherheitsarchitektur in einem drahtlosen Sensornetz*, Verantwortlicher Hochschullehrer Prof. Dr.-Ing. habil. A. Finger, 2012 (siehe S. 77 f.).

[50] E.-O. Blaß, H. Junker und M. Zitterbart, „Effiziente Implementierung von Public-Key-Algorithmen für Sensornetze“, Apr. 2005 (siehe S. 86).

[51] N. Gura, A. Patel, A. Wander, H. Eberle und S. C. Shantz, „Comparing elliptic curve cryptography and rsa on 8-bit cpus“, in *Cryptographic Hardware and Embedded Systems – CHES 2004, 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, Bd. 3156, Springer Berlin Heidelberg, 2004, S. 119–132. DOI: 10.1007/978-3-540-28632-5_9 (siehe S. 86).

[52] H. Lang. (2009). Modulare exponentiation (variante 2), Adresse: http://www.inf.fh-flensburg.de/lang/krypto/algo/modexp2.htm (besucht am 25. 11. 2013) (siehe S. 89).

Sonstige Literatur

[53] J. Ihlenfeld, „IOT, Konsortium will Internet der Dinge voranbringen“, *Golem.de*, 8. Jan. 2013. Adresse: http://www.golem.de/news/iot-konsortium-will-internet-der-dinge-voranbringen-1301-96746.html (besucht am 11. 09. 2013) (siehe S. 11).

[54] JuraforumWiki-Redaktion. (2013). Schutz der Privatsphäre, Adresse: http://www.juraforum.de/lexikon/schutz-der-privatsphaere (besucht am 15. 08. 2013) (siehe S. 12).

[55] Frauenhofer ITWM, *HexaBus – the Kranken of Home Automation!, Standardkonformes Open-Source-Hausbussystem*, Flyer, eingebettet in das Projekt „mySmartGrid“, 2013. Adresse: https://dl.dropboxusercontent.com/u/64608113/Flyer/hpc_flyer_Hexabus_DE-cut.pdf (besucht am 09. 08. 2013) (siehe S. 15, 31).

[56] ——, (o. J.). HexaBus – the Kranken of Home Automation!, Wiki, Adresse: https://github.com/mysmartgrid/hexabus/wiki (besucht am 09. 08. 2013) (siehe S. 15, 31).

[57] ——, *Geräte der mySmartGrid-Platform*, Flyer, 2012. Adresse: https://www.mysmartgrid.de/sites/default/files/HPC_Flyer_Geraete_der_mySmartGrid-Plattform_A3_DE-Web.pdf (besucht am 29. 09. 2013) (siehe S. 31).

[58] T. I. Inc. (o. J.). Overview for fram series, Adresse: http://www.ti.com/lsds/ti/microcontroller/16-bit_msp430/fram/overview.page (besucht am 01. 12. 2013) (siehe S. 50).

[59] A. Donath, „Satis Smart Toilet, Toilettenschüssel gehackt“, *Golem.de*, 5. Aug. 2013. Adresse: http://www.golem.de/news/satis-smart-toilet-toilettenschuessel-gehackt-1308-100782.html (besucht am 08. 09. 2013) (siehe S. 57).

[60] Fraunhofer-Gesellschaft. (1. Aug. 2012). Smart wirless power outlets. Research News, Adresse: http://www.fraunhofer.de/en/press/research-news/2012/august/smart-wireless-power-outlets.html (besucht am 29. 09. 2013) (siehe S. 69).

[61] Thingsquare AB. (o. J.). Thingsquare homepage, About thingsquare, Adresse: http://thingsquare.com (besucht am 02. 12. 2013) (siehe S. 69).

[62] ——, *Technology white paper*, Mai 2013. Adresse: http://thingsquare.com/download/Thingsquare%20Technology%20Whitepaper%20-%2020130524.pdf (besucht am 02. 12. 2013) (siehe S. 69).

[63] ——, *Product brief*, Mai 2013. Adresse: http://thingsquare.com/download/Thingsquare%20Product%20Brief%20-%2020130524.pdf (besucht am 02. 12. 2013) (siehe S. 69).

[64] ——, *Thingsquare evaluation kit user guide*, Nov. 2013. Adresse: http://thingsquare.com/download/Thingsquare%20kit%20user%20guide%20-%2020131101.pdf (besucht am 02. 12. 2013) (siehe S. 69).

[65] J. Schmidt, „Zukunftssicher verschlüsseln mit Perfect Forward Secrecy“, *Heise.de*, 25. Juli 2013. Adresse: http://www.heise.de/security/artikel/Zukunftssicher-Verschluesseln-mit-Perfect-Forward-Secrecy-1923800.html (besucht am 02. 12. 2013) (siehe S. 73 f.).

[66] T. Simonite, „Die Krypto-Strategie der NSA ist keine Überraschung“, *Heise.de*, 13. Sep. 2013. Adresse: http://www.heise.de/tr/artikel/Die-Krypto-Strategie-der-NSA-ist-keine-Ueberraschung-1955013.html (besucht am 01. 12. 2013) (siehe S. 86).

[67] (2013). AVR-Crypto-Lib. es ist kein Name als Author angegeben, sondern nur bg@das-labor.org, Adresse: http://www.das-labor.org/wiki/AVR-Crypto-Lib (besucht am 13. 11. 2013) (siehe S. 87).

[68] E. Dostert, „Wohlfühlen am Arbeitsplatz, Orte des Schreckens“, *Süddeutsche Zeitung*, 17. Mai 2010. Adresse: http://www.sueddeutsche.de/karriere/wohlfuehlen-am-arbeitsplatz-orte-des-schreckens-1.539060 (besucht am 06. 09. 2013) (siehe S. 93).

Glossar

Aggregationsknoten

Aggregationsknoten sammeln Sensorinformationen von den Sensorknoten, um sie anschließend auswerten zu können. Der Regelwerkknoten ist ein Aggregationsknoten. Es können aber noch weitere Aggregationsknoten in einem Sensor- bzw. Sensor-Aktor-Netz existieren.

[7]

Aktorknoten

Aktorknoten sind Netzknoten, die dem Hausautomatisierungssystem eine Schnittstelle zur Ansteuerung der integrierten Aktoren bereitstellen. Da mit Hilfe von Aktoren Aktionen ausgeführt werden können, sind sie essentielle Bestandteile des Systems.

[63]

Anwendungsebene

Die Anwendungsebene soll sich im Gegensatz zur Anwendungsschicht (Schicht 7) nicht auf eine konkrete Schicht des OSI-Referenzmodells beziehen. Es sind vielmehr allgemein die höheren Schichten des OSI-Referenzmodells (ab Schicht 4) gemeint. Wenn ein Sensorknoten mit dem Regelwerkknoten kommuniziert, dann wird seine Nachricht i. A. über mehrere Hops zum Regelwerkknoten geroutet. Wenn diese Hops für den jeweiligen Sachverhalt überflüssig sind, dann wird allgemein in dieser Masterarbeit von einer *Kommunikation auf Anwendungsebene* gesprochen.

[62 f.]

Automationsebene

Die Automationsebene ist eine der zwei logischen Ebenen im Hausautomatisierungssystem und kümmert sich um alle Belange des Automationsprozesses. Die zweite logische Ebene ist die Managementebene.

Die Automationsebene beinhaltet folgende Komponenten: mind. ein Regelwerkknoten, Sensorknoten, Aktorknoten sowie Routingknoten sowie Mischformen dieser Netzknoten.

Diese Komponenten tragen dazu bei, dass dem Hausbewohner der Umgang im Haus durch eine automatische Ausführung gewisser Tätigkeiten erleichtert wird.

[63]

dedizierte Regelverarbeitung

Der Begriff „dedizierte Regelverarbeitung“ ist in dieser Masterarbeit ein Synonym

für die zentral-dedizierte Regelverarbeitung. Diese Masterarbeit hat zwar System-Architekturen mit verteilt-dedizierter Regelverarbeitung untersucht, ist aber zum Schluss gekommen ist, dass die verteilt-dedizierte Regelverarbeitung nur für größere bzw. komplexe Hausautomatisierungsszenarien eingesetzt werden sollte. Solche komplexen Hausautomatisierungsszenarien werden allerdings eher die Ausnahme sein..

[28]

freie Hardware

Freie Hardware (engl. *open source hardware*) ist nach lizenzkostenfreien Bauplänen gefertigt worden. Dies bedeutet nicht, dass die verwendete Hardware oder teile hiervon frei von einer Lizenz oder einem Patent sein muss.

[15]

Hausautomationskoordinator

Der Hausautomationskoordinator ist eine Komponente in der Managementebene des Hausautomatisierungssystems und kümmert sich um die Verwaltung (Koordination) der Automationsregeln und die mit diesen verbundene, notwendige Kommunikation im Hausautomatisierungssystem. Mit anderen Worten: Diese Komponente legt fest, mit wem ein Netzknoten kommunizieren muss, damit die Automationsregeln erfolgreich umgesetzt werden. Der Hausautomationskoordinator darf nicht mit dem Koordinator eines PAN-Netzes (PAN-Coordinator) verwechselt werden.

[63]

Interface

Das Interface ist eine Komponente in der Managementebene des Hausautomatisierungssystems und kümmert sich um die Belange des Nutzers bzw. Hausbewohners, indem es eine Kommunikationsschnittstelle (i. A. grafisch) zur Verfügung stellt. Mit dieser Komponente ist der Nutzer in der Lage, das Hausautomatisierungssystem zu konfigurieren und zu überwachen. Beispielsweise kann der Nutzer Automationsregeln anlegen, ändern oder löschen und mit anderen kombinieren. Aber auch das Erzeugen von Übersichten, die die Sensorinformationen und die Automationsvorgänge zeigen, ist Aufgabe des Interfaces.

[63]

Interoperabilität

Die Interoperabilität ist die Fähigkeit unterschiedlicher Systeme, möglichst nahtlos zusammenzuarbeiten.

[13, 15, 19, 42]

MAC-Adresse

Die Media-Access-Control-Adresse (MAC-Adresse) ist die Hardware-Adresse jedes einzelnen Netzwerkadapters, die als eindeutiger Identifikator des Geräts in einem Rechnernetz dient.

[40, 99]

Managementebene

Die Managementebene ist eine der zwei logischen Ebenen im Hausautomatisierungssystem und kümmert sich um alle Belange des Hausbewohners. Die zweite logische Ebene ist die Automationsebene.
Die Managementebene beinhaltet folgende Komponenten: mind. ein Interface und einen Hausautomationskoordinator.
Diese Komponenten sorgen dafür, dass der Hausbewohner mit dem Hausautomatisierungssystem kommunizieren kann und das seine Regeln für die Steuerung des Automationsprozesses an die Automationsebene weitergegeben werden..

[63]

Medium Access Control

Medium Access Control (MAC) ist eine Schicht im Netzwerk-Stack, die sich auf der Sicherungsschicht (engl. *Link Layer*) befindet. Diese Schicht hat die Aufgabe Zugriffe auf das Medium – in drahtlosen Sensornetzen also die Kommunikation über die Luft – zu steuern.
Diese Abkürzung darf nicht mit Message Authentification Code (Message Authentification Code) verwechselt werden.

[*siehe* Message Authentification Code]

Message Authentification Code

Der Message Authentication Code (MAC) ist eine Prüfinformation, die einer Nachricht angehänt wird, und dient zur Prüfung der Echtheit der Nachricht, also der Authentizität und Integrität.
Diese Abkürzung darf nicht mit Medium Access Control verwechselt werden. Aus diesem Grund wird in dieser Arbeit die deswegen beliebte Abkürzung Message Integration Check anstelle von Message Authentifiation Code verwendet.

[113, *siehe* Medium Access Control]

Message Integration Check

Der Message Integration Check (MIC) ist eine Methode zur Prüfung der Echtheit einer Nachricht, also der Authentizität und Integrität. Der MIC untersucht dabei eine Prüfinformation, die der Nachricht angehänt wird.
Diese Abkürzung ist eine alternative Bezeichnung zu Message Authentication Code (Message Authentification Code), welche leicht mit Medium Access Control (Medium Access Control) verwechselt werden kann. Aus diesem Grund wird in dieser Arbeit die Bezeichung MIC equivalent zu Messag Authentification Code gehandhabt, um die eben genannte Verwechselung zu vermeiden.

[113]

OSI-Referenzmodell

Das Open System Interconnection (OSI) Referenzmodell, 1984 von der International Organisation for Standardization als Standard veröffentlicht, ist bis heute noch ein Referenzmodell zum Vergleich von Netzwerkprotokollen und gliedert Protokolle

in sieben Schichten ein: von der Bitübertragungsschicht bis hin zur Anwendungsschicht.

[9 f., *siehe* TCP/IP-Referenzmodell]

Out-of-Bound-Kanal

Der Out-of-Bound-Kanal ist ein Kanal, der normalerweise für die Kommunikation zwischen Netzteilnehmern nicht genutzt wird. Manchmal wissen Netzteilnehmer eines Systems nicht einmal, dass ein solcher Kanal existiert, über den Informationen ausgetauscht werden.
Der Angreifer nutzt einen Out-of-Bound-Kanal, wenn er eine Wormhole Attacke durchführt, indem er Nachrichten nicht über den üblichen Kanal (IEEE 802.15.4) versendet, sondern beispielsweise über WLAN oder Ethernet.

[73, 75 f., 79 f., 85, 92]

Regelwerkknoten

Regelwerkknoten sind Netzknoten, die eine Regelverarbeitung (Regelwerk) besitzen. In einem Hausautomatisierungssystem ist mindestens ein Regelwerkknoten vorhanden. Das Regelwerk wird genutzt, um die Automatisierung des Hauses zu realisieren, und gibt an, welche Bedingungen an bestimmte Aktoransteuerungen geknüpft sind. Hierzu werden insbesondere die Sensorinformationen genutzt; aber auch zeitliche Bedingungen sind gebräuchlich.

[63]

Routingebene

Genauso wie die Anwendungsebene soll sich die Routingebene nicht auf eine konkrete Schicht des OSI-Referenzmodells beziehen. Es sind vielmehr allgemein die niedrigeren Schichten des OSI-Referenzmodells (bis zur Schicht 3) gemeint.
Wenn ein Sensorknoten mit dem Regelwerkknoten kommuniziert, dann wird seine Nachricht i. A. über mehrere Hops zum Regelwerkknoten geroutet. Wenn diese Hops für den jeweiligen Sachverhalt wichtig sind, dann wird allgemein in dieser Masterarbeit von einer *Kommunikation auf Routingebene* gesprochen.
Gleiches gilt bei der Organisation des Routings: Auch hier wird von einer Kommunikation auf Routingebene gesprochen.

[62]

Sensorknoten

Sensorknoten sind Netzknoten, die Sensorinformationen an das Hausautomatisierungssystem weitergeben. Diese Sensorinformationen beziehen sich entweder auf den Netzknoten selbst (Batteriestatus, Temperatur im Gerät) oder auf die Umgebung (Umgebungstemperatur, Luftfeutigkeit).

[7, 63]

TCP/IP-Referenzmodell

Das TCP/IP Referenzmodell ist ähnlich dem OSI-Referenzmodell dazu da, Netzprotokolle zum Vergleich in Schichten einzuteilen. Dieses Modell wurde speziell

für die TCP/IP-Protokoll-Familie erstellt, da diese mehr als 500 Protokolle umfasst. Dabei werden diese Protokolle in drei Schichten eingeteilt: Internet-, Transport- und Anwendungsschicht. Die Netzzugangsschicht enthält keine Protokolle der TCP/IP-Familie, sondern dient lediglich als Platzhalter, da die Internetprotokolle die Aufgabe haben, Pakete über mehrere Punkt-zu-Punkt-Verbindungen weiterzuleiten und sich daher nicht um den Zugriff auf das Übertragungsmediums kümmern.
[*siehe* OSI-Referenzmodell]

verteilt-dedizierte Regelverarbeitung
Hausautomatisierungssysteme, die eine System-Architektur mit verteilt-dedizierter Regelverarbeitung umsetzen, verteilen die Regeln auf verschiedene, dediziert-arbeitende Regelwerkknoten. Diese Netzknoten besitzen keine weiteren Komponenten der Automationsebene und kümmern sich vor allem um diejenigen Regeln, die ein bestimmtes Subsystem im Hausautomatisierungssysteme betreffen.
[28]

verteilte Regelverarbeitung
Hausautomatisierungssysteme, die eine System-Architektur mit verteilter Regelverarbeitung umsetzen, verteilen die Regeln auf verschiedene, i. A. nicht-batteriebetriebene Netzknoten. Diese Netzknoten besitzen meist noch weitere Komponenten der Automationsebene.
[28, 31]

zentral-dedizierte Regelverarbeitung
Hausautomatisierungssysteme, die eine System-Architektur mit zentral-dedizierter Regelverarbeitung umsetzen, verteilen die Regeln *nicht* auf verschiedene Netzknoten, sondern bewahren alle Regeln in einem zentralen, dedizierten Netzknoten auf. Diese Netzknoten besitzen keine weiteren Komponenten der Automationsebene.
[28, 63]

Akronyme

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AE	Authenticated Encryption
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
BSI	Bundesamt für Sicherheit in der Informationstechnik
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CoAP	Constrained Application Protocol
CTR	Counter
DTLS	Datagram Transport Layer Security
FFD	Full Function Device
GCM	Galois Counter Mode
HC	Header Compression
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protokoll
IPsec	Internet Protokoll Security
IPv4	Internet Protokoll Version 4
IPv6	Internet Protokoll Version 6
LLN	Low-power and Lossy Network
LoWPAN	Low-power Wireless Personal Area Network
MAC	Message Authentication Code
PAN	Personal Area Network
PFS	Perfect Forward Secrecy
PSK	Pre-Shared Key
RFD	Reduced Function Device
RPL	Routing Protocol for Low-power and Lossy Networks
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security

UDP	User Datagram Protocol
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

Dieses Dokument wurde mit dem Textsatzprogramm LaTeX gesetzt.